# Process Model Forecasting and Change Exploration Using Time Series Analysis of Event Sequence Data

Johannes De Smedt[a], Anton Yeshchenko[b], Artem Polyvyanyy[c], Jochen De Weerdt[a], Jan Mendling[d]

[a] *Research Centre for Information Systems Engineering, KU Leuven, Leuven, Belgium*
[b] *Vienna University of Economics and Business, Austria*
[c] *The University of Melbourne, Victoria, 3010, Australia*
[d] *Humboldt-Universität zu Berlin, Berlin, Germany*

**Abstract**

Process analytics is a collection of data-driven techniques for, among others, making predictions for individual process instances or overall process models. At the instance level, various novel techniques have been recently devised, tackling analytical tasks such as the next activity, remaining time, or outcome prediction. However, there is a notable void regarding predictions at the process model level. It is the ambition of this article to fill this gap. More specifically, 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 description of the overall process for a given period in the future. Such a forecast helps, for instance, to anticipate and prepare for the consequences of upcoming process drifts 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 time series forecasting techniques can be applied. Our implementation demonstrates the feasibility of process model forecasting using real-world event data. A user study using our Process Change Exploration tool confirms the usefulness and ease of use of the produced process model forecasts.

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

## 1. Introduction

The growth in the use of information systems has fuelled a wide range of data analysis techniques that intend to describe and improve their inner workings.

*Email addresses:* johannes.desmedt@kuleuven.be (Johannes De Smedt), anton.yeshchenko@wu.ac.at (Anton Yeshchenko), artem.polyvyanyy@unimelb.edu.au (Artem Polyvyanyy), jochen.deweerdt@kuleuven.be (Jochen De Weerdt), jan.mendling@hu-berlin.de (Jan Mendling)

Postprint, January 2023

Process mining [1] is a fast-growing field in information systems analysis that encompasses a wide range of techniques performed on event data generated by these systems, including the visualization, conformance checking, and enhancement of process models that implement business processes in organizations [1]. Process analytics is a subarea that encompasses Predictive Process Monitoring (PPM) aimed at making predictions for individual process instances or overall process models [1]. Many PPM techniques have surfaced to support the prediction of the next activity in the process, the remaining process cycle time, and other goal-oriented process outcomes [2]. These techniques make use of various predictive architectures, including neural networks [3], stochastic Petri nets [4], and general classification techniques [5].

PPM techniques, however, typically focus on a short time horizon or a well-scoped outcome in terms of prediction [5]. Indeed, it is known that the next activity prediction techniques often perform poorly on long time horizons [6]. This, consequently, limits the range of insights that can be obtained. A process (model) can change over time, for example, as a response to new regulations, customer demands, and novel ways of supporting business processes. While a process outcome can be predicted accurately, this might still obfuscate the underlying drivers for that outcome. Process analysts could benefit from obtaining a more evolutionary image of the process design [7], including the stability or change of (parts of) the process model, such as process drifts, that can inform improvement ideas [8]. At the model level, there is a notable void in terms of predictive analytics. Many process analysis tasks such as identifying bottlenecks, planning major changes to process-aware information systems, and so on, require an understanding of the current as-is and the anticipated will-be processes. A key challenge in this context is the consideration of evolution as processes are known to be subject to change [9, 7, 10, 11]. A forecast can then inform the process analyst how the will-be processes differ from the current as-is processes, thus providing input for decisions on improving the future processes.

This article presents a technique to forecast a process model, a description of the will-be processes. To this end, we develop an algorithm 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 time series forecasting techniques can be applied. The DFs are used widely in process mining as components of Directly-Follows Graphs (DFGs), which are semi-formal process models, often appealing to practitioners [12]. The latter are widely used in process mining as a representation for processes and hence are a type of process model, albeit one without clear-cut execution semantics [12]. Our implementation on six real-life event logs demonstrates that forecasted models with medium-sized alphabets (10-30 activities) obtain below 15% mean average percentage error in terms of conformance of forecasted processes, outperforming the proposed baselines. Furthermore, we introduce the Process Change Exploration (PCE) system which allows to visualise past and present models discovered from event logs and compare them with forecasted models. In a user study, we test and confirm its perceived ease-of-use and usefulness.

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

In the field of process mining, research on and use of predictive modelling techniques has attracted much attention in the last five years. PPM techniques are usually developed with a specific purpose in mind, ranging from the next activity prediction [13, 3], over remaining time prediction [14], to outcome prediction [15]. For a systematic literature review of the field, we refer the reader to [16]. Beyond the PPM field, this work is related to previous research on stage-based process mining [17], in which a technique is presented to decompose an event log into stages, and work on the detection of time granularity in event logs [18].

The shift from fine-granular PPM techniques, including next activity, remaining time, and outcome prediction, to model-based prediction allows obtaining new insights into the global development of the process. Consider the example of the sepsis event log[1] shown in Figure 1. Here it is partitioned into 100 intervals in which an equal number of DF relations occur (referred to below as 'equisized' aggregation, see Section 3.2). The figure contains three subfigures: The top one shows a DFG constructed over 2 months from the first 50 intervals of the event log used as a training set to establish the as-is state during the first half of the system's execution in terms of frequency. The middle shows a DFG constructed over 2 months from the next 25 intervals used as a test set showing the actual state of the information system after those initial 50 intervals. The bottom of the figure shows the forecasted DFG through predictions of its separate DFs over the same 2 months (forecast horizon of 25) using a GARCH model (see Section 3.3). Typical process discovery techniques establish visualisations over the full event log, i.e., all 100 intervals, while predictive process modelling techniques typically use increasingly long prefixes of the individual historical traces to train predictive models using varying amounts of historical information. The proposed approach allows to combine different historical intervals over all aggregated historical cases to obtain process model forecasting in the bottom figure, which brings the following unique insights at a glance:

1. The relative frequencies of major activities and DF relations in the DFGs do not change after the first 50 intervals (actual vs. actual).

2. The forecast errors vary between 21% and 28% for the overall activity frequencies in the DFGs (actual vs. forecasted).

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

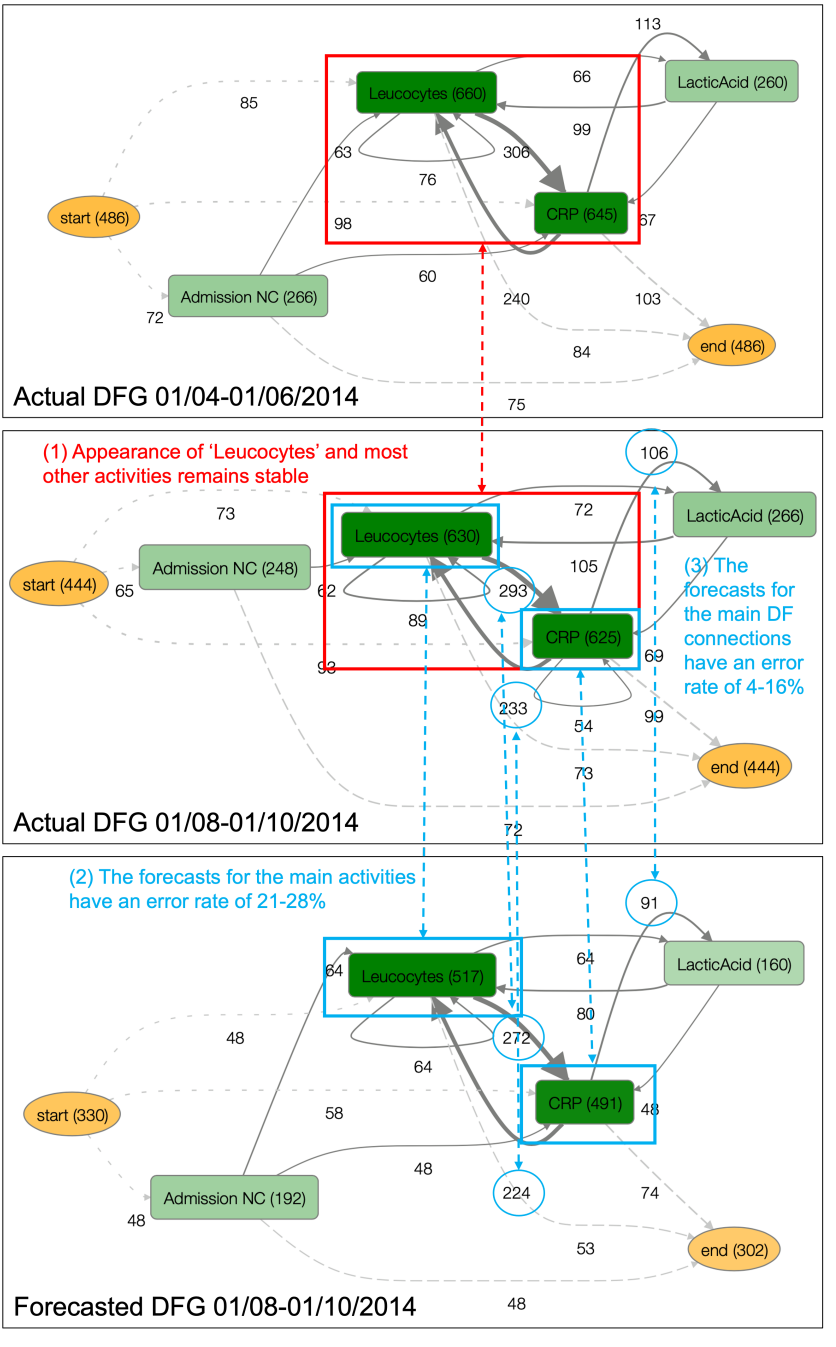



Figure 1: Directly-follows graphs of the 50 first intervals of the event log, as well as a forecasted and actual DFG of the 25 next intervals.

