Multivariate Approaches for Process Model Forecasting

Yongbo Yu, Jari Peeperkorn, Johannes De Smedt, and Jochen De Weerdt

Research Center for Information Systems Engineering (LIRIS), KU Leuven, Belgium
{FirstName}.{LastName}@kuleuven.be

Abstract. Recently, inspired by predictive process monitoring, the modeling and prediction of the entire process information system has been proposed as process model forecasting. By forecasting individual elements of a directly-follows graph, the future state of the system can be predicted. However, the current state-of-the-art principally employs univariate forecasting of direct-follows relationships (DFs). This univariate approach overlooks the process structure and possible relations between different elements within the process. This paper introduces a comprehensive deployment of multivariate time series models, more specifically a range of different machine- and deep learning approaches, to forecast DFs. These are benchmarked on different event logs collected from real-life event processes. Our extensive experiments reveal that the performance of these forecasting models varies significantly across different processes, highlighting the importance of model selection.

Keywords: Process Model Forecasting · Time Series Forecasting · Deep Learning.

1 Introduction

In recent years numerous research advancements in the field of predictive process monitoring (PPM) have been proposed, driven by the rapid development and widespread application of machine learning and deep learning. PPM aims to forecast future elements of ongoing cases in the information system, including the most probable next activities [8], outcomes [26], and remaining runtimes [25]. Notably, the integration of recurrent neural networks (RNNs) into this domain has improved predictive performance significantly.

Recently, a new paradigm known as Process Model Forecasting (PMF) has emerged, focusing on predicting future states of the overall process model over a long-term horizon [6]. The forecasted process model represents the will-be process, enabling the exploration of tactical and strategic questions such as "Are my bottlenecks persistent over time?" and "Will the ratio of granted loan applications change in the next quarter?". The evolution of process behavior can be captured through the shift in the direct-follows occurrences (DFs) over time. By deconstructing the time dimensions, DFs are predicted by univariate time series techniques and transformed to the directly-follows graphs (DFGs) as the forecasted process model. However, the dependencies and interactions between DFs might influence each other or be influenced by the temporal evolution of the system in general (e.g. drifts), which is not considered by univariate time series forecasting techniques that handle each DF separately. For example, if a DF pair of activity A to B is followed by another DF pair of activity B to C, the growth of the former DF tends to also result in an increase in the occurrence of the latter one. The underlying pattern of DFs could be captured by leveraging the multivariate time series forecasting models.

Inspired by the extensive research of deep learning applications in PPM, we explore the use of current state-of-the-art time series forecasting models based on machine learning (ML) and deep learning (DL) to do multivariate PMF. More specifically, we incorporate the classes of Gradient Boosting Decision Tree (GBDT), Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), Multilayer Perceptrons (MLPs), and Transformers to capture the dependencies between DFs and tackle the high dimensionality stemming from a high number of DFs interacting over time when making predictions. In a benchmark, we aim to quantify the benefits of using multivariate PMF, and specifically the use of state-of-the-art ML and DL time series techniques.

The rest of this paper is structured as follows. Section 2 discusses related work and introduces several state-of-the-art time series forecasting approaches. Section 3 gives a high-level overview of how PMF works. Section 4 introduces the data used in our benchmark and shows the model selection. The section also explains the benchmark setup and accompanying results. Next, Section 5 discusses the implications of these results and some of the limitations. Finally, Section 6 summarizes the main findings and provides some suggestions for future work.

