Process Model Forecasting Using Time Series Analysis of Event Sequence Data

Johannes De Smedt[1], Anton Yeshchenko[2], Artem Polyvyanyy[3],
Jochen De Weerdt[1], and Jan Mendling[4]

[1] KU Leuven, Leuven, Belgium
{johannes.desmedt;jochen.deweerdt}@kuleuven.be
[2] Vienna University of Economics and Business, Austria
{anton.yeshchenko;jan.mendling}@wu.ac.at
[3] The University of Melbourne, Melbourne, Australia
artem.polyvyanyy@unimelb.edu.au
[4] Humboldt-Universität zu Berlin, Berlin, Germany
jan.mendling@hu-berlin.de

Abstract. Process analytics is the field focusing on predictions for individual process instances or overall process models. At the instance level, various novel techniques have been recently devised, tackling next activity, remaining time, and outcome prediction. At the model level, there is a notable void. It is the ambition of this paper to fill this gap. To this end, we develop a technique to forecast the entire process model from historical event data. A forecasted model is a will-be process model representing a probable future state of the overall process. Such a forecast helps to investigate the consequences of drift and emerging bottlenecks. Our technique builds on a representation of event data as multiple time series, each capturing the evolution of a behavioural aspect of the process model, such that corresponding forecasting techniques can be applied. Our implementation demonstrates the accuracy of our technique on real-world event log data.

Keywords: Process model forecasting, predictive process modelling, process mining, time series analysis

1 Introduction

Process analytics is an area of process mining [1] focusing on predictions for individual process instances or overall process models. At the instance level, various novel techniques of predictive process monitoring (PPM) have been recently devised, tackling problems such as next activity, remaining cycle time, or other outcome predictions [6]. These techniques make use of neural networks [26], stochastic Petri nets [23], and general classification techniques [27].

At the model level, there is a notable void. Many analytical tasks require not only an understanding of the current as-is, but also the anticipated will-be process model. A key challenge in this context is the consideration of evolution

as processes are known to be subject to drift [15, 20, 31, 32]. A forecast can then inform the process analyst how the will-be process model might differ from the current as-is if no measures are taken, e.g., against emerging bottlenecks.

This paper presents the first technique to forecast whole process models. To this end, we develop a technique that builds on a representation of event data as multiple time series. Each of these time series captures the evolution of a behavioural aspect of the process model in the form of directly-follows relations (DFs), such that corresponding forecasting techniques can be applied for directly-follows graphs (DFGs). Our implementation on six real-life event logs demonstrates that forecasted models with medium-sized alphabets (10-30 activities) obtain below 10% mean average percentage error in terms of conformance. Furthermore, we introduce the Process Change Exploration (PCE) system which allows to visualise past and present models from event logs and compare them with forecasted models.

This paper is structured as follows. Section 2 discusses related work and motivates our work. Section 3 specifies our process model forecasting technique together with the PCE visualisation environment. Section 4 describes our evaluation, before Section 5 concludes the paper.

2   Related work and motivation

Within the field of process mining, research on and use of predictive modelling techniques has attracted plenty of attention in the last five years. PPM techniques are usually developed with a specific purpose in mind, ranging from next activity prediction [5, 26], over remaining time prediction [29], to outcome prediction [11]. For a systematic literature review of the field, we refer to [18]. Beyond the PPM field, this work is related to previous research on stage-based process mining [19], in which a technique is presented to decompose an event log into stages, and work on the detection of time granularity in event logs [22].

The shift from fine-granular PPM techniques, including next activity, remaining time, and outcome prediction, to model-based prediction, allows to obtain new insights into the global development of the process. Consider the example in Figure 1 where the road fine traffic management event log[5] is partitioned into 100 intervals in which an equal number of DF relations occur. The DFs in the first 50 intervals are used to predict the next 25 intervals. The DFGs show how process model forecasting and change exploration can provide multiple unique insights at a glance:

1. Compared to the initial 50 intervals the proportion of fines sent decreases in the later intervals;
2. The number of penalties added remains comparable between the first 50 and next 25 time intervals;
3. The number of occurrences and arc weights between *Create Fine* and *Send Fine* are predicted with reasonable error (+15%)

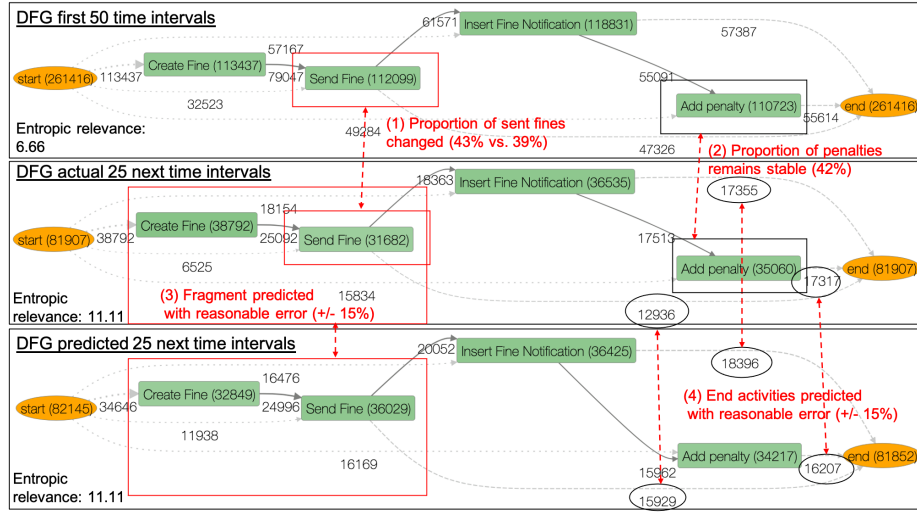[5]https://doi.org/10.4121/uuid:270fd440-1057-4fb9-89a9-b699b47990f5

Fig. 1: Directly-follows graphs of the 50 first intervals of the event log, as well as a forecasted and actual DFG of the 25 next intervals.