3. The forecast errors for the individual DF relations are lower and vary between 4% and 16% (actual vs. forecasted).

These results provide insight both in terms of the past and present model, refer to (1), and the quality of the forecasts between the actual and forecasted model, refer to (2)–(3). Being able to construct such forecasts allows stakeholders to make estimates regarding how the overall business process and the corresponding information system will evolve and allows them to answer questions such as "Will the number of admitted patients be stable?", "Will there be more patients referred to a leucocytes test after a C-Reactive Protein (CRP) test?", and "Will as many cases reach the end state as before?". These questions can be used to evaluate the overall behaviour of the system and understand where bottlenecks might arise, activity relations have changed, or the process might have changed overall.

This motivating example shows that, where process mining in terms of discovery focuses on learning the as-is model to reason about trajectories of future cases and suggest potential repairs and improvements, process model forecasting allows us to grasp the future state of the full process model in terms of a will-be model.

Note that, in this example, the aggregated forecasts in the form of activity frequency prediction (aggregated by calculating the sum of all incoming DF relations) have higher error rates. To address this issue, an appropriate evaluation measure for model-wide evaluation is necessary. A suitable means to evaluate the forecasts quantitatively is entropic relevance [19]. 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 as it 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, then both DFGs represent the future behaviour similarly well. The entropic relevance of the actual DFG over the 25 intervals of the test set has a value of 23.12, while the forecasted DFG over the test set has an entropic relevance of 25.35. This means that the forecasted DFG requires 10% more bits to encode the information of the event log with the forecasted model, which means it has a 10% percentage error and gives a quantitative, model-wide conformance evaluation.

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 [7]; most notably process forecasting helps understanding the incremental changes and adaptations that happen to the process model and to project them into the future. Given the proposed approach focuses on control flow, these changes are typically driven by process owners aiming at achieving various properties such as runtime adaption [20], context adaptation [21],

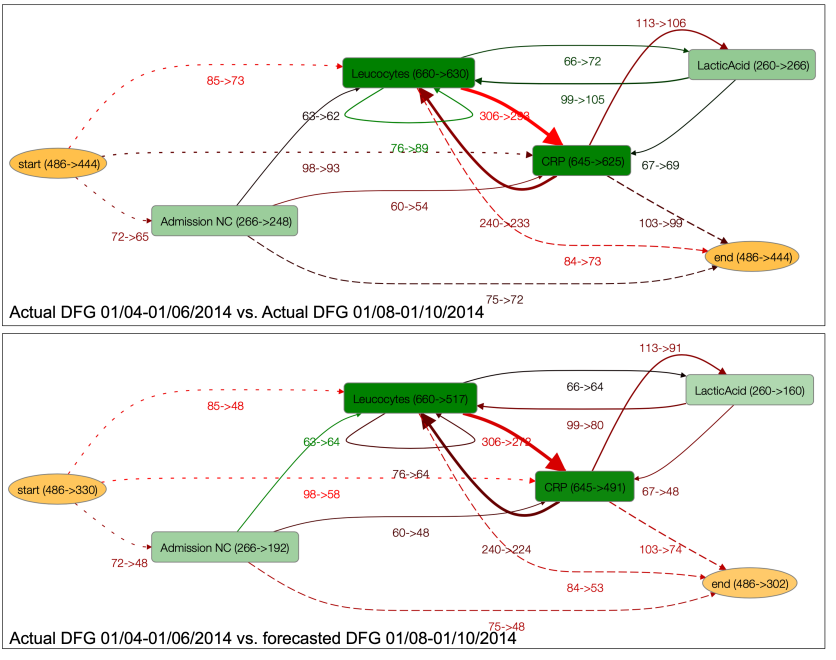



Figure 2: Directly-follows graphs of the 50 first intervals of the event log, as well as a forecasted and actual DFG of the 25 next intervals.

or flexibility [22]. In terms of visualisation principles, we follow the "Visual Information-Seeking Mantra": *overview first, zoom and filter, then details-on-demand* [23]. 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 with 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.

Figure 2 illustrates how the proposed solution in the form of the Process Change Exploration (PCE) tool of Section 4.4 allows to address T1 by automatically visualising the differences between the top and middle, and middle and bottom DFGs from Figure 1. Red arrows indicate how the older model's DFs have decreased, while green arrows indicate how the older model's DFs have increased. For example, the number of DF occurrences between activities 'Leucocytes' and 'LacticAcid' increased from 66 to 72 (green arrow $66 \rightarrow 72$) between 01/04 and 01/08 (top of Figure 2).

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

### *3.1. From event log to directly-follows time series*

An event log $L$ contains the recording of traces $\sigma \in L$ which are sequences of events produced by an information system during its execution. A trace $\sigma = \langle e_1, ..., e_{|\sigma|} \rangle \in \Sigma^*$ is a finite sequence over the alphabet of activities $\Sigma$ which serves as the set of event types. Directly-follows relations between activities in an event log can be expressed as counting functions over activity pairs $>_L$: $\Sigma \times \Sigma \to \mathbb{N}$ so $>_L (a_1, a_2)$ counts the number of times activity $a_1$ is immediately followed by activity $a_2$ in the event log $L$. Directly-follows relations can be calculated over all traces or a subset of subtraces of the log. Finally, a Directly-Follows Graph (DFG) of the process then is the weighted directed graph with the activities as nodes and DF relations as weighted edges, i.e., $DFG = (\Sigma, >_L)$.

In order to obtain forecasts 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 [3, 5, 17], e.g., predictions at a point in the process rely on prefixes of a certain length, or particular state aggregations [24]. In the forecasting approach proposed here, we integrate concepts from time-series analysis. Hence, the evolution of the DFGs is monitored over intervals of the log where multiple aggregations are possible:

- **Equitemporal aggregation:** each sublog $L_s \in L$ of interval $s$ contains a part of the event log of some fixed 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 $L_s \in L$ of interval $s$ contains a part of the event log where an equal amount of DF pairs occurred which leads to well-populated sublogs when enough events are available.

Tables 1 and 2 exemplify the aggregations. These aggregations are useful for the following reasons. First, an equisized aggregation, in general, has a higher likelihood of the underlying DFs approaching a white noise time series which

is required for a wide range of time series forecasting techniques [25]. Second, both offer different thresholds at which forecasting can be applied. In the case of the equisized aggregation, it is easier to quickly construct a desired number of intervals by simply dividing an event log into the equisized intervals. However, most time series forecasting techniques rely on the time intervals being of equal duration which is embodied into the equitemporal aggregation [26]. 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 for the intervals.

| Case ID | Activity | Timestamp |
|---|---|---|
| 1 | $a_1$ | 11:30 |
| 1 | $a_2$ | 11:45 |
| 1 | $a_1$ | 12:10 |
| 1 | $a_2$ | 12:15 |
| 2 | $a_1$ | 11:40 |
| 2 | $a_1$ | 11:55 |
| 3 | $a_1$ | 12:20 |
| 3 | $a_2$ | 12:40 |
| 3 | $a_2$ | 12:45 |

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

| DF | Equitemporal | Equisized |
|---|---|---|
| $<_{Ls} (a_1, a_1)$ | (0,1,0) | (1,0,0) |
| $<_{Ls} (a_1, a_2)$ | (1,1,1) | (1,1,1) |
| $<_{Ls} (a_2, a_1)$ | (0,1,0) | (0,1,0) |
| $<_{Ls} (a_2, a_2)$ | (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)). Note that these aggregations are ordered based on the timestamp of the second activity.

An overview of the full pre-processing is given in Figure 3.

### *3.2. From DF time series to process model forecasts*

The goal of process model forecasting is to obtain a forecast for future DFGs by combining the forecasts of all the DF time series. To this purpose, we propose to use time series techniques to forecast the DFG at time $T+h$ given time series up until $T$ $\widehat{DFG}_{T+h} = (\Sigma, \{\hat{>}_{T+h|T_{a_1,a_2}} | a_1, a_2 \in \Sigma \times \Sigma\})$ for which various algorithms can be used. In time series modelling, the main objective is to obtain a forecast $\hat{y}_{T+h|T}$ for a horizon $h \in \mathbb{N}$ based on previous $T$ values in the series $(y_1, ..., y_T)$ [25]. For example, the naive forecast simply uses the last value of the time series $T$ as its forecast $\hat{y}_{T+h|T} = y_T$. An alternative naive forecast uses the average value of the time series $T$ as its forecast $\hat{y}_{T+h|T} = \frac{1}{T}\Sigma_i^T y_i$.