2 Background and related work

2.1 Background of PMF

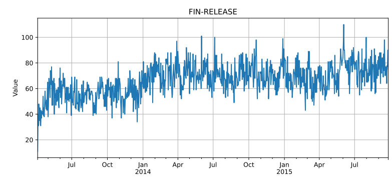
There has been a significant surge of interest in the exploration and application of predictive modeling techniques in process analytics. PMF moves from a case-level perspective towards process system-wide predictions. In an event log, directly-follows relations between activities can be calculated as counting functions for activity pairs over traces. Thus, a directly-follows graph can be obtained with the activities as nodes and DF relations as weighted edges. In this sense, the dynamics of a process model are expressed as the evolution of a DF Graph (DFG). By utilizing aggregations, the event log is split into multiple intervals, and the DFGs are constructed from every subset of the log. In PMF, the directly-follows time series over the time intervals are modeled and forecasted to build up a sequence of predicted DFGs, reflecting the long-term system-wide changes. To predict the DFs, [6] leverage Holt Winter's (HW) model, autoregressive model (AR), ARIMA model, GARCH model, and VAR models. Note that the first four models are classical univariate time series models, and VAR is a simple multivariate time series model that does not operate well on high-dimensional time series. The key distinction lies in whether considering the time series as separate single time-dependent variables (univariate) or multiple interrelated time-dependent variables together (multivariate). Thus, the univariate time series forecasting models overlook correlations between different DFs induced by the process structure. As the time series in actual applications become increasingly higher dimensional and complex, the importance of multivariate time series techniques becomes more important to capture the relationships and interactions between the different time series. In addition to VAR, as an extension of the AR model, multivariate time series models based on deep learning such as RNNs attract more attention recently, and are widely used in PPM [23]. Recently, other time series-based approaches to analyze these system-wide aspects of a process system have surfaced, such as [21] proposing a shift towards proactive and future-oriented business process management (BPM). [22] provide a generic approach to create time series abstractions of event logs. [24] present time-based conformance checking. [19] use a transition matrix with probabilities, possibly over time, to detect concept drift. The most similar work is [18], who demonstrate the effectiveness of temporal convolution networks on the prediction of work-in-progress (WiP) distributions in business processes. However, the predicted WiP is calculated as the total of all activities rather than on individual changes, resulting in insufficient granularity in describing and forecasting the process as PMF focuses on the joint prediction of the individual time series.

In the various DF time series selected from the event logs used in the evaluation section visualized in Figure 1, it can be seen that DF time series are typically not well-behaved and exhibit a variety of particular time series behavior. Sample 1a shows a common white noise serie with a trend and cycle, however, Sample 1b contains intermittency common to process systems such as weekends, resting periods, or activities not used throughout a process in combination with another (e.g. batching is happening, or resources are on holidays). Sample 1c shows similar low and high spikes which are typically hard to model with parametric (univariate) time series like ARIMA. Finally, Sample 1d shows the warm-up period of the system, which requires appropriate trimming of the time series. Given that these patterns can become complex and are often intertwined, e.g., Samples 1c through 1d are from the same system and can have been produced under similar resource schedules, it is necessary to use models that can cope with these irregularities which are not common to, e.g., econometric time series for which many typical time series techniques are tailored to. Below, we cover the most recent machine learning and deep learning approaches to tackle these.

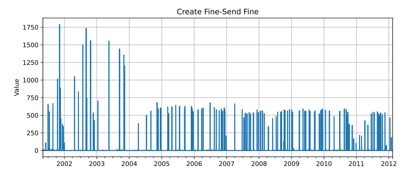
2.2 ML and DL time series Forecasting

Forecasting plays a crucial role in anticipating future trends by extrapolating time series data. The communities of data science and operations research have extensively researched time series forecasting by incorporating machine learning and deep learning techniques. In comparison to traditional forecasting methods, modern approaches often involve handling large sets of interconnected time series data, all of which require simultaneous forecasting. Gradient Boosting Decision Tree (GBDT) [9] is a widely-used machine learning algorithm due to its accuracy and interpretability, and Extreme Gradient Boosting (XGBoost) [4] and Light Gradient Boosting Machine (LightGBM) [12] were developed to be highly efficient and scalable. XGBoost grows trees level-wise and introduces regularization to prevent overfitting, while LightGBM grows trees leaf-wise, allowing deeper trees and better performance in some cases.

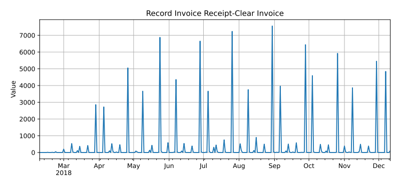
In deep learning, Recurrent Neural Networks (RNNs) are specifically designed to capture and learn patterns in sequential data such as time series. Connecting the hidden layers recurrently back to themselves enables the neural network to build



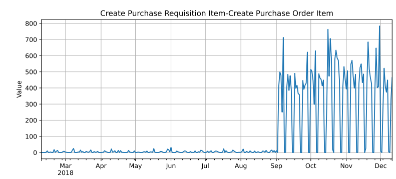
(a) Sample from the Hospital Billing log [17]



(b) Sample from the RTFMP log [14]



(c) Sample from the BPI 2019 [7]



(d) Sample from BPI 2019 [7]