4. The arc weights of the ending activities are predicted with reasonable error (+15%)

These provide insight both in terms of the past en present model ((1)-(2)) and the quality of forecasts between the actual and forecasted model ((3)-(4)). Being able to construct such forecasts allows stakeholders to make estimates regarding how the overall fine system will evolve and allows to answer questions such as "How many more fines will be received?", "Will the backlog of fines be reduced?", "Will all fines be paid", and 'Will the ratio of unpaid fines stay the same?' This motivating example shows that, where process mining focuses on learning the as-is model to reason about trajectories of future cases and suggest potential repairs and improvements, process model forecasting allows to grasp the future stage of the overall process model in terms of a will-be model.

A suitable means to evaluate the forecasts quantitatively is entropic relevance [21]. This measure captures the quality of the discovered and forecasted DFGs with respect to the event logs they represent. Entropic relevance penalises the discrepancies in the relative frequencies of traces recorded in the log and described by the DFG. Entropic relevance stands for the average number of bits used to encode a log trace using the DFG, with small values being preferable to large ones. If the entropic relevance of the forecasted DFG and the actual future DFG with respect to the test log is the same, it suggests that both DFGs represent the future behaviour similarly well. The entropic relevance of the historical DFG derived from the training log with respect to the testing log is 6.66 as indicated in Figure 1, suggesting that the future behaviour shifts and the historical

DFG still represents the behaviour in the log better than both the actual and forecasted DFGs which sit at an entropic relevance of 11.11.

Measurement values are not enough to fully reveal the change of behaviour to the analyst. To this end, we complement the model-level prediction technique with a visualisation system to enable analysts to understand the forthcoming changes to the processes. Various process analysis tasks benefit from process forecasting [20], most notably process forecasting helps understanding the incremental changes and adaptations that happen to the process model and to project them into the future. In terms of visualisation principles, we follow the "Visual Information-Seeking Mantra": *overview first, zoom and filter, then details-on-demand* [25]. Thus, we expect the design of our visualisation system to assist in the following tasks:

T1. **Identify process adaptations:** The visualisation system should assist the user in identifying the changes that happen in the process model of the future in respect to the past;

T2. **Allow for interactive exploration:** The user should be able to follow the visual information-seeking principles, including overview first, filtering, zooming, and details-on-demand.

Forecasting entire process models provides a new perspective on predictive process monitoring. The prediction horizon is substantially longer as compared to what existing next-activity prediction models can achieve. Moreover, where next activity and related PPM techniques have a strong case-level focus, a prediction at the model level provides a more comprehensive picture of the future development of a process.

3   Process model forecasting

This section outlines how time series of directly-follows relationships are extracted from event logs as well as how they are used to obtain process model forecasts with a range of widely-used forecasting techniques. Finally, the visualisation of such predictions is introduced.

3.1   From event log to directly-follows time series

An event log L which contains the recording of traces $\sigma \in L$ produced by an information system during its execution and contains a sequence of events. Events in these traces are part of the power set over the alphabet of activities Σ which exist in the information system $\sigma = \langle e_1, ..., e_{|\sigma|} \rangle \subseteq \Sigma^*$. Directly follows relations between activities in an event log can be expressed as a counting function over activity pairs $>_L : \Sigma \times \Sigma \to \mathbb{N}$ with $>_L (a_1, a_2) = |\{e_n = a_1, e_{n+1} = a_2, \forall e_i \in L\}|$. Directly follows relations can be calculated on traces and subtraces in a similar fashion. A Directly-Follows Graph (DFG) of the process then exists as the weighted directed graph with the activities as nodes and their DF relations as weighted edges $DFG = (\Sigma, >_L)$.

In order to obtain predictions regarding the evolution of the DFG we construct DFGs for subsets of the log. Many aggregations and bucketing techniques exist for next-step, performance, and goal-oriented outcome prediction [26, 27, 19], e.g., predictions at a point in the process rely on prefixes of a certain length, or particular state aggregations [3]. In the proposed forecasting approach, we integrate concepts from time-series analysis. Hence, the evolution of the DFGs has to be monitored over intervals of the log where multiple aggregations are possible:

- **Equitemporal aggregation:** each sublog $L_s \in L$ contains a part of the event log of equal time duration. This can lead to sparsely populated sublogs when the events' occurrences are not uniformly spread over time, however, it is easy to apply (on new traces).
- **Equisized aggregation:** each sublog $L_s \in L$ contains a part of the event log where an equal amount of DFs occurred which leads to well-populated sublogs when enough events are available.