A choice 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 machine learning methods necessarily outperform traditional statistical approaches. The study in [27] found that this is not the case on a large number of datasets and the authors note that machine learning algorithms require significantly more computational power. This result was later reaffirmed, although it is noted that hybrid solutions are effective [28]. For longer horizons, traditional time series approaches still outperform machine learning-based models. Given

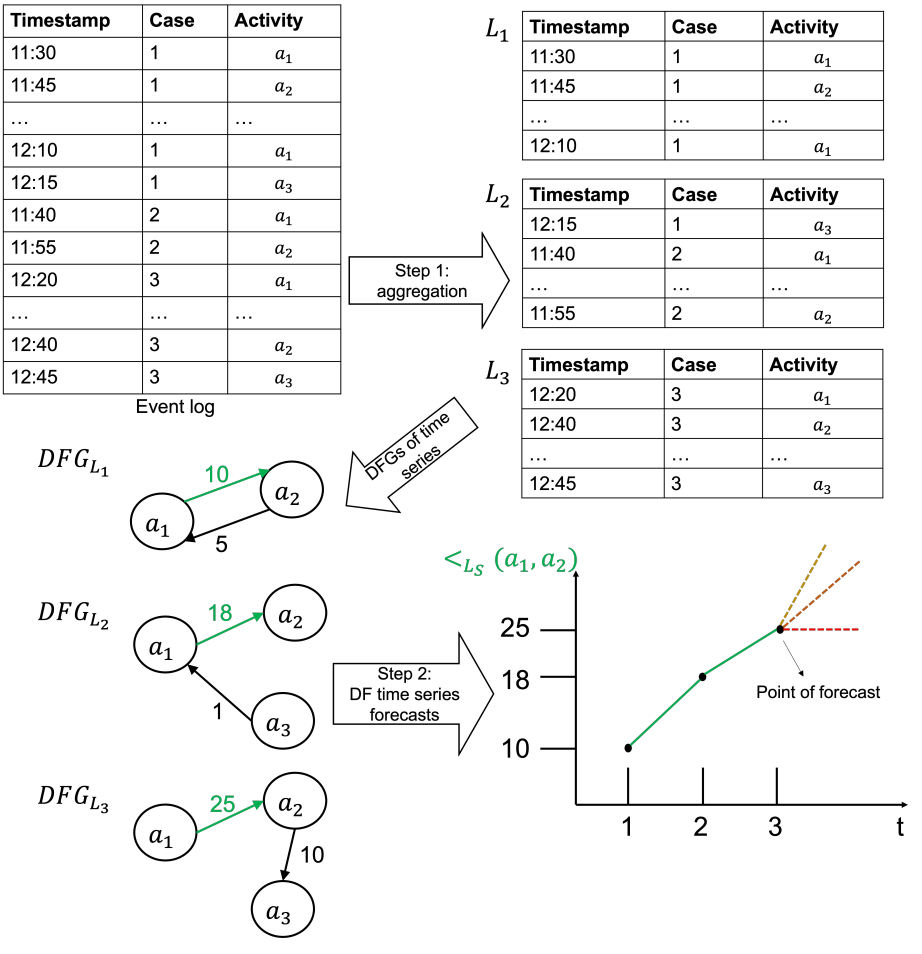



Figure 3: Overview of the PMF setup for equisized aggregation.

the potentially high number of DF pairs in a DFG, the proposed approach uses a time series algorithm for each DF series separately in a univariate setting, as well as in a multivariate setting. VAR models allow for the latter, but require a high number of intervals (at least as many as there are directly-follows times series times the lag coefficient). To estimate all parameters of all the time series despite their potentially strong performance can result in unstable performance and estimation of the parameters [29, 30]. Machine learning models could potentially leverage interrelations between the different DFs but again would require a 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, with at least one observation per sublog/time interval present.

Finally, the impact of the number of intervals the log is divided in can strongly impact results and also relate to the fact whether there is an impact of underlying drifts [10]. With fewer intervals, the leaps in time series signals might become overwhelming, while too many intervals can stretch out the signal such that it does not contain enough information for time series to extract any useful trend into the future. Indeed, time series analysis concepts such as smoothing and differencing can resolve both to a certain extent as will be illustrated in Section 3.3, even though they do not fully eliminate the impact of the number of intervals on results.

### *3.3. Time series approaches*

A wide array of other forecasting techniques exist, ranging from simple models such as naive forecasts over to more advanced approaches such as exponential smoothing and auto-regressive models. Autoregressive, moving averages, AutoRegressive Integrating Moving Average (ARIMA), and varying variance models make up the main families of traditional time series forecasting techniques[25]. Many also exist in a seasonal variant due to their application in contexts such as sales forecasting.

The Simple Exponential Smoothing (SES) model uses a weighted average of past values whose importance exponentially decays as they are further into the past according to a smoothing parameter $\alpha$, where the Holt's models introduce a trend in the forecast:

$$\hat{y}_{t+h|t} = l_t$$

with $l_t$ the smoothing equation

$$l_t = \alpha y_t + (1 - \alpha) l_{t-1}$$

The Holt's and Holt-Winters' model (HW) are an extension to SES model by adding trend and trend and seasonality effects respectively. The additive HW model (i.e. the seasonal component is additive to the trend and not multiplicative) can be formalised as follows:

$$\hat{y}_{t+h|h} = l_t + hb_t + s_{t+h-m(k+1)}$$

with

$$l_t = \alpha(y_t - s_{t-m}) + (1 - \alpha)(l_{t-1} + b_{t-1})$$
$$b_t = \beta(l_t - l_{t-1}) + (1 - \beta)b_{t-1}$$
$$s_t = \gamma(y_t - l_{t-1} - b_{t-1}) + (1 - \gamma)s_{t-m}$$

where $b_t$ is the trend component and $s_t$ is the seasonal component with respective smoothing parameters $\beta$ and $\gamma$. Exponential smoothing models often perform very well despite their simple setup [27]. ARIMA models are based on autocorrelations 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 $p$ and a Moving Average (MA) model of order $q$. An AR($p$)

model uses the past $p$ values in the time series and to apply a regression over them as follows:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \epsilon_t$$

with $c$ the intersect prediction, and $\epsilon_t$ the error term. An MA($q$) model regresses the forecast errors as follows:

$$y_t = c + \epsilon_t + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q}$$

Given the necessity of using a white noise series for AR and MA models, data is often differenced to obtain such series [25], with a differenced observation written $y^{'}$. ARIMA models then combine both AR and MA models where the integration occurs after modelling, as these models are fitted over differenced time series:

$$y^{'} = c + \phi_1 y^{'}_{t-1} + \cdots + \phi_p y^{'}_{t-p} + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q} + \epsilon_t$$

ARIMA models are considered to be one of the strongest time series modelling techniques [25]. An extension to ARIMA, which is widely used in econometrics, are the (Generalized) AutoRegressive Conditional Heteroskedasticity ((G)ARCH) models [31]. These models relax the assumption that the variance of the error term has to be constant over time, and 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 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 and the past values of variance:

$$y_t = x^{'}_t b + \epsilon_t$$

with $\epsilon_t \sim \mathcal{N}(0, \sigma_t^2)$ and

$$\sigma_t^2 = \omega + \theta_1 \epsilon_{t-1}^2 + \cdots + \theta_{t-q}^2 + \phi_1 \sigma_{t-1}^2 + \cdots + \phi_p \sigma_{t-p}^2$$

Note how the differenced observations of the AR part are replaced by the error variance $\sigma$. (G)ARCH models often outperform ARIMA models in contexts such as the forecast of financial indicators, in which the variance often changes over time [31]. In general, we can regard linear SES models as a subset of ARIMA models, where (G)ARCH models are specializations of ARIMA models that can be regarded as increasingly complex and better capable of modelling particular intricacies in the time series. However, the success of different models for forecasting purposes does not depend on their complexity, and the most suitable technique is mainly determined by performance on training and test sets.

A multivariate generalisation of autoregressive models exists in the form of vector autoregressions. Such regressions are comprised of a forecast per time series present which is made out of an autoregression for a particular lag $l$ with

all time series present in the model, for example for two time series and a lag of 1:

$$y_{1,t} = c_1 + \phi_{11,1} y_{1,t-1} + \phi_{12,1} y_{2,t-1} + e_{1,t}$$
$$y_{2,t} = c_2 + \phi_{21,1} y_{1,t-1} + \phi_{22,1} y_{2,t-1} + e_{2,t}$$

where $e_{1,t}$ and $e_{2,t}$ are white noise processes that can be correlated. $e_{11,1}$ is the effect of time series 1 on itself at lag 1, and $e_{1,2}$ is the effect of time series 2 on time series 1 at lag 1.