Fig. 1: Sample of DF time series

memory. Furthermore, Long Short-Term Memory networks (LSTMs) [10] and a simplified version called Gated Recurrent Units (GRUs) [5] improve the gradient vanishment problem when revealing long-term dependencies. In process mining, LSTMs are widely applied in Predictive Process Monitoring (PPM) [25]. Besides RNNs, Multilayer Perceptrons (MLPs) as a simpler architecture are adopted for time series forecasting to achieve faster training and better generalization. N-BEATS [20] is designed specifically for forecasting tasks, relying on a structure of stacked MLPs. Each MLP block has a backward residual connection to improve the learning of the trend and seasonality components. Furthermore, N-HITS [3] introduces a hierarchical interpolation mechanism for multi-scale modeling of time series data, capturing different scales of information and features along the time axis. Unlike the general-purpose time series forecasting of N-BEATS, N-HITS provides better performance when time series data involves multiple temporal resolutions or significant short-term and long-term variations. Convolutional Neural Networks (CNNs) are tailored to handling input data such as images and time series A variation is dilated casual convolutions which are stacked on top of each other for efficient modeling of long-range dependencies in sequences. [2] propose a simple dilated casual convolution model, Temporal Convolutional Networks (TCNs), and provide empirical evidence showing its outperforming traditional recurrent models in many sequence modeling tasks.

Another recent architecture for time series forecasting is based on the attention mechanism [1] by focusing on specific parts of the input data and dynamically weighting different elements. The Transformers [27] are built entirely on the attention mechanism allowing parallelized training and a large number of parameters. Temporal Fusion Transformers (TFTs) [16] combine RNNs and attention mechanisms to capture both temporal dynamics and feature importance. In addition, the gating components in TFTs allow the model to skip irrelevant parts of the context, increasing flexibility and reducing the risk of overfitting. For long-term time series forecasting, DLinear and NLinear [29] were recently proposed as lightweight MLP model alternatives to

Transformers. DLinear decomposes a time series into trend and seasonal components and modeling these components separately using linear models, while NLinear directly applies a linear model to the raw time series data without separating trend and seasonality. [29] demonstrate that these linear models outperform more complex Transformer models for long-term time series forecasting on several benchmarks.

An effective variant of neural networks for prediction tasks is the family of Graph Neural Networks (GNNs), modeling network-based data like traffic networks [11]. GNNs allow to model both spatial and temporal dependency together for time series forecasting. For instance, [28] propose STGCN to tackle the time series prediction problem in the traffic domain with complete convolutional structures.

Given the various modeling capabilities of these multivariate models, a study into whether these multivariate correlations can be captured in DF time series, and how these relate to the characteristics of business processes underpinning various systems. This is done by an extensive benchmark by a wide range of the aforementioned techniques over a set of widely-used event logs.

3 Methodology

This work aims to perform DF forecasting using time series. More concretely, these are built by counting the occurrence of each DF in certain predefined timesteps (e.g. each day), determined by looking at the occurrence of the corresponding activity pairs in a process. Note that, for now, we assume atomic events, i.e., we only have a completion timestamp for each performed activity. In this way, we assume the activities taking a long time to finish and a long time to start are both reflected in a bottleneck in the DF. Figure 2 illustrates the transformation from event logs to DF time series. By dividing the event log into day-based intervals, a sequence of DFG matrices can be extracted from each subset of logs. The DFG matrices are flattened to one-dimensional DF vectors and stacked together in chronological order to obtain tabular time series data. However, a significant amount of DF relations in such a matrix never occur in the event log (as the activity pairs forming the DFs beginning- and end points are not present), so those are filtered out. Thus, the final tabular DF data excludes the empty DF columns and retains the observed DFs coinciding with the number of activity pairs present in the event log (of which there can be many) for prediction. To summarize, our approach takes an event log as input and predicts the DF time series as output. In the end, the predicted DFs support the construction of a DFG to represent the forecasted process model, which is not in the scope of this paper.

4 Experimental Evaluation

In this section, we give an overview of the data, its preprocessing, the models used, and finally we present the results of the benchmark.