Tables 1 and 2 provide an example of both. These aggregations can be useful for the following reasons. First, equisized aggregation will have a higher likelihood of the underlying DFs approaching a white noise time series which is required for a wide range of time series techniques [9]. Second, both offer a different threshold at which the forecasting can be applied in practice. In the case of equisized aggregation, it is easier to quickly construct a desired number of intervals by simply dividing an event log to obtain equisized intervals. However, most time series techniques rely on the time intervals being of equal size which is embodied by the equitemporal aggregation [10]. Time series for the DFs $>_{T_{a_1,a_2}} = \langle >_{L_1} (a_1,a_2),\ldots,>_{L_s} (a_1,a_2)\rangle, \forall a_1,a_2 \in \Sigma \times \Sigma$ can be obtained for all activity pairs where $\bigcup_{L_1}^{L_s} = L$ by applying the aforementioned aggregations to obtain the sublogs.

Case ID	Activity	Timestamp
1	A1	11:30
1	A2	11:45
1	A1	12:10
1	A2	12:15
2	A1	11:40
2	A1	11:55
3	A1	12:20
3	A2	12:40
3	A2	12:45

Table 1: Example event log with 3 traces and 2 activities.

DF	Equitemporal	Equisized
$<_{L_s} (A1, A1)$	(0,1,0)	(1,0,0)
$<_{L_s} (A1, A2)$	(1,1,1)	(1,1,1)
$<_{L_s} (A2, A1)$	(0,1,0)	(0,1,0)
$<_{L_s} (A2, A2)$	(0,0,1)	(0,0,1)

Table 2: An example of using an interval of 3 used for equitemporal aggregation (75 minutes in 3 intervals of 25 minutes) and equisized intervals of size 2 (6 DFs over 3 intervals)).

3.2   From DF time series to process model forecasts

The goal of process model forecasting is to obtain a prediction for future DFGs by combining the predictions of all the DF pairs time series. To this purpose, we propose to use time series techniques to forecast $\widehat{DFG}_{T+h} = (\Sigma, \{\hat{>}_{T+h|T_{a_1,a_2}} |a_1, a_2 \in \Sigma \times \Sigma\})$ for which various algorithms can be used. In time series modelling, the main objective is to obtain a forecast or prediction $\hat{y}_{T+h|T}$ for a horizon $h \in \mathbb{N}$ based on previous T values in the series $(y_1, ..., y_T)$ [9]. For example, the naive forecast simply uses the last value of the time series T as its prediction $\hat{y}_{T+h|T} = y_T$. An alternative naive forecast uses the average value of the time series T as its prediction $\hat{y}_{T+h|T} = \frac{1}{T}\Sigma_i^T y_i$.

A trade-off exists between approaching DFGs as a multivariate collection of DF time series, or treating each DF separately. Traditional time series techniques use univariate data in contrast with multivariate approaches such as Vector AutoRegression (VAR) models, and machine learning-based methods such as neural networks or random forest regressors. Despite their simple setup, it is debated whether traditional statistical approaches are necessarily outperformed by machine learning methods. [16] found that this is not the case on a large number of datasets and note that the machine learning algorithms require significantly more computational power, a result that was later reaffirmed although it is noted that hybrid solutions are effective [17]. Especially for longer horizons, traditional time series approaches still outperform machine learning-based models. Given the potentially high number of DF pairs in a process' DFG, the proposed approach uses a time series algorithm for each DF series separately. VAR models would require a high number of intervals (at least as many as there are DFs times the lag coefficient) to be able to estimate all parameters of such a high number of time series despite their potentially strong performance [28]. Machine learning models could potentially leverage interrelations between the different DFs but again would require training set way larger than typically available for process mining to account for dimensionality issues due to the potentially high number of DFs. Therefore, in this paper, traditional time series approaches are chosen and applied to the univariate DF time series where these have at least one observation per sublog/time interval present.

We briefly discuss smoothing models, autoregressive, moving averages and ARIMA models, and varying variance models which make up the main families of traditional time series forecasting [9]. A wide array of such forecasting techniques exist, ranging from simple models such as naive forecasts over to more advanced approaches such as exponential smoothing and auto-regressive models. Many also exist in a seasonal variant due to their application in contexts such as sales forecasting.

A Simple Exponential Smoothing (SES) model uses a weighted average of past values where their importance exponentially decays as they are further into the past where Holt's models introduce a trend in the forecast, meaning the forecast is not flat. Exponential smoothing models often perform very well despite their simple setup [16]. AutoRegressive Integrating Moving Average (ARIMA) models are based on auto-correlations within time series. They combine auto-

regressions with a moving average over error terms. It is established by a combination of an AutoRegressive (AR) model of order p uses the past p values in the time series and applies a regression over them and a Moving Average (MA) model of order q which creates a moving average of the past forecast errors. Given the necessity of using a white noise series for AR and MA models, data is often differenced to obtain such series. ARIMA models then combine both AR and MA models where the integration is taking place after modelling as these models are fitted over differenced time series. ARIMA models are considered to be one of the strongest time series modelling techniques. An extension to ARIMA which is widely used in econometrics exists in (Generalized) AutoRegressive Conditional Heteroskedasticity ((G)ARCH) models [7]. They resolve the assumption that the variance of the error term has to be equal over time, but rather model this variance as a function of the previous error term. For AR-models, this leads to the use of ARCH-models, while for ARMA models GARCH-models are used as follows. An ARCH(q) model captures the change in variance by allowing it to both gradually increase over time, or to allow for short bursts of increased variance. A GARCH(p,q) model combines both the past values of observations as well as the past values of variance. (G)ARCH models often outperform ARIMA models in contexts such as the prediction of financial indicators of which the variance often changes over time [7].