This allows the inference of an integrated model where the correlations over time between time series is used to estimate the coefficients of the model which typically happens by equation-by-equation maximum likelihood estimation using ordinary least squares [32]. Given that there are at least as many parameters as the number of time series times the lag, the number of parameters to be estimated grows quickly which impedes calculability as the covariance matrix of coefficients becomes singular [33]. Forecasts are obtained recursively given each time series has lag coefficients in every equation of the model.

### *3.4. Process change exploration*

In Sections 3.1 and 3.2 we described the approach for forecasting process models. To that end, gaining actual insights from such forecasted 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. 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 colour is close to black. For colouring edges, we reused the idea of the three colour schema from [34].

**Timeline view with brushed regions.** This view represents the area chart graph that shows how the number of activity executions changes with time. The colour of the area chart is split into two parts, one for the actual data and the other to show the time range of forecasted values. 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. Analysts can also brush two regions of the area chart to select two time ranges, updating the DFG to the 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 in green.

**Activity and path sliders.** We adopt two sliders to simplify the DFG [35] and the aDFG for detailed exploration of the models.

Based on the described views, we conjecture that the analyst can accomplish 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 entropic relevance of the actual against that of the forecasted DFG, as discussed in Section 2. This is done for various parts of the log, i.e. forecasts for the middle time spans 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 results for both the equisized and equitemporal aggregation. In Subsection 4.4 the implementation of the Process Change Exploration tool is discussed, followed by the results of a user study on this tool verifying its perceived usefulness and ease-of-use in Section 4.5.

### *4.1. Re-sampling and test setup*

To obtain training data, time series are constructed by specifying the number of intervals (i.e., time steps in the DF time series) using either equitemporal or equisized aggregation, as described in Section 3.1. Time series algorithms are parametric and sensitive to sample size requirements [36]. Depending on the number of parameters a model uses, a minimum size of at least 50 steps is not uncommon. However, 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 of length 25 is used meaning all test sets contain 25 intervals, but the training sets are varied from $ts = 25$ to $ts = 75$ intervals; the forecasts progressively target the forecast of intervals 25-50 (the second quarter of intervals) over to 75-100 (the last quarter of intervals). This allows us to inspect the difference in results when only a few data points are used, or data points in the middle or towards the end of the available event data are used.

A model from each time series family discussed in Section 3.3 is selected, i.e., a Holt Winter's model (HW) for exponential smoothing, an autoregressive model (AR), an ARIMA model, a GARCH model, and a VAR model. For each of these models, the best-performing parameters were retained resulting in the use of AR(2), ARIMA(2,1,2), GARCH(1), and a VAR(1). Besides, a naive model (NAV) in the form of an average forecast is used where $\hat{y}_{T+h|T} = \frac{\Sigma_{i=1}^{T} y_i}{T}$ as a simple baseline.

Resampling is applied based on a 10-fold cross-validation constructed following a rolling window approach for all horizon steps $h \in [1, 25]$ (i.e. the number of steps forecasted ahead) where a recursive strategy is used to iteratively obtain a forecast $\hat{y}_{t+h|t+h-1}$ where $t$ is 25, 50, and 75 to obtain forecasts up til time steps 50, 75, 100, hence $(y_1, \ldots, y_t, \ldots, \hat{y}_{t+h-1})$ [37]. The cross-validation builds ten training sets of length $ts$ which range from $(y_1, \ldots, y_{t-h-f})$ and 10 test sets which range from $(y_{t-h-f+1}, \ldots, y_{t-f})$ with $f \in [0, 9]$ the fold index [38]. 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 forecasts.

Six often-used, publicly available event logs are used: the BPI challenge of 2012[2], 2017[3], and 2018[4], the sepsis cases event log, an Italian help desk event log[5], and a Road Traffic Fine Management Process log (RTFMP) event log[6]. Each of these logs has a diverse set of characteristics in terms of case and activity volume and average trace length, as shown in Table 3.

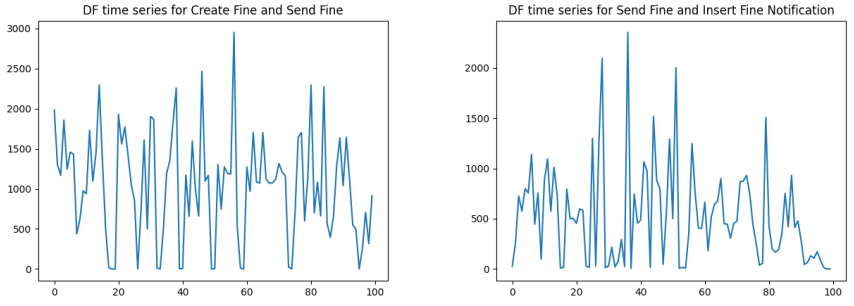
| Event log | # cases | # activities | Average trace length |
|---|---|---|---|
| **BPI 12** | 13,087 | 36 | 20.02 |
| **BPI 17** | 31,509 | 26 | 36.83 |
| **BPI 18** | 43,809 | 170 | 57.39 |
| **Sepsis** | 1,050 | 16 | 14.49 |
| **RTFMP** | 150,370 | 11 | 3.73 |
| **Italian** | 4,580 | 14 | 4.66 |

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

An example of applying the equisized or equitemporal aggregation to the RTFMP event log with 100 intervals results in the DF time series of Figure 4, where the DF occurrences of the most frequently occurring activity pair is included. For the equisized aggregation, the number of DFs is indeed relatively stable over the log's timeline where for the equitemporal aggregation a noticeable decline of DF pairs is visible towards the end of the series which is due to the fact we only consider complete cases. The latter is done in order to avoid mismatches between beginning and end points in the DFGs as we want to retain a (relatively) sound DFG in every interval. This phenomenon is typical in event logs, as processes usually have particular endpoint activities, but can also be due to the unequal distribution of events over the event log's timeline.

There are a few considerations concerning the DF time series in these event logs. Firstly, some of the event logs contain a warm-up and cool-down phase where the DF time series are ramping up and slowing down. The equitemporal aggregation can suffer from event logs in which events do not occur frequently throughout the complete log's timespan. For instance, the sepsis log's number

[2]https://doi.org/10.4121/uuid:3926db30-f712-4394-aebc-75976070e91f
[3]https://doi.org/10.4121/uuid:5f3067df-f10b-45da-b98b-86ae4c7a310b
[4]https://doi.org/10.4121/uuid:3301445f-95e8-4ff0-98a4-901f1f204972
[5]https://doi.org/10.4121/uuid:0c60edf1-6f83-4e75-9367-4c63b3e9d5bb
[6]https://doi.org/10.4121/uuid:270fd440-1057-4fb9-89a9-b699b47990f5




(a) Most common DF - equisize

(b) Most common DF - equitemp

Figure 4: Example of the DF time series of the most frequently occurring activity pair of the RTFMP log.

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). Secondly, suppose the level of occurrences of the DF pairs is low and close to zero. In that case, the series might be too unsuitable for analysis using white noise series analysis techniques that assume stationarity, i.e., for the lags to be identically and independently distributed with a zero mean [32]. Autoregressive models, including AR, ARIMA, VAR, and GARCH assume stationarity, but exponential smoothing models such as HW does not. Ideally, every time series should be evaluated using a stationarity test such as the Dickey-Fuller unit root test [39], and an appropriate lag order established for differencing to ensure a white noise process is used for training. Furthermore, for each algorithm, especially for ARIMA-based models, (partial) auto-correlation has to be established to obtain 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 were used for $p$, $d$, and $q$ and applied to all DF pairs where only the best-performing are reported below for comparison with the other time series techniques as discussed earlier.

### *4.2. Results*

All pre-processing was done in Python with a combination of *pm4py*[7] and the *statsmodels* package [40]. The code is publicly available[8].