4.1 Selected Data and Preprocessing

In the experiment, three publicly available event logs are used: BPI challenge of 2019 [7], a Hospital Billing event log [17], and a Road Traffic Fine Management

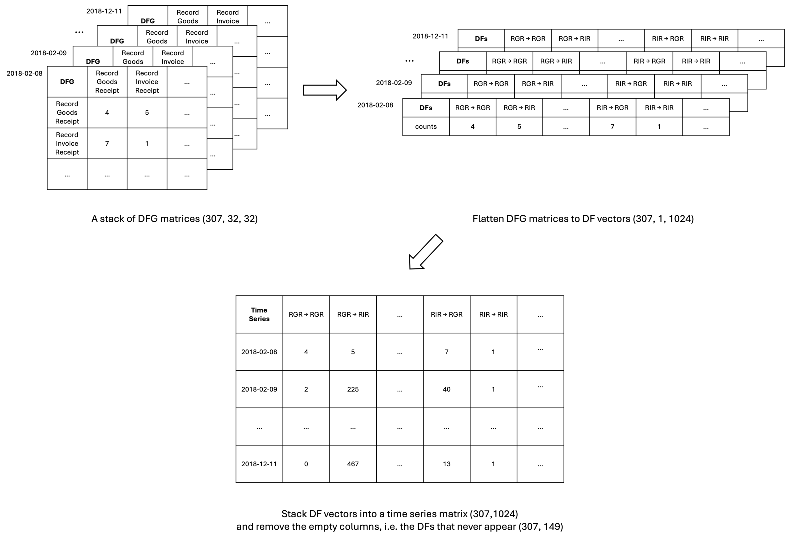


Fig. 2: Data transformation of BPI2019_1 [7] as an example, where "RGR" is short for "Record Goods Receipt" and "RIR" is short for "Record Invoice Receipt". The time step granularity is set at 1 day.

Process log (RTFMP) [14], covering a diverse set of processes. The BPI challenge of the 2019 event log contains four types of flows, and the used sub-log is the category of “3-way match, invoice before GR”, indicated as BPI2019_1. In the current experiments, to more accurately characterize process changes and provide insights for practical applications, we take timesteps of one day.

Table 1 describes three event logs and the preprocessing. Firstly, to only retain process behavior with enough signal in the event log, the infrequent variants are removed, and we retain the variants with a coverage percentage of 99.99% of the number of traces. Secondly, artificial “start” and “end” activities are added to the beginning and end of every case. Finally, the time lengths of the filtered event logs are reduced by trimming the first and last 10% of the time horizon as many systems have warm-up periods in which behavior is different, as illustrated in Figure 1d. By focusing on the steady-state part of processes, models can be trained on data that better represents typical operations, improving its generalization capabilities and predictive performance. Due to a large number of variants occurring rarely, the preprocessed event logs contain less than half of the variants but keep more than 90% traces. The activities “start” and “end” increase the number of activities and possible DFs after preprocessing.

4.2 Model Selection

As shown in section 2.1, the intricacies of the DF time series such as intermittency, long-distance dependencies and severe multicollinearity motivate the selection of time series forecasting approaches. The BPI2019_1 event log has a narrow time range (so fewer time steps) and may be trackable for ML-based models of XGBoost and

Table 1: Log Preprocessing

	BPI2019_1			Hospital Billing			RTFMP		
	original	preprocessed		original	preprocessed		original	preprocessed	
time range (days)	383	307	80%	1,132	906	80%	4,917	3,938	80%
# variants	7,835	740	9%	1,020	301	30%	231	106	46%
# traces	221,010	198,018	90%	100,000	90,604	91%	150,370	138,260	92%
# activities	39	32	82%	18	17	94%	11	13	118%
# events	1,234,494	1,301,182	105%	451,359	539,658	120%	561,470	720,625	128%
# unique DFs	413	149	36%	143	69	48%	70	39	56%
# DFs	1,013,484	1,103,164	109%	351,359	449,054	128%	411,100	582,365	142%

LightGBM due to their low training data requirements. The widely used RNNs and their capabilities of finding autoregressive and potentially longer-distance dependencies (in PPM) encourage us to extend them to PMF, including vanilla RNNs, LSTMs, and GRUs. Given the effectiveness of simple MLP architectures, N-BEATS is utilized for its robustness and N-HITS is included to capture the multi-scale seasonality, such as for the DFs in Figure 1b and 1c. TCNs have the advantage of forecasting business process changes by [18]. For long-term time series predictions like the RTFMP log, uncovering the dynamic trends involves using Transformers, TFTs, DLinear, and NLinear.