3.3   Process change exploration

In Section 3.1 and Section 3.2 we described the approach for forecasting process models. To that end, gaining actual insights from such predicted values remains a difficult task for the analyst. This section sets off to present the design of a novel visualisation system to aid analysts in the exploration of the event logs and their corresponding (forecasted) discovered process models.

Following the user tasks T1 and T2 from Section 2, we designed a Process Change Exploration (PCE) system to support the interpretation of the process model forecasts. PCE is an interactive visualisation system that consists of three connected views.

Adaptation Directly-Follows Graph (aDFG) view. This is the main view of the visualisation that will show the model of the process. In order to accomplish task T1, we modify the DFG syntax. In order to display the process model adaptation from time range $T_{i_0} - T_{j_0}, i_0 < j_0$ to $T_{i_1} - T_{j_1}, i_1 < j_1$ we display the union of the process models of these regions, annotating the nodes and edges with the numbers of both ranges. We colour the aDFG as follows: we use colour saturation to show the nodes with higher values. We colour edges with a diverging saturation (red-black-green) schema. This colouring applies red colour to edges that are dominant in the $T_{i_0} - T_{j_0}$ range, and green if edges are dominant in the $T_{i_1} - T_{j_1}$ range, otherwise the edge is close to black. For coloring edges, we used idea of the three colour schema from [12].

Timeline view with brushed regions. This view represents the area chart graph that shows how the number of activity executions change with time. The area chart colour is split in two parts, one for the actual data, and the other

one to show the time range where values are predicted. Analysts can brush one region in order to zoom in, creating one region of interest $T_{i_0}, - T_{j_0}, i_0 < j_0$ that is displayed on the DFG. Analyst can also brush two regions of the area chart to select two time ranges, updating the DFG to aDFG representation. The brushed regions are coloured accordingly to the schema for colouring aDFG transitions. The earlier brushed region is coloured in red, while the second one is coloured green.

Activity and path sliders. We adopt two sliders that are used to simplify the DFG [13] and the aDFG for detailed exploration of the models.

Based on the described views, we conjecture that the analyst is able to accomplish the tasks T1, and T2 with ease.

4   Implementation and evaluation

In this section, an experimental evaluation over six real-life event logs is reported. The aim of the evaluation is to measure to what extent the forecasted DFG process models are capable of correctly reproducing actual future DFGs in terms of allowing for the same process model behaviour. To this end, we benchmark the actual against the forecasted entropic relevance discussed in Section 2. This is done at various parts of the trace, i.e. forecasts for the middle of the event logs up to the later parts of the event log to capture the robustness of the forecasting techniques in terms of the amount of data required to obtain good prediction results for both the equisized and equitemporal aggregation.

4.1   Re-sampling and test setup

To obtain training data, time series are obtained by specifying a number of intervals (i.e. time steps in the DF time series) using either equitemporal or equisized aggregation a described in Section 3.1. Time series algorithms are parametric and sensitive to sample size requirements [8]. Depending on the number of parameters a model uses, a minimum size of at least 50 steps is not uncommon, although typically model performance should be monitored at a varying number of steps. In the experimental evaluation, the event logs are divided into 100 time intervals with a varying share of training and test intervals. A constant and long horizon $h = 25$ is used meaning all test sets contain 25 intervals, but the training sets are varied from $ts = 25$ to $ts = 75$ intervals, meaning the forecasts progressively target the prediction of intervals 25-50 (the second quarter of intervals) over to 75-100 (the last quarter of intervals). This allows to both inspect the difference in results when only few data points are used, and whether there is a difference forecasting data points in the middle or towards the end of the available event data.

Resampling is applied based on a 10-fold cross-validation constructed following a rolling window approach for all horizon values $h \in [1, 25]$ where a recursive strategy is used to iteratively obtain $\hat{y}_{t+h|T_{t+h-1}}$ with $(y_1, \ldots, y_T, \ldots, \hat{y}_{t+h-1})$ [30]. 10 training sets are hence constructed for each training set length ts and

exist from $(y_1, \ldots, y_{T-h-f})$ and the test sets from $(y_{T-h-f+1}, \ldots, y_{T-f})$ with $f \in [0, 9]$ the fold index [4]. While direct strategies with a separate model for every value of h can be used as well and avoid the accumulation of error, they do not take into account statistical dependencies for subsequent predictions.

Three often-used, publicly-available event logs are used: the BPI challenge of 2012 log[6], 2017[7], and 2018[8], the Sepsis cases event log[9], an Italian help desk log[10], and the Road Traffic Fine Management Process log (RTFMP) event log (see Section 2). Each of these logs has a diverse set of characteristics in terms of case and activity volume, as well as average trace length, as can be seen in Table 3.

Event log	# cases	# activities	Average trace length
BPI 12	13,087	36	20.02
BPI 17	31,509	26	36.83
BPI 18	43,809	170	57.39
Sepsis	1,050	16	14.49
RTFMP	150,370	11	3.73
Italian	4,580	14	4.66

Table 3: Overview of the characteristics of the event logs used in the experimental evaluation.