To get a grasp of the forecasting performance in combination with the actual use of DFGs (which are rarely used in their non-aggregated form [12]) 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%

[7] https://pm4py.fit.fraunhofer.de
[8] https://github.com/JohannesDeSmedt/pmf

| aggregation | intervals | technique | BPI12 mean | BPI12 std | BPI12 min | BPI12 max | sepsis mean | sepsis std | sepsis min | sepsis max | RTFMP mean | RTFMP std | RTFMP min | RTFMP max |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| equisize | 50 | VAR | 88.07 | 6.91 | 73.78 | 101.03 | 85.46 | 10.87 | 63.97 | 111.84 | 358.26 | 674.74 | 97.95 | 3101.67 |
| | | ar2 | 10.63 | 5.24 | 4.04 | 42.10 | 68.22 | 14.42 | 36.83 | 116.17 | 381.40 | 982.03 | 7.30 | 5502.33 |
| | | arima212 | 11.08 | 5.53 | 3.88 | 49.39 | 67.46 | 14.36 | 40.43 | 116.81 | 293.57 | 934.19 | 7.59 | 5893.36 |
| | | garch | 11.47 | 2.91 | 4.45 | 18.60 | 67.99 | 14.18 | 38.89 | 116.48 | 357.39 | 1085.66 | 6.15 | 5849.74 |
| | | hw | 8.61 | 2.32 | 3.78 | 14.13 | 63.91 | 13.55 | 38.72 | 115.55 | 397.46 | 1156.50 | 5.91 | 7596.44 |
| | | nav | 9.74 | 2.80 | 4.30 | 18.52 | 62.62 | 13.21 | 37.46 | 110.80 | 290.45 | 946.94 | 7.33 | 5641.56 |
| equisize | 75 | VAR | 89.21 | 8.56 | 75.95 | 108.16 | 86.98 | 6.45 | 75.51 | 103.85 | 277.55 | 505.87 | 114.72 | 2673.45 |
| | | ar2 | 8.45 | 1.95 | 4.96 | 12.53 | 60.80 | 12.14 | 33.46 | 88.77 | 104.93 | 337.42 | 10.68 | 1790.98 |
| | | arima212 | 10.60 | 4.32 | 4.66 | 33.35 | 60.07 | 13.06 | 35.08 | 92.05 | 104.42 | 338.38 | 9.04 | 1878.87 |
| | | garch | 8.60 | 1.96 | 5.01 | 12.47 | 62.07 | 12.47 | 39.21 | 89.32 | 109.74 | 362.62 | 8.84 | 1970.37 |
| | | hw | 8.96 | 2.11 | 5.00 | 13.97 | 58.09 | 12.08 | 30.26 | 89.06 | 110.36 | 345.68 | 7.55 | 2199.72 |
| | | nav | 8.56 | 1.96 | 4.90 | 12.28 | 57.73 | 12.05 | 33.04 | 89.01 | 105.81 | 343.72 | 9.96 | 1807.61 |
| equisize | 100 | VAR | 88.78 | 9.19 | 72.44 | 113.82 | 89.18 | 6.75 | 77.10 | 117.81 | 322.52 | 573.78 | 124.95 | 3100.75 |
| | | ar2 | 9.62 | 3.28 | 3.98 | 29.19 | 62.44 | 11.30 | 37.02 | 105.59 | 110.14 | 326.57 | 12.36 | 1791.10 |
| | | arima212 | 14.34 | 13.80 | 3.98 | 104.21 | 58.99 | 10.99 | 27.54 | 100.18 | 248.45 | 760.90 | 4.77 | 7434.31 |
| | | garch | 10.17 | 3.25 | 4.67 | 30.55 | 60.34 | 11.51 | 37.51 | 102.28 | 117.15 | 348.42 | 11.55 | 1857.00 |
| | | hw | 10.11 | 3.21 | 3.99 | 29.70 | 57.09 | 11.11 | 33.59 | 93.20 | 130.36 | 393.31 | 7.59 | 2327.77 |
| | | nav | 9.82 | 3.30 | 4.09 | 29.56 | 58.03 | 11.25 | 37.70 | 97.96 | 115.34 | 341.69 | 12.71 | 1813.71 |
| equitemp | 50 | VAR | 81.77 | 25.33 | 58.39 | 194.60 | 83.83 | 7.73 | 68.41 | 118.07 | 196.13 | 62.50 | 118.48 | 374.17 |
| | | ar2 | 21.53 | 20.59 | 5.01 | 158.45 | 73.99 | 10.65 | 38.71 | 102.43 | 109.47 | 146.13 | 9.49 | 712.35 |
| | | arima212 | 16.78 | 16.69 | 3.70 | 97.46 | 70.75 | 12.32 | 43.21 | 108.57 | 92.33 | 89.39 | 8.31 | 715.96 |
| | | garch | 17.80 | 14.70 | 7.43 | 89.54 | 71.80 | 11.52 | 38.86 | 104.51 | 51.23 | 44.37 | 8.60 | 602.22 |
| | | hw | 13.09 | 15.22 | 3.20 | 95.26 | 71.60 | 11.06 | 39.99 | 114.96 | 62.94 | 55.93 | 11.34 | 683.62 |
| | | nav | 15.57 | 14.30 | 5.14 | 86.81 | 68.01 | 11.45 | 37.58 | 102.64 | 50.92 | 44.47 | 8.69 | 603.81 |
| equitemp | 75 | VAR | 82.69 | 17.55 | 58.18 | 138.38 | 87.16 | 9.24 | 76.30 | 119.53 | 168.72 | 38.68 | 121.04 | 304.63 |
| | | ar2 | 9.97 | 5.13 | 5.40 | 36.72 | 56.89 | 11.09 | 34.16 | 80.91 | 29.74 | 20.61 | 6.99 | 109.49 |
| | | arima212 | 13.09 | 7.58 | 4.83 | 45.64 | 56.40 | 10.88 | 31.15 | 80.06 | 28.55 | 21.01 | 9.96 | 117.09 |
| | | garch | 10.29 | 5.20 | 5.57 | 37.43 | 57.75 | 10.70 | 36.07 | 81.65 | 30.40 | 21.96 | 11.38 | 109.56 |
| | | hw | 10.46 | 5.48 | 5.44 | 42.08 | 56.11 | 10.54 | 32.25 | 80.97 | 29.30 | 21.49 | 8.49 | 128.28 |
| | | nav | 10.14 | 5.07 | 5.48 | 36.46 | 56.96 | 11.07 | 35.75 | 80.38 | 29.70 | 20.53 | 11.37 | 109.42 |
| equitemp | 100 | VAR | 85.70 | 21.55 | 63.34 | 180.27 | 159.86 | 154.56 | 77.71 | 708.27 | 186.89 | 63.38 | 106.26 | 610.55 |
| | | ar2 | 12.43 | 6.81 | 3.96 | 36.83 | 110.88 | 153.37 | 35.24 | 705.31 | 36.01 | 36.65 | 12.33 | 302.98 |
| | | arima212 | 11.99 | 6.44 | 4.51 | 39.59 | 101.22 | 126.07 | 31.75 | 615.20 | 33.83 | 37.01 | 2.35 | 313.23 |
| | | garch | 12.71 | 6.84 | 5.37 | 36.80 | 114.21 | 162.07 | 35.91 | 736.41 | 36.00 | 36.62 | 12.69 | 302.43 |
| | | hw | 12.09 | 6.27 | 4.00 | 37.07 | 100.73 | 128.02 | 17.49 | 615.20 | 36.10 | 37.83 | 9.05 | 302.09 |
| | | nav | 12.63 | 6.86 | 5.32 | 36.33 | 113.56 | 160.20 | 35.94 | 736.41 | 36.15 | 36.82 | 12.64 | 304.22 |

Table 4: Overview of the mean percentage error in terms of entropic relevance for the non-reduced DFGs for datasets BPI 12, sepsis, and RTFMP. Best (lowest) value indicated per column and aggregation/intervals combination in red, blue colour coding indactes higher (worse) values.

reduction which is node-based (i.e. only the Q2/Q3 percentile of nodes in terms of frequency is retained). The reduction takes place after the full DFG has been forecasted. Hence, we obtain a measure of accuracy in terms of the discrepancy of the actual and forecasted model behaviour. 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 7. NAs/NaNs are reported when the algorithms did not converge, no data was available (e.g. sepsis for the 75-100 equitemporal intervals).

When no reduction to the DFGs is applied, the summary statistics over the 10 cross-validation folds show that, on average, it is possible to have an error rate below 10% to 15% for all aggregation types and number of intervals used for the training set for BPI12 and BPI17. For the Italian helpdesk event log the errors are between 13% and 34% for the best-performing models. For the sepsis log, errors are between 57-100% on average for the best-performing model with high standard deviations. For the RTFMP logs, the errors are over 100% for the equisize aggregation, but not the equisize aggregation, with mean errors between 28 and 51%.