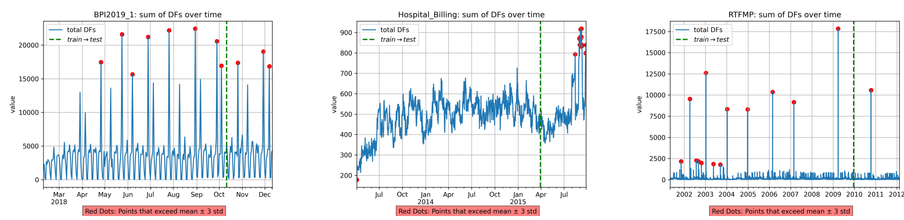
Therefore, the selected models include XGBoost, LightGBM, RNNs, LSTMs, GRUs, N-BEATS, N-HITS, TCNs, Transformers, TFTs, DLinear, and NLinear, compared with baselines of persistence, mean forecast, and linear regression. The persistence forecast, also known as naive forecast, takes the value of the last observed data point as the prediction for the future. The mean forecast predicts future values as the average of all past observations. The RNN, LSTM, and GRU have the same structure, with 2 hidden layers and 64 units in each. The N-BEATS and N-HITS have the same structure of 3 blocks with 4 hidden layers and 256 units on each. All the deep learning models are trained for 100 epochs.

GNN-based approaches, such as DCRNN [15], which would more directly use the process graph structure (e.g. the DFG), were omitted from this benchmark due to preliminary results indicating the data requirements are too strong for this type of problem (without significant architectural changes). Besides, the DFGs have time series on the edges instead of the nodes, which is not typical of GNNs.

4.3 Experimental Setup

First, the event logs are preprocessed by removing infrequent variants, adding "start" and "end" activities, and reducing the time length, as described in section 4.1. Then, the number of occurrences of each DF within the determined time step of one day is extracted as time series. To cover sufficient business days and provide strategic forecasting, which is the main aim of PMF being a system-wide forecasting exercise, we designed the model to learn the historical one-month data for predicting the future half month. For this experiment, different sample time series of 32 time steps (days) were used to forecast over an output horizon consisting of the next 16 days. Time series data is divided into training and test sets by a 4:1 ratio.

To visualize the overall evolution of DFs on three event logs, Figure 3 illustrates the summed DF time series, where the dotted green line represents the boundary between training and test sets and the red dots indicate the points exceeding the 3-sigma limit, i.e. the points that lie out of an interval 3 standard deviations removed from the mean. The high spikes in Figure 3a and 3c show the challenges for modeling. Note that BPI2019_1 appears to exhibit periodic patterns while RTFMP contains more intermittency. The majority of red dots in Figure 3b lie in the test set, posing challenges to learning these from the training data.



(a) BPI2019_1 (b) Hospital Billing (c) RTFMP

Fig. 3: Summed DFs Time Series Plots

We evaluate the forecasted horizon of 16 days in different ways. Firstly, we measure the mean absolute error (MAE) and root mean square error (RMSE) between the forecasted time series and the true values over the whole horizon. However, since forecasting one timestep or multiple (like 16) time steps ahead can be regarded as different problems we also evaluate the MAE and RMSE between forecast and true value for only the last timestep 16 days ahead, as this reflects the most difficult part of the horizon to forecast.

All preprocessing and models are implemented in Python with pm4py [1] and Darts [2] packages separately. The models' hyper-parameters are selected as the default settings in Darts. The code is publicly available[3].

4.4 Results

As mentioned earlier, the predictions are evaluated for all 16 time steps (all predictions of output sequences) as measured in Table 2 and the last time step (the last predictions of output sequences) as measured in Table 3 by mean absolute error (MAE) and root mean square error (RMSE). Note that the MAE and RMSE in Table 2 is the average error over the full horizon (16 days). The bottom lines in Table 2 and 3 show the number of evaluated output sequences and the number of predicted DFs in each sequence. The additional "Rank" columns indicate the prediction performance of different models according to the metrics of MAE or RMSE. The results show that

[1] https://pm4py.fit.fraunhofer.de
[2] https://unit8co.github.io/darts/index.html
[3] https://github.com/YongboYu/multi_PMF

different approaches rank vastly differently for different processes but XGBoost is best-performing in general. The boosting models produce more accurate predictions across all three datasets. However, the models of the RNN class perform poorly overall, especially on the BPI2019_1 log. N-HITS outperforms N-BEATS, potentially due to the capability of multi-scale time series techniques, but both achieve only moderate predictive capabilities. TCN stands out only on the RTFMP but loses the strength on the BPI2019_1. Transformer and TFT models have shown limited forecasting accuracy across the three datasets. DLinear and NLinear models demonstrate poor performance on the Hospital Billing and RTFMP, similar to the poor results of the linear regression.