There are a few considerations concerning the DF time series in these event logs. Firstly, DFs of activity pairs containing end point activities (i.e. at the start/end of a trace) often only contain meaningful numbers at very particular parts of the series and are hard to process by longitudinal algorithms which require a longer pattern to extract a meaningful pattern for prediction. Secondly, especially the equitemporal aggregation can suffer from event logs in which events do not occur as frequently throughout the full log's development. E.g., the Sepsis log's number of event occurrences tails off towards the end which can be alleviated by pre-processing (not done here to remain consistent over the event logs). Finally, if the level of occurrences of the DF pair is low and close to 0, the series might be too unsuitable for analysis with white noise series analysis techniques that assume stationarity. Ideally, every time series is tested using a stationarity test such as the Dickey-Fuller unit root test [14] and an appropriate lag order is established for differencing. Furthermore for each algorithm, especially ARIMA-based models, (partial) auto-correlation could establish the ideal p and q parameters. However, for the sake of simplicity and to avoid solutions where each activity pair has to have different parameters, various values are used for p, d, and q and applied to all DF pairs where only the best-performing are

---

[6]https://doi.org/10.4121/uuid:3926db30-f712-4394-aebc-75976070e91f

[7]https://doi.org/10.4121/uuid:5f3067df-f10b-45da-b98b-86ae4c7a310b

[8]https://doi.org/10.4121/uuid:3301445f-95e8-4ff0-98a4-901f1f204972

[9]https://doi.org/10.4121/uuid:915d2bfb-7e84-49ad-a286-dc35f063a460

[10]https://doi.org/10.4121/uuid:0c60edf1-6f83-4e75-9367-4c63b3e9d5bb

reported below for comparison with the other time series techniques. The results contain the best-performing representative of each forecasting family.

4.2   Results

All pre-processing was done in Python with a combination of *pm4py*[11] and the *statsmodels* package [24]. The code is available here[12].

In order to get a grasp of the forecasting performance in combination with the actual use of DFGs (which are seldom used in their non-aggregated form [2]) we present the mean absolute percentage error (MAPE) between the entropic relevance of the actual and forecasted DFGs at both full size, at 50%, and 75% reduction which is node-based (i.e. only the Q2/Q3 percentile of nodes in terms of frequency is retained). Using different levels of aggregation also balances recall and precision, as aggregated DFGs are less precise but possibly less overfitting. The results can be found in Tables 4 to 6. NAs are reported when the algorithms did not converge, no data was available (e.g. Sepsis for the 75-100 equitemporal intervals), or extremely high values were predicted.

When no reduction is applied, Table 4 shows that for the BPI12 and BPI17 logs a below 10% error can be achieved, mostly for equisized aggregation. For the Italian help desk log, results are in the 10-37% bracket, while for the other logs results are often well above a 100% deviation (with the entropic relavance of the actual DFGs being lower, hence better, than the entropic relevance of the forecasted DFGs). For the RTFMP and BPI18 log, however, results are better when more training points are used (e.g. 50 or 75 to obtain predictions for the 50-75 and 75-100 intervals). There is no strong difference between equisized and equitemporal aggregation except for the occasional outliers. Overall, the percentage error is lower in Table 5 when a reduction of 50% is applied with sub-10% results for the BPI12, Sepsis, and BPI17 logs. The results for the RTFMP log are occassionally better, but mostly worse, similar to BPI18. Finally, the results in Table 6 show a further reduction of errors for the BPI12, Sepsis, BPI17, and Italian logs, but also a drastic decrease to close to 0% for RTFMP. The results for the BPI18 log remain bad at over 100% error rates.

These results are commensurate the findings in [21] which contains entropic relevance results for the BPI12, Sepsis, and RTFMP logs indicating that entropic relevance of larger DFGs is lower (better) for RTFMP/Sespsis, and the entropic relevance goes up strongly for small models of RTFMP meaning the drastically improved error rates reported here are for models performing worse in terms of recall and precision. The entropic relevance for the BPI12 log is stable for the full spectrum of DFG sizes as per [21] which is reflected in the consistently good error rates presented here. This means that the low error rates reported are produced by the reduced DFGs which still score strongly in terms of recall and precision. Matching all results to the event log characteristics, we notice that the event logs with longer average traces with medium-sized alphabets (>20) such as

---

[11]https://pm4py.fit.fraunhofer.de

[12]https://github.com/JohannesDeSmedt/pmf

BPI12 and BPI17 consistently report good results. The BPI18 log's high number of activities seems to inflate error rates quickly, which is further aggravated when DFGs are reduced. Given that DFGs are based on activity pairs, this result is not surprising. For Sepsis and the Italian event logs good error rates are obtained once DFGs are reduced, indicating that forecasting the low-frequent edges and activities might lead to high error rates when the alphabet is smaller and traces are shorter, which is potentially also caused by the lack of precision as witnessed with the RTFMP log.

Overall, there exist many scenarios in which process model forecasting is delivering strong results. For the BPI12, BPI17, Italian, and Sepsis event logs, sub-10% error rates can be achieved both for equisized and equitemporal aggregation combined with model reductions which are typically applied by readers of DFGs. In some cases, even a naive forecast is enough to obtain a low error rate, however, the AR and ARIMA models report the best error rates in most cases. Nevertheless, results are often close except when fewer training points are used. Then, results are often varying widely. In future work, the robustness of the forecast algorithms will be further investigated, e.g., via scrutinising the confidence intervals of the predicted DF outcomes.