Overall, the difference between equisize and equitemp aggregation is smaller for BPI12, BPI17, BPI18 (except when 100 intervals are used), sepsis, and the Italian helpdesk log. For RTFMP, the equitemp aggregation performs better

| aggregation | intervals | technique | BPI17 mean | BPI17 std | BPI17 min | BPI17 max | Italian mean | Italian std | Italian min | Italian max | BPI18 mean | BPI18 std | BPI18 min | BPI18 max |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| equisize | 50 | VAR | 112.68 | 4.91 | 103.46 | 120.41 | 139.25 | 24.36 | 106.69 | 210.14 | 85.46 | 10.87 | 63.97 | 111.84 |
| | | ar2 | 6.54 | 2.33 | 3.17 | 13.88 | 23.00 | 10.92 | 4.38 | 62.34 | 542.70 | 1599.48 | 42.29 | 21666.56 |
| | | arima212 | 18.84 | 22.25 | 2.76 | 105.08 | 23.18 | 10.95 | 3.23 | 50.81 | 326.00 | 278.98 | 51.43 | 2066.30 |
| | | garch | 6.61 | 2.58 | 3.06 | 15.95 | 21.68 | 10.15 | 7.16 | 45.14 | 316.03 | 274.01 | 54.18 | 2072.31 |
| | | hw | 6.74 | 2.45 | 2.44 | 15.00 | 22.04 | 10.16 | 3.45 | 49.81 | 329.31 | 290.54 | 23.69 | 2058.44 |
| | | nav | 6.53 | 2.39 | 2.41 | 14.67 | 22.08 | 10.08 | 7.89 | 45.14 | 278.43 | 169.98 | 43.63 | 706.46 |
| equisize | 75 | VAR | 106.34 | 4.88 | 92.61 | 114.15 | 143.36 | 15.50 | 113.13 | 173.80 | 86.98 | 6.45 | 75.51 | 103.85 |
| | | ar2 | 35.25 | 66.48 | 2.80 | 393.70 | 23.38 | 17.21 | 7.24 | 122.01 | 241.23 | 247.02 | 19.16 | 1153.25 |
| | | arima212 | 8.54 | 2.86 | 3.17 | 16.90 | 15.04 | 5.39 | 3.72 | 34.30 | 274.73 | 281.29 | 20.51 | 1403.02 |
| | | garch | 8.25 | 2.92 | 2.96 | 14.95 | 31.95 | 6.84 | 21.03 | 50.19 | 257.39 | 256.22 | 47.29 | 2085.05 |
| | | hw | 8.21 | 2.79 | 2.99 | 15.00 | 13.13 | 4.04 | 4.16 | 26.75 | 318.56 | 363.47 | 15.32 | 2819.24 |
| | | nav | 8.24 | 2.89 | 3.01 | 14.82 | 21.11 | 6.61 | 9.64 | 44.03 | 225.10 | 234.22 | 39.30 | 1123.19 |
| equisize | 100 | VAR | 107.80 | 5.17 | 95.09 | 123.71 | 167.47 | 74.47 | 103.79 | 495.34 | 89.18 | 6.75 | 77.10 | 117.81 |
| | | ar2 | 5.90 | 2.46 | 3.01 | 27.00 | 29.35 | 20.24 | 5.28 | 105.86 | 160.97 | 133.48 | 38.21 | 1054.59 |
| | | arima212 | 5.79 | 2.51 | 2.54 | 23.77 | 28.78 | 20.10 | 5.47 | 109.74 | 245.85 | 132.92 | 58.72 | 963.78 |
| | | garch | 6.26 | 2.57 | 3.58 | 27.32 | 53.47 | 27.48 | 18.77 | 202.86 | 194.94 | 141.55 | 78.23 | 803.37 |
| | | hw | 5.87 | 2.46 | 3.26 | 26.50 | 31.76 | 20.70 | 4.69 | 112.55 | 311.46 | 201.63 | 23.19 | 1281.15 |
| | | nav | 6.01 | 2.45 | 3.54 | 27.35 | 30.68 | 20.30 | 7.77 | 107.63 | 133.19 | 81.83 | 62.90 | 497.39 |
| equitemp | 50 | VAR | 109.56 | 5.17 | 99.68 | 119.07 | 141.20 | 17.12 | 107.64 | 174.88 | 83.83 | 7.73 | 68.41 | 118.07 |
| | | ar2 | 8.23 | 5.94 | 2.71 | 46.27 | 23.53 | 10.87 | 4.79 | 46.27 | nan | nan | nan | nan |
| | | arima212 | 15.29 | 18.60 | 3.76 | 90.33 | 22.18 | 10.28 | 5.55 | 57.02 | 224.07 | 110.67 | 4.75 | 732.36 |
| | | garch | 7.83 | 2.99 | 4.01 | 19.54 | 23.35 | 10.77 | 6.17 | 46.98 | nan | nan | nan | nan |
| | | hw | 7.02 | 2.99 | 3.18 | 19.35 | 22.60 | 11.29 | 4.07 | 53.29 | 230.04 | 90.34 | 132.78 | 439.18 |
| | | nav | 7.10 | 2.89 | 3.44 | 18.71 | 23.38 | 10.79 | 7.21 | 46.89 | nan | nan | nan | nan |
| equitemp | 75 | VAR | 102.95 | 5.67 | 88.52 | 115.82 | 172.40 | 61.99 | 88.45 | 467.22 | 87.16 | 9.24 | 76.30 | 119.53 |
| | | ar2 | 47.42 | 97.11 | 3.25 | 475.89 | 27.55 | 26.78 | 3.01 | 174.96 | 787.09 | 1298.30 | 65.95 | 9230.37 |
| | | arima212 | 9.09 | 3.51 | 3.17 | 21.84 | 25.34 | 20.00 | 2.15 | 120.47 | 278.41 | 155.15 | 90.80 | 1001.49 |
| | | garch | 8.09 | 2.89 | 3.19 | 17.54 | 41.74 | 18.41 | 17.60 | 110.58 | 282.05 | 171.57 | 87.70 | 990.92 |
| | | hw | 8.27 | 2.93 | 2.34 | 18.02 | 19.96 | 19.68 | 2.45 | 141.88 | 251.51 | 159.76 | 58.06 | 1203.91 |
| | | nav | 8.04 | 2.85 | 3.28 | 17.25 | 26.56 | 19.59 | 8.70 | 110.64 | 223.44 | 172.12 | 71.61 | 1314.16 |
| equitemp | 100 | VAR | 107.76 | 8.31 | 92.05 | 145.53 | 130.46 | 35.64 | 90.24 | 515.18 | 159.86 | 154.56 | 77.71 | 708.27 |
| | | ar2 | 7.93 | 7.86 | 2.97 | 60.95 | 35.09 | 13.97 | 21.86 | 179.18 | 368.73 | 1045.55 | 39.00 | 12059.52 |
| | | arima212 | 9.52 | 12.54 | 2.83 | 77.44 | 34.14 | 15.88 | 16.68 | 186.61 | 166.21 | 163.37 | 37.50 | 967.42 |
| | | garch | 12.06 | 7.89 | 6.62 | 62.59 | 50.77 | 12.82 | 30.15 | 147.94 | 283.94 | 629.02 | 115.60 | 7014.30 |
| | | hw | 7.18 | 7.26 | 2.07 | 58.68 | 34.06 | 12.68 | 18.65 | 136.92 | 173.25 | 172.96 | 56.13 | 977.89 |
| | | nav | 8.19 | 8.33 | 3.44 | 61.01 | 35.46 | 11.07 | 24.07 | 151.50 | 200.49 | 624.88 | 57.44 | 6981.11 |

Table 5: Overview of the mean percentage error in terms of entropic relevance for the non-reduced DFGs for datasets BPI 17, Italian helpdesk, and BPI 18. The colour coding is similar to Table 4.

which might have to do with the very spread-out occurring of events in the log with events spread out evenly over the recording history, or the very low average trace length.

The models that score best on average include the Holt-Winters' model, AR model, and sometimes the naive forecast. In the latter case, however, the results of the other models are similar which does not hold for the former case. Interestingly, the VAR models perform worse than all other models except for the BPI18 dataset where it achieves the lowest mean error and standard deviation of all techniques. This indicates that, in a setting when a very high number of activities is present and traces are longer, modelling the correlations among the directly-follows time series seems to become important.

The average error rates after a 50% reduction of the DFGs are reported in Table 6. The error rates drop to the 5-20% range for BPI12, BPI17, and sepsis (except for 100 intervals/equitemp), and the 11-20% range for Italian. Similar 100%+ rates for RTFMP with much worse results for the equitemp aggregation are recorded, next to a jump to the 145-282% range for BPI18. For BPI17, many NaNs are recorded which is due to the fact that the ER computation is hampered by non-fitting models which are reduced to an extent that replaying traces over them is not possible. In this case, however, VAR models perform best. For the other settings, AR/HW models perform well with good results.