Table 2: Evaluation for All 16 Output Predictions of All DFs

	BPI2019_1				Hospital Billing				RTFMP			
Model	MAE	Rank	RMSE	Rank	MAE	Rank	RMSE	Rank	MAE	Rank	RMSE	Rank
Persistence	26.80	5	47.66	10	2.47	2	2.99	2	3.75	3	6.89	3
Mean	25.76	4	41.53	6	**2.09**	**1**	**2.54**	**1**	3.48	2	**6.27**	**1**
LinearRegression	24.13	3	35.43	3	5.13	15	5.80	15	16.95	15	23.14	15
XGBoost	**17.99**	**1**	35.04	2	2.92	4	3.48	4	**3.22**	**1**	7.15	4
LightGBM	22.54	2	**33.34**	**1**	2.89	3	3.39	3	4.72	5	8.67	6
RNN	39.69	12	56.08	11	3.72	11	4.35	10	7.59	9	11.18	8
LSTM	49.22	15	60.99	15	3.54	9	4.19	9	9.04	10	13.94	10
GRU	45.15	14	58.06	12	3.78	12	4.45	12	11.97	12	16.48	12
N-BEATS	31.14	9	45.83	8	3.13	6	3.59	5	10.78	11	14.93	11
N-HiTS	27.87	6	40.75	5	3.12	5	3.59	5	6.87	8	10.56	7
TCN	41.63	13	60.35	14	3.34	7	3.84	7	3.93	4	6.79	2
Transformer	29.81	7	44.52	7	3.45	8	3.94	8	5.28	6	8.37	5
TFT	34.11	10	58.47	13	3.59	10	4.37	11	6.35	7	12.86	9
DLinear	35.84	11	46.97	9	4.56	13	5.16	13	15.64	14	21.02	14
Nlinear	30.06	8	40.59	4	4.84	14	5.50	14	15.02	13	20.31	13
(#seq, #DF)	(15, 149)				(134, 69)				(741, 39)			

Table 3: Evaluation for Last One Output Prediction of All DFs

	BPI2019_1				Hospital Billing				RTFMP			
Model	MAE	Rank	RMSE	Rank	MAE	Rank	RMSE	Rank	MAE	Rank	RMSE	Rank
Persistence	33.95	7	161.23	9	2.90	2	9.76	3	3.94	3	15.44	4
Mean	31.28	6	151.23	6	**2.28**	**1**	**7.54**	**1**	3.58	2	12.17	2
LinearRegression	28.09	3	140.83	4	5.70	15	18.10	15	17.18	15	56.79	15
XGBoost	**22.11**	**1**	**105.27**	**1**	3.01	3	8.93	2	**3.37**	**1**	12.46	3
LightGBM	24.94	2	120.39	2	3.25	5	10.15	6	4.74	5	16.16	5
RNN	33.96	8	167.09	10	4.11	12	12.50	11	9.10	9	25.74	9
LSTM	53.53	15	272.12	15	3.83	10	11.87	10	9.26	10	28.67	10
GRU	47.77	13	245.05	14	3.83	10	11.13	8	14.77	12	44.70	11
N-BEATS	40.92	12	205.09	12	3.28	6	10.30	7	13.41	11	46.32	12
N-HiTS	28.18	4	134.41	3	3.19	4	9.86	4	6.96	8	22.73	7
TCN	49.07	14	241.16	13	3.40	7	10.10	5	3.94	3	**11.42**	**1**
Transformer	34.34	11	159.63	8	3.73	9	12.57	12	6.27	6	20.23	6
TFT	34.16	9	172.24	11	3.67	8	11.41	9	6.63	7	24.32	8
DLinear	34.27	10	157.84	7	5.08	13	14.58	13	15.92	14	52.36	14
Nlinear	30.60	5	146.63	5	5.25	14	15.55	14	15.30	13	50.23	13
(#seq, #DF)	(15, 149)				(134, 69)				(741, 39)			

5 Discussion

Given the results in Section 4.4, it is not possible to definitively confirm whether multivariate approaches are superior. In some cases, even a naive forecast or a mean forecast can lead to only a small prediction error in terms of MAE and RMSE. On the other hand, two traditional ensemble models perform better than deep learning approaches overall, especially on the BPI2019_1 and Hospital Billing, which consist of shorter time lengths and higher dimensions of DF time series. In this section, we discuss the challenges of DF time series prediction in PMF for general forecasting models and multivariate deep learning models specifically.