		BPI 12			Sepsis			RTFMP			BPI17			Italian			BPI18		
		50	75	100	50	75	100	50	75	100	50	75	100	50	75	100	50	75	100
equisize	nav	9.74	8.56	9.82	97.09	97.40	100.76	437.14	**105.81**	115.34	6.86	8.80	**7.00**	25.93	16.52	37.71	82.10	**99.90**	38.41
	arima212	12.41	9.75	10.80	NA	**83.31**	**100.58**	**398.66**	NA	NA	10.03	**8.54**	13.23	24.60	9.17	39.01	82.81	NA	30.12
	ar2	NA	**8.45**	**9.62**	**97.04**	97.40	100.76	NA	NA	**110.14**	6.83	14.84	13.83	23.81	13.98	**36.89**	**78.82**	NA	NA
	hw	**8.61**	8.96	10.14	97.09	97.40	100.76	402.83	110.17	130.10	**6.81**	8.68	186.94	**22.54**	**9.14**	43.31	81.04	NA	NA
	garch	11.47	8.60	10.17	97.09	97.40	100.76	426.71	109.79	117.15	6.89	8.82	186.94	25.48	31.29	65.54	72.89	NA	**28.59**
equitemp	nav	15.57	10.14	12.63	98.51	100.75	NA	199.69	29.70	36.15	7.12	8.63	**13.41**	27.12	26.86	39.94	NA	NA	54.57
	arima212	NA	11.67	**12.00**	**89.07**	**100.39**	NA	**122.63**	**28.55**	**33.82**	8.13	158.70	18.74	26.59	24.26	38.03	NA	**42.83**	NA
	ar2	NA	**9.97**	12.43	98.37	100.75	NA	NA	29.74	NA	7.09	NA	19.60	**26.33**	30.02	38.68	NA	NA	NA
	hw	**13.09**	10.46	12.08	98.40	100.75	NA	162.94	29.34	36.15	**7.07**	**8.35**	186.91	26.90	**23.57**	**36.20**	NA	43.02	NA
	garch	17.80	10.29	12.71	95.75	100.75	NA	199.13	30.44	36.00	7.37	187.45	186.91	27.11	45.58	55.67	NA	46.69	**42.97**

Table 4: Overview of the mean percentage error in terms of entropic relevance for the full DFGs.

		BPI 12			Sepsis			RTFMP			BPI17			Italian			BPI18		
		50	75	100	50	75	100	50	75	100	50	75	100	50	75	100	50	75	100
equisize	nav	**4.65**	**5.83**	**11.50**	**8.35**	8.80	6.29	234.18	295.99	203.68	7.82	9.22	11.04	23.05	14.18	21.66	**252.76**	**231.44**	**160.66**
	arima212	7.96	22.89	13.43	8.55	8.81	**6.14**	234.14	288.27	198.86	4.49	5.98	10.67	22.31	7.32	23.17	369.47	252.24	218.93
	ar2	24.53	27.58	30.81	8.54	8.72	6.30	234.57	293.10	201.21	**4.27**	6.08	**10.22**	21.26	11.91	**20.56**	NA	230.02	NA
	hw	45.80	38.02	13.13	8.73	**8.65**	6.17	233.05	**151.19**	**111.89**	4.51	**5.37**	11.06	**20.33**	7.28	26.12	391.38	280.59	226.27
	garch	26.30	23.77	48.86	8.63	8.87	7.06	**231.70**	295.93	203.45	4.50	9.41	11.09	23.18	29.07	45.31	315.99	234.79	217.62
equitemp	nav	**7.15**	**6.86**	17.86	6.41	**7.73**	NA	**75.48**	36.18	86.46	5.93	7.23	30.98	24.35	18.64	26.48	NA	**219.13**	410.13
	arima212	49.87	10.59	19.59	**4.91**	8.13	NA	135.97	40.22	86.74	**5.64**	**5.13**	**30.30**	**23.48**	16.32	**20.45**	**205.21**	261.81	**253.38**
	ar2	21.06	7.49	18.85	7.17	8.08	NA	95.44	36.30	86.60	5.70	NA	30.97	23.67	21.71	25.44	NA	NA	443.76
	hw	7.41	7.02	**17.54**	6.72	10.34	NA	77.93	36.62	86.76	5.95	7.39	30.86	23.55	**15.66**	22.91	NA	236.37	439.04
	garch	57.44	32.85	37.98	6.76	7.85	NA	**75.48**	36.20	86.52	5.93	7.33	31.12	24.24	36.51	40.40	NA	283.40	492.85

Table 5: Overview of the mean percentage error in terms of entropic relevance for the DFGs with a 50% reduction.