When a 75% reduction is applied as in Table 7, the error results go up for

| intervals | aggregation technique | equisize BPI12 | BPI17 | BPI18 | Italian | RTFMP | sepsis | equitemp BPI12 | BPI17 | BPI18 | Italian | RTFMP | sepsis |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 50 | VAR | 32.57 | 5.79 | 391.71 | 195.32 | 396.28 | 12.96 | nan | nan | 233.55 | 200.84 | 177.60 | 11.72 |
| | ar2 | 5.23 | nan | 546.62 | 21.10 | 276.17 | 7.90 | 13.23 | nan | nan | 20.23 | 120.33 | 6.32 |
| | arima212 | 8.63 | nan | 338.01 | 19.91 | 259.16 | 7.86 | 10.68 | nan | 223.93 | 20.31 | 126.95 | 5.92 |
| | garch | 4.72 | nan | 304.34 | 19.14 | 301.48 | 7.96 | 19.13 | nan | nan | 19.90 | 77.79 | 6.16 |
| | hw | 11.83 | nan | 335.00 | 19.14 | 262.00 | 7.64 | 7.18 | nan | 229.95 | 19.38 | 79.98 | 5.87 |
| | nav | 7.30 | nan | 282.67 | 19.12 | 303.54 | 7.86 | 9.43 | nan | nan | 20.00 | 77.79 | 6.01 |
| 75 | VAR | 25.62 | 6.43 | 380.13 | 194.41 | 299.66 | 15.44 | 34.17 | nan | 290.70 | 226.33 | 120.72 | 10.51 |
| | ar2 | 5.46 | nan | 269.12 | 20.93 | 213.02 | 8.78 | 6.59 | nan | 793.74 | 22.15 | 38.23 | 6.87 |
| | arima212 | 5.77 | nan | 297.56 | 13.24 | 213.23 | 8.81 | 8.62 | nan | 281.25 | 20.58 | 38.08 | 7.54 |
| | garch | nan | nan | 274.27 | 29.99 | 212.82 | 9.04 | 5.56 | nan | 281.88 | 36.39 | 38.15 | 6.92 |
| | hw | 8.71 | nan | 343.31 | 11.42 | 209.68 | 8.12 | 6.52 | nan | 245.68 | 15.21 | 36.18 | 6.76 |
| | nav | 5.44 | nan | 232.22 | 19.08 | 213.22 | 8.92 | 6.52 | nan | 221.28 | 21.36 | 38.14 | 6.79 |
| 100 | VAR | 29.06 | 19.28 | 352.40 | 194.25 | 129.02 | 9.56 | 39.71 | 93.58 | 345.67 | 138.33 | 138.99 | 72.74 |
| | ar2 | 5.57 | nan | 426.38 | 16.99 | 36.22 | 6.33 | 12.20 | nan | 368.92 | 22.35 | 48.92 | 71.97 |
| | arima212 | 9.75 | nan | 265.04 | 17.11 | 57.11 | 5.89 | 16.36 | nan | 203.26 | 20.08 | 48.79 | 40.68 |
| | garch | nan | nan | 195.73 | 38.22 | 36.18 | 6.56 | 11.62 | nan | 300.41 | 34.45 | 49.72 | 68.11 |
| | hw | 8.19 | nan | 312.57 | 19.03 | 41.28 | 5.44 | 11.76 | nan | 256.30 | 20.20 | 50.14 | 68.96 |
| | nav | 6.80 | nan | 145.51 | 18.28 | 36.25 | 6.32 | 11.88 | nan | 225.90 | 21.07 | 49.64 | 68.12 |

Table 6: Overview of the mean percentage error in terms of entropic relevance for the DFGs with a 50% reduction.

| intervals | aggregation technique | equisize BPI12 | BPI17 | BPI18 | Italian | RTFMP | sepsis | equitemp BPI12 | BPI17 | BPI18 | Italian | RTFMP | sepsis |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 50 | VAR | 27.70 | 4.60 | 361.71 | 190.52 | 396.03 | 11.66 | 39.03 | nan | 222.59 | 196.96 | 177.41 | 9.60 |
| | ar2 | 19.75 | 1.54 | 521.13 | 187.47 | 435.97 | 10.07 | 33.91 | nan | nan | 193.56 | 207.51 | 10.11 |
| | arima212 | 20.24 | 1.46 | 343.66 | 187.11 | 405.31 | 10.21 | 31.61 | nan | 223.93 | 195.34 | 166.15 | 9.66 |
| | garch | 19.77 | 1.54 | 328.34 | 186.98 | 396.03 | 10.24 | 34.25 | nan | nan | 193.17 | nan | 8.78 |
| | hw | 19.73 | 1.52 | 340.12 | 186.97 | 398.52 | 10.24 | 31.42 | nan | 229.95 | 194.33 | 172.03 | 9.45 |
| | nav | 19.77 | 1.53 | 286.83 | 186.94 | 391.54 | 10.20 | 32.75 | nan | nan | 193.08 | nan | 9.04 |
| 75 | VAR | 21.06 | 4.60 | 351.36 | 187.05 | 299.31 | 11.59 | 29.03 | nan | 284.64 | 219.44 | 120.43 | 11.28 |
| | ar2 | 13.75 | 1.55 | 305.06 | 182.30 | 92.85 | 10.60 | 20.84 | nan | 436.85 | 215.91 | nan | 10.88 |
| | arima212 | 13.64 | 1.50 | 316.51 | 182.65 | 338.36 | 12.45 | 20.81 | nan | 301.97 | 216.15 | 101.14 | 11.20 |
| | garch | 13.80 | 1.53 | 323.75 | 182.14 | 299.31 | 10.66 | 20.85 | nan | 280.73 | 215.80 | nan | 11.42 |
| | hw | 13.66 | 1.58 | 347.12 | 181.99 | 289.34 | 11.39 | 20.73 | nan | 269.60 | 217.55 | nan | 11.08 |
| | nav | 13.82 | 1.53 | 273.17 | 182.28 | nan | 10.86 | 20.88 | nan | 247.11 | 215.89 | nan | 10.67 |
| 100 | VAR | 23.88 | 18.12 | 279.36 | 186.91 | nan | 10.02 | 33.07 | 99.91 | 340.78 | 133.80 | nan | 70.71 |
| | ar2 | 16.03 | 15.71 | 306.52 | 181.72 | 131.30 | 9.72 | 23.76 | 97.73 | 422.46 | 128.22 | nan | 69.73 |
| | arima212 | 17.04 | 15.75 | 266.32 | 182.15 | 107.36 | 9.69 | 23.72 | 100.63 | 222.26 | 134.97 | nan | 69.77 |
| | garch | 16.04 | 15.71 | 193.54 | 181.47 | nan | 9.48 | 23.76 | 97.75 | 234.44 | 128.19 | nan | 70.03 |
| | hw | 16.03 | 15.75 | 308.79 | 182.42 | nan | 9.50 | 23.63 | 97.54 | 297.85 | 150.56 | nan | 70.06 |
| | nav | 16.07 | 15.70 | 149.33 | 181.80 | nan | 9.79 | 23.79 | 97.72 | 315.43 | 128.46 | nan | 70.10 |

Table 7: Overview of the mean percentage error in terms of entropic relevance for the DFGs with a 75% reduction.

BPI12 and the Italian helpdesk logs, stay the same for BPI18, the RTFMP, and the sepsis event logs, and go down/become computable again for BPI17. For BPI17 there is a very strong impact on computability of ER with reduced models, which makes the results less revealing.

The improvement on some event logs after a 50% reduction can be caused by the fact that fewer variants cause convoluted DFGs and the forecasts are better capable of predicting the most prevalent paths through the activity graph without being punished in ER calculations for mistakes for missing activities and paths. However, when a very strong 75% reduction is made it appears that most techniques struggle to come up with a model which is replayable and compatible with ER calculations. Nevertheless, even in this scenario 10-20% error rates can be found for BPI12, BPI17, and sepsis for both aggregation types. The ER results are commensurate with the findings in [19], which contains entropic relevance results for the BPI12, sepsis, and RTFMP logs, indicating that entropic relevance of larger DFGs is lower (better) for RTFMP/Sepsis, and

the entropic relevance goes up strongly for reduced 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 and 50% reduced spectrum of DFG sizes as per [19], 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 traces with medium-sized alphabets (>20) such as the BPI12, BPI17, and Italian helpdesk logs 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. It is interesting to see that in this case multivariate models are providing an alternative to obtain good error rates. For the 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.

### *4.3. Reflection on the statistical analysis*

Overall, there exist many scenarios in which process model forecasting is delivering solid 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 readers of DFGs typically apply. In some cases, even a naive forecast is enough to obtain a low error rate. However, the HW, AR and ARIMA models report the best error rates in most cases. There does not seem any particular benefit of using heteroskedacity-aware methods such as (G)ARCH. Nevertheless, results are often close except when fewer training points are used. Still, we would like to point out that the results from Tables 4 to 5 do show a wide spread with strongly diverging minimum and maximum values for most models, datasets, number of intervals, and aggregation approaches. Especially for the RTFMP and BPI18 event logs the impact of DF information present in the different folds can lead to massive differences indicating that most techniques are susceptible to picking up or missing out on particular, local behavior. Given that these two logs exhibit the highest number of traces and activities respectively, it is clear that further experimentation is necessary in terms of choosing the number of intervals (e.g. shorter event logs might not be suitable for equitemporal aggregation with a high number of intervals), pre-selecting the number of activity pairs that could result in valid time series, and parameterize the statistical models. In general, it means that the application of the techniques should always be accompanied by a thorough statistical analysis to ensure results are robust to anomalies in the data and in line with the properties in the event log. In future work, the robustness of forecast algorithms will be further investigated, e.g., via scrutinising the confidence intervals of the forecasted DF outcomes.