In general, the evolution in the patterns of DF time series extracted from the event logs are challenging to capture and model. Different information systems tailor the timing of events and uniformly process them based on business content (i.e. batching). Thus, many DF pairs occur intermittently with exceptionally high values and some are present in clusters of a certain period, posing challenges for overall time series forecasting techniques. Concerning the setting up, the DF time series with the step of one day also introduces significant intermittency and fluctuations. A higher time aggregation level is possible to smooth the DF time series and enhance the predictive effect, however, this may again cause even sparser time series levels. In addition, the window size of 32 days might not be large enough to predict the time horizon of 16 days. Large fluctuations and long-term seasonality and trends in DF time series may require a longer window of inputs to learn and fit for more accurate predictions. The above three aspects impact the time series forecasting approaches including univariate and multivariate analysis techniques.

For multivariate deep learning approaches, we can identify several additional reasons for their poor performance in the experiments. First of all, event logs with short time ranges (a limited number of observations) are inadequate for training deep-learning models. Both time series and deep learning models require a large amount of data to fully learn complex multivariate relations and the fluctuations, seasonality, and trends in the time dimension. For example, TCN's performance ranks higher on the event logs with longer time ranges and fewer DFs. On the other hand, training traditional machine learning models is more manageable on small data sets (most commonly used event logs can be considered small). This leads to XGBoost and LightGBM reporting lower error rates in the overall experiments. Secondly, the diverse and widely fluctuating patterns of individual DF time series suggest that the relationships among them are also complex to reveal. The constructed multivariate connections might be insufficient and mislead the information propagated in the neural networks.

6 Conclusion and Future Work

In this paper, we propose data preprocessing for DF time series predictions in PMF. The various intricacies tied to DF time series such as intermittency, long-range dependencies, multi-scale presence, and so on, complicate the (multivariate) forecasting exercise, hence a wide range of techniques of different multivariate forecasting methods from machine learning and deep learning forecasting approaches with diverse architectures were benchmarked over three real-life event logs. XGBoost, as a traditional machine learning technique, performs better than deep learning-based models in general, meaning that the most sophisticated state-of-the-art fails to handle the DF time series' inherent complexity. In line with the several other insights and limitations raised in Section 5, several avenues for future work in PMF can be proposed.

For significant intermittency in the DF time series, specialized mechanisms such as [13] can be incorporated to boost the predictive performance. Even though GNNs are not covered in this paper, using the inherent capability to incorporate graph structures of process models would still be worth exploring. In addition to DFs from the control-flow aspect, forecasting process elements along other dimensions such as resource allocation, bottlenecks, and decision points could provide rich insights and enable proactive intervention. To generalize PMF, more event logs and process characters will be investigated.

Acknowledgements

This study was financed by the Research Foundation Flanders under grant number G039923N and Internal Funds KU Leuven under grant number C14/23/031.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
2. Bai, S., Kolter, J.Z., Koltun, V.: An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271 (2018)
3. Challu, C., Olivares, K.G., Oreshkin, B.N., Ramirez, F.G., Canseco, M.M., Dubrawski, A.: Nhits: Neural hierarchical interpolation for time series forecasting. In: Proceedings of the AAAI conference on artificial intelligence. vol. 37, pp. 6989–6997 (2023)
4. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. pp. 785–794 (2016)
5. Cho, K., Van Merriënboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259 (2014)
6. De Smedt, J., Yeshchenko, A., Polyvyanyy, A., De Weerdt, J., Mendling, J.: Process model forecasting and change exploration using time series analysis of event sequence data. Data & Knowledge Engineering **145**, 102145 (2023)
7. van Dongen, B.: Bpi challenge 2019 (2019). https://doi.org/10.4121/uuid:d06aff4b-79f0-45e6-8ec8-e19730c248f1, https://data.4tu.nl/articles/dataset/BPI_Challenge_2019/12715853/1
8. Evermann, J., Rehse, J.R., Fettke, P.: Predicting process behaviour using deep learning. Decision Support Systems **100**, 129–140 (2017)
9. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Annals of statistics pp. 1189–1232 (2001)
10. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**, 1735–1780 (1997)
11. Jiang, W., Luo, J.: Graph neural network for traffic forecasting: A survey. Expert systems with applications **207**, 117921 (2022)

12. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.Y.: Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems **30** (2017)
13. Kourentzes, N.: Intermittent demand forecasts with neural networks. International Journal of Production Economics **143**(1), 198–206 (2013)
14. de Leoni, M.M., Mannhardt, F.: Road Traffic Fine Management Process (2 2015). `https://doi.org/10.4121/uuid:270fd440-1057-4fb9-89a9-b699b47990f5`, `https://data.4tu.nl/articles/dataset/Road_Traffic_Fine_Management_Process/12683249`
15. Li, Y., Yu, R., Shahabi, C., Liu, Y.: Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. arXiv preprint arXiv:1707.01926 (2017)
16. Lim, B., Arık, S.Ö., Loeff, N., Pfister, T.: Temporal fusion transformers for interpretable multi-horizon time series forecasting. International Journal of Forecasting **37**(4), 1748–1764 (2021)
17. Mannhardt, F.: Hospital Billing - Event Log (8 2017). `https://doi.org/10.4121/uuid:76c46b83-c930-4798-a1c9-4be94dfeb741`, `https://data.4tu.nl/articles/dataset/Hospital_Billing_-_Event_Log/12705113`
18. Mehrdad Bibalan, Y., Far, B., Eshragh, F., Ghiyasian, B.: Work in progress prediction for business processes using temporal convolutional networks. In: International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems. pp. 109–121. Springer (2024)
19. Meira Neto, A.C., de Sousa, R.G., Fantinato, M., Peres, S.M.: Revisiting the transition matrix-based concept drift approach: Improving the detection task reliability through additional experimentation. SN Computer Science **5**, 188 (2024)
20. Oreshkin, B.N., Carpov, D., Chapados, N., Bengio, Y.: N-beats: Neural basis expansion analysis for interpretable time series forecasting. arXiv preprint arXiv:1905.10437 (2019)
21. Poll, R., Polyvyanyy, A., Rosemann, M., Röglinger, M., Rupprecht, L.: Process forecasting: Towards proactive business process management. In: Business Process Management: 16th International Conference, BPM 2018, Sydney, NSW, Australia, September 9–14, 2018, Proceedings 16. pp. 496–512. Springer (2018)
22. Pourbafrani, M., van der Aalst, W.M.: Extracting process features from event logs to learn coarse-grained simulation models. In: International Conference on Advanced Information Systems Engineering. pp. 125–140. Springer (2021)
23. Rama-Maneiro, E., Vidal, J.C., Lama, M.: Deep learning for predictive business process monitoring: Review and benchmark. IEEE Transactions on Services Computing **16**(1), 739–756 (2021)
24. Schuster, D., Schade, L., van Zelst, S.J., van der Aalst, W.M.: Temporal performance analysis for block-structured process models in cortado. In: International Conference on Advanced Information Systems Engineering. pp. 110–119. Springer (2022)
25. Tax, N., Verenich, I., La Rosa, M., Dumas, M.: Predictive business process monitoring with lstm neural networks. In: Advanced Information Systems Engineering: 29th International Conference, CAiSE 2017, Essen, Germany, June 12-16, 2017, Proceedings 29. pp. 477–492. Springer (2017)
26. Teinemaa, I., Dumas, M., Rosa, M.L., Maggi, F.M.: Outcome-oriented predictive process monitoring: Review and benchmark. ACM Transactions on Knowledge Discovery from Data (TKDD) **13**(2), 1–57 (2019)
27. Vaswani, A.: Attention is all you need. arXiv preprint arXiv:1706.03762 (2017)
28. Yu, B., Yin, H., Zhu, Z.: Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. arXiv preprint arXiv:1709.04875 (2017)
29. Zeng, A., Chen, M., Zhang, L., Xu, Q.: Are transformers effective for time series forecasting? In: Proceedings of the AAAI conference on artificial intelligence. vol. 37, pp. 11121–11128 (2023)