		BPI 12			Sepsis			RTFMP			BPI17			Italian			BPI18	
		50	75	100	50	75	100	50	75	100	50	75	100	50	75	100	50	75	100
equisize	nav	0.96	0.87	1.37	2.40	2.91	**3.47**	0.00	0.00	0.01	0.12	**0.29**	**0.14**	13.11	15.11	27.02	**247.32**	223.34	**146.27**
	arima212	0.98	**0.84**	**1.33**	**2.38**	2.85	3.65	0.00	0.00	0.01	**0.06**	0.30	0.16	13.08	14.94	26.97	346.94	236.73	211.17
	ar2	**0.86**	0.85	1.35	2.42	2.58	**3.47**	0.00	0.00	0.01	0.12	0.30	**0.14**	13.10	15.02	26.97	NA	**222.06**	NA
	hw	1.00	0.85	1.35	2.73	2.90	**3.47**	0.00	0.00	0.01	0.11	0.31	0.15	**12.98**	**14.79**	27.08	333.80	255.83	207.86
	garch	0.89	0.86	1.35	2.51	**2.49**	**3.47**	0.00	0.00	0.01	0.12	**0.29**	**0.14**	13.13	15.00	**26.89**	299.59	248.75	182.07
equitemp	nav	4.92	3.55	4.05	**2.31**	**1.65**	NA	0.03	0.02	0.11	**0.05**	**0.11**	5.92	8.18	30.10	20.61	NA	**203.92**	562.09
	arima212	4.93	3.59	3.86	2.77	2.62	NA	0.03	0.02	0.11	0.11	0.14	**5.82**	8.39	30.20	20.54	**180.36**	245.18	**191.25**
	ar2	4.85	**3.52**	4.04	2.35	2.93	NA	0.03	0.02	0.11	0.06	NA	5.91	8.19	30.17	20.58	NA	NA	559.14
	hw	**4.82**	**3.52**	3.84	2.47	1.67	NA	0.03	0.02	0.11	0.06	0.13	5.85	8.45	**20.00**	**15.90**	NA	228.78	384.37
	garch	7.97	3.54	4.02	2.34	2.92	NA	0.03	0.02	0.11	**0.05**	0.12	5.93	**8.17**	30.03	20.52	NA	226.58	606.03

Table 6: Overview of the mean percentage error in terms of entropic relevance for the DFGs with a 75% reduction.

4.3   Visualising Process Model Forecasts

In Section 4.2 we evaluated forecasting results, ensuring the conformance and interpretability of the predicted process models. To that end, gaining actual insights from such predicted data remains a difficult task for the analyst. This section sets off to present the implementation of a novel visualisation system to aid analysts in exploration of the event logs. The process of designing and implementing the system started by designing several prototypes that undergone rounds of discussions to mature into the implemented visualisation system.

The design of the PCE system is shown in Figure 2. It shows an interactive visualisation system with several connected views. The system is implemented with D3.js JavaScript library and is available as an open source project[13].
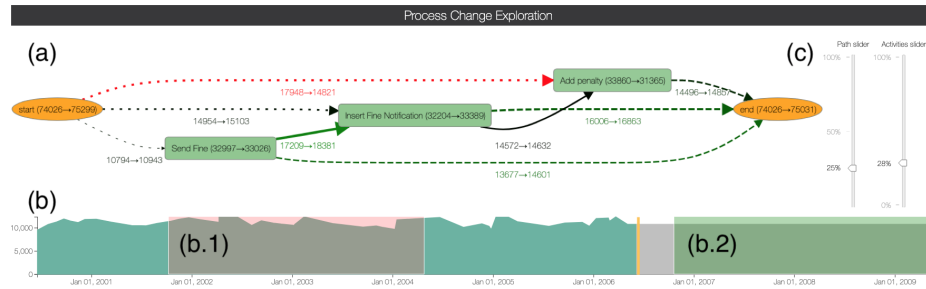

Fig. 2: Process Change Exploration (PCE) system. *(a)* shows *Adaptation Directly-Follows Graph (aDFG)* view. *(b)* shows the *Timeline view with brushed regions* view. Users can brush one or more regions on this graph in order to filter the scope of the analysis *(b.1, and b.2)*. Two additional views on *(c)* show the *activity and path sliders.*

---

[13]See https://github.com/yesanton/Process-Change-Exploration-Visualizations

5   Conclusion

In this paper, we presented the first genuine approach to forecast a process model as a whole. To this end, we developed a technique based on time series analysis of DF relations to forecast entire DFGs from historical event data. In this way, we are able to make accurate forecasts regarding the future development of the process, including whether process drifts might occur in particular parts of the process. This is supported by the Process Change Exploration system which allows analysts to compare various parts of the past, present, and forecasted future behaviour. Our empirical evaluation demonstrates that, most notably for reduced process models with medium-sized alphabets, we can obtain below 10% MAPE in terms of conformance to the true model for event logs with medium-sized alphabets.

In future research, we plan to evaluate alternative machine learning techniques for process model prediction. Furthermore, we want to explore opportunities for enriching our forecasted process models with confidence intervals. Finally, we will conduct design studies with process analysts to evaluate the usability of different visualisation techniques.

References

1. van der Aalst, W.: Data Science in Action. In: Process Mining. Springer (2016)
2. van der Aalst, W.M.P.: A practitioner's guide to process mining: Limitations of the directly-follows graph. Procedia Computer Science 164, 321–328 (2019)
3. van der Aalst, W.M.P., Rubin, V.A., Verbeek, H.M.W., van Dongen, B.F., Kindler, E., Günther, C.W.: Process mining: a two-step approach to balance between underfitting and overfitting. Softw. Syst. Model. 9(1), 87–111 (2010)
4. Bergmeir, C., Benítez, J.M.: On the use of cross-validation for time series predictor evaluation. Information Sciences 191, 192–213 (2012)
5. Evermann, J., Rehse, J.R., Fettke, P.: Predicting process behaviour using deep learning. Decision Support Systems 100, 129–140 (2017)
6. Francescomarino, C.D., Ghidini, C., Maggi, F.M., Milani, F.: Predictive process monitoring methods: Which one suits me best? In: BPM. Lecture Notes in Computer Science, vol. 11080, pp. 462–479. Springer (2018)
7. Francq, C., Zakoian, J.M.: GARCH models: structure, statistical inference and financial applications. John Wiley & Sons (2019)
8. Hanke, J.E., Reitsch, A.G., Wichern, D.W.: Business forecasting, vol. 9. Prentice Hall New Jersey (2001)
9. Hyndman, R.J., Athanasopoulos, G.: Forecasting: principles and practice. OTexts (2018)
10. Kil, R.M., Park, S.H., Kim, S.: Optimum window size for time series prediction. In: 19th Annual Int. Conf. IEEE Engineering in Medicine and Biology Society. vol. 4, pp. 1421–1424. IEEE (1997)
11. Kratsch, W., Manderscheid, J., Röglinger, M., Seyfried, J.: Machine learning in business process monitoring: A comparison of deep learning and classical approaches used for outcome prediction. Business & Inf. Systems Engineering pp. 1–16 (2020)

12. Kriglstein, S., Rinderle-Ma, S.: Change visualizations in business processes - requirements analysis. In: GRAPP/IVAPP. pp. 584–593. SciTePress (2012)
13. Leemans, S., Poppe, E., Wynn, M.: Directly follows-based process mining: A tool. In: Proceedings of the ICPM Demo Track 2019. pp. 9–12 (2019)
14. Leybourne, S.J., et al.: Testing for unit roots using forward and reverse dickey-fuller regressions. Oxford Bulletin of Economics and Statistics 57(4), 559–571 (1995)
15. Maaradji, A., Dumas, M., La Rosa, M., Ostovar, A.: Detecting sudden and gradual drifts in business processes from execution traces. IEEE Transactions on Knowledge and Data Engineering 29(10), 2140–2154 (2017)
16. Makridakis, S., Spiliotis, E., Assimakopoulos, V.: Statistical and machine learning forecasting methods: Concerns and ways forward. PloS one 13(3), e0194889 (2018)
17. Makridakis, S., Spiliotis, E., Assimakopoulos, V.: The m4 competition: 100,000 time series and 61 forecasting methods. Int. Journal of Forecasting 36(1), 54–74 (2020)
18. Neu, D.A., Lahann, J., Fettke, P.: A systematic literature review on state-of-the-art deep learning methods for process prediction. Artificial Int. Review pp. 1–27 (2021)
19. Nguyen, H., Dumas, M., ter Hofstede, A.H., La Rosa, M., Maggi, F.M.: Business process performance mining with staged process flows. In: International Conference on Advanced Information Systems Engineering. pp. 167–185. Springer (2016)
20. Poll, R., Polyvyanyy, A., Rosemann, M., Röglinger, M., Rupprecht, L.: Process forecasting: Towards proactive business process management. In: BPM. Lecture Notes in Computer Science, vol. 11080, pp. 496–512. Springer (2018)
21. Polyvyanyy, A., Moffat, A., García-Bañuelos, L.: An entropic relevance measure for stochastic conformance checking in process mining. In: ICPM. pp. 97–104 (2020)
22. Pourbafrani, M., van Zelst, S.J., van der Aalst, W.M.P.: Semi-automated time-granularity detection for data-driven simulation using process mining and system dynamics. In: Conceptual Modeling. pp. 77–91. Cham (2020)
23. Rogge-Solti, A., Weske, M.: Prediction of Remaining Service Execution Time Using Stochastic Petri Nets with Arbitrary Firing Delays. In: ICSOC. Lecture Notes in Computer Science, vol. 8274, pp. 389–403. Springer (2013)
24. Seabold, S., Perktold, J.: Statsmodels: Econometric and statistical modeling with python. In: 9th Python in Science Conference. vol. 57, p. 61 (2010)
25. Shneiderman, B.: The eyes have it: A task by data type taxonomy for information visualizations. In: VL. pp. 336–343. IEEE Computer Society (1996)
26. Tax, N., Verenich, I., Rosa, M.L., Dumas, M.: Predictive business process monitoring with LSTM neural networks. In: CAiSE. LNCS, vol. 10253, pp. 477–492 (2017)
27. Teinemaa, I., Dumas, M., Rosa, M.L., Maggi, F.M.: Outcome-oriented predictive process monitoring: Review and benchmark. ACM Trans. Knowl. Discov. Data 13(2), 17:1–17:57 (2019)
28. Thomakos, D.D., Guerard Jr, J.B.: Naive, arima, nonparametric, transfer function and var models: A comparison of forecasting performance. International Journal of Forecasting 20(1), 53–67 (2004)
29. Verenich, I., Dumas, M., Rosa, M.L., Maggi, F.M., Teinemaa, I.: Survey and cross-benchmark comparison of remaining time prediction methods in business process monitoring. ACM Transactions on Intelligent Systems and Technology (TIST) 10(4), 1–34 (2019)
30. Weigend, A.S.: Time series prediction: forecasting the future and understanding the past. Routledge (2018)

31. Yeshchenko, A., Di Ciccio, C., Mendling, J., Polyvyanyy, A.: Comprehensive process drift detection with visual analytics. In: International Conference on Conceptual Modeling. pp. 119–135. Springer (2019)
32. Yeshchenko, A., Di Ciccio, C., Mendling, J., Polyvyanyy, A.: Visual drift detection for sequence data analysis of business processes. IEEE Transactions on Visualization and Computer Graphics (2021)