Finally, it is clear that there is no evident benefit of using any advanced statistical forecasting approach used dealing with properties such as seasonality, (long-term) autocorrelation, strong trends, correlation between series, and other concepts which are modeled by approaches. This means that either the aggregations are not appropriately capturing these concepts, they are not present in (all) DF time series, or the approaches are not capable of extracting them appropriately. In future work, the impact of the pre-processing and aggregation in function of the event log properties will be further investigated to come up with robust representations of a process model over time, which will allow a more thorough analysis of what aspects are needed in a forecasting approach to model (a particular) business process over time.

In this sense, the main contribution of this paper is to provide a blueprint of what process model forecasting can be by suggesting a shift from the case perspective commonly found in predictive process monitoring through the analysis of model-wide abstractions in the form of a single instantiation which is the use of DF time series to forecast a DFG. As pointed out in this section, there are several angles along which extensions and improvements on this paradigm shift, including the use of stronger modelling approaches and different data aggregations, can be investigated.

### *4.4. 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 insights from such predicted data remains a difficult task for the analyst. This section sets off to present a novel visualisation system to aid analysts in exploring the event logs. The process of designing and implementing the system started by designing several prototypes that underwent rounds of discussions between the authors and peers from the process mining domain to mature into the implemented visualisation system.

The design of the PCE system is shown in Figure 5. It offers an interactive visualisation system with several connected views. The system is implemented using the D3.js JavaScript library and is available as an open-source project.[9]

### *4.5. User study*

In this section we discuss a user study on the perceived usefulness and ease-of-use of the PCE tool. To this purpose, a user study was performed with twelve participants with extensive knowledge of process mining and its tooling and who are familiar with DFGs. Of these 12 participants, 5 were from industry with either a consultancy or software engineering background which makes these results valid within a practitioner's context. The other 7 participants were from academia and were either in a PhD researcher or postdoc position which advanced knowledge of process mining. Their understanding of DFGs was verified using a series of introductory questions at the start of the study.

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

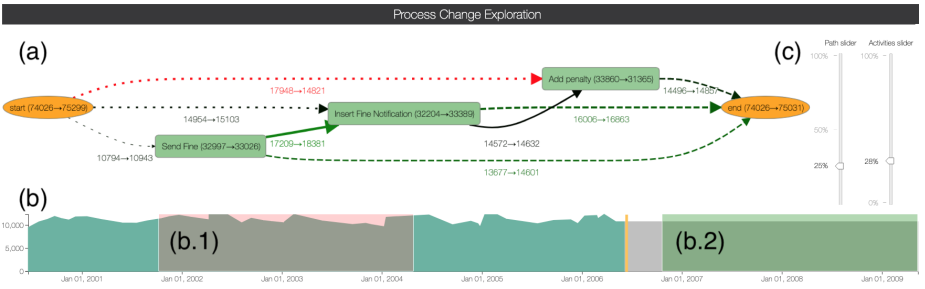



Figure 5: Process Change Exploration (PCE) system. *(a)* shows *Adaptation Directly-Follows Graph (aDFG)* view. *(b)* shows the *Timeline view with brushed regions* view. Users can brush one or more regions on this graph in order to filter the scope of the analysis *(b.1*, and *b.2)*. Two additional controls in *(c)* show the *activity and path sliders*.

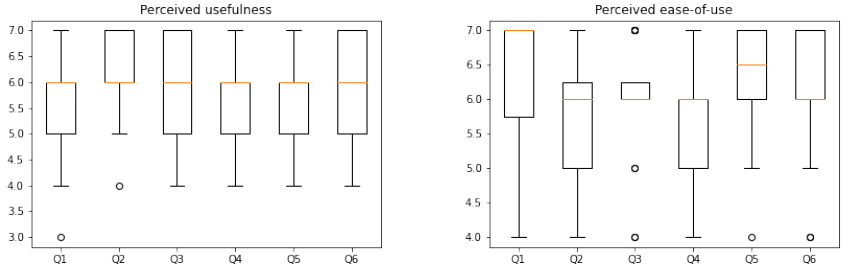
The user study setup required users to answer questions both regarding the longitudinal aspects of the PCE tool, i.e., whether the aDFG presentation is useful and easy to use, as well as whether the forecasting functionality can be used towards longitudinal analysis of processes. In the first question the user was tasked with describing all major changes between two DFGs from two historical (i.e. the training data) time spans. Besides, they had to answer questions about precise changes, e.g., whether a particular directly-follows relation occurred more or fewer times between the two time spans under scrutiny, to verify their use of the tool was commensurate its intended use. A second question was included which had a similar form but was focused on identifying changes between a historical DFG (from the training time span) and a forecasted DFG. All participants were capable of answering the questions correctly, indicating that their comprehending the tool was sufficient to answer the questions regarding its perceived usefulness and ease-of-use was adequate.

After the questions introduced the users to the functionality of the tool, the perceived usefulness and ease-of-use were tested using questions on a Likert scale of 1 to 7 using the Technology Acceptance Framework (TAM) with six questions per construct [41, 42] as detailed in Table 8. The results are included in Figure 6.

The statistical evidence indicates the users attributed the tool to both a high perceived usefulness and ease-of-use as demonstrated by the distribution with a median of 6 or over out of a maximum of 7 for all questions. There are no subquestions with significantly lower particular scores, indicating that the participants are unanimous both in terms of usefulness and ease-of-use. Especially Q1 and Q5 for ease-of-use obtained high scores, indicating the PCE tool is very easy to pick up quickly.

Besides questions focused on understanding the functionality, open feedback was collected on both the positive aspects of the tool as well as points of improvement.

Most users appreciated the clear interface, which was mentioned in five out of six cases, as the PCE tool builds on top of the DFG representation which




(a) Perceived usefulness

(b) Perceived ease-of-use

Figure 6: Boxplots of the results of the questions. The orange lines indicate the median values.

| | **Perceived usefulness** | **Perceived ease-of-use** |
|---|---|---|
| **1** | Using the PCE tool to analyze the event log would enable me to accomplish my analysis tasks more quickly. | Learning to operate the PCE tool would be easy for me. |
| **2** | Using the PCE tool would improve my performance in analyzing the event log. | I would find it easy to get the PCE tool to do what I want it to do. |
| **3** | Using the PCE tool would increase my analysis productivity. | My interaction with the PCE tool would be clear and understandable. |
| **4** | Using the PCE tool would enhance my effectiveness on the analysis job. | I would find the PCE tool to be flexible to interact with. |
| **5** | Using the PCE tool would make it easier to do my analysis job. | It would be easy for me to become skillful at using the PCE tool. |
| **6** | I would find the PCE tool useful for my analysis job. | I would find the PCE tool easy to use. |

Table 8: Questions used in the study.

is common in other DFG process mining tools such as ProM[10] and Disco[11] and provides an intuitive extension in this respect. Besides, the users liked the seamless integration of comparing both actual and forecasted DFGs.

Majority of the participants point out the utility of visualising and comparison of the two time frames. Participant P1 likes *"The ability to visualise and compare different timeframes in the event log without having to create two separate DFGs"*, while P2 states *"I can analyse two time frames, which might be helpful to compare e.g. before and after process changes."*

Points of improvement, besides minor implementation details such as date

[10]https://www.promtools.org/doku.php
[11]https://fluxicon.com/disco/

selection precision and arrow types, include the desire of users to be able to store time spans/enact different scenarios, adjusting the forecast on the go, and obtaining relative instead of absolute figures for the activity and directly-follows occurrences.

However, besides the inclusion of relative instead of absolute figures, none of these issues impede the overarching functionality of being able to compare DFGs between two time spans in detail drastically. Hence, based on both the quantitative and qualitative feedback the tool is deemed useful and easy to use. Participant P4 summarizes that PCE system is a *"novel way of interacting with data"*, while P6 states that the tool has *"nice user interface."*

## 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 promising forecasts regarding the future development of the process, including whether process drifts or major changes might occur in particular parts of the process. The presented forecasting approach is supported by the Process Change Exploration system, which allows analysts to compare various parts of the past, present, and forecasted future behaviour of the process. Our empirical evaluation demonstrates that, most notably for reduced process models with medium-sized alphabets, we can obtain below 15% MAPE in terms of conformance to the true models. In a user study on the PCE system, it was shown that adding the longitudinal and forecasting aspect of Adaptation Directly-Follows Graphs was desirable for users and the PCE system ease to use and useful for its purpose.

In future research, we plan to evaluate the use of machine learning techniques for process model forecasting. More specifically, we aim at using recurrent neural networks or their extension in long short-term memory networks (LSTMs) and transformer-based architectures, as well as hybrid methods or ensemble forecasts with the traditional time series approaches presented here. Furthermore, we want to explore opportunities for enriching our forecasted process models with confidence intervals by calculating the entropic relevance at different confidence levels and reporting the confidence intervals in the PCE system.

## Acknowledgements

This research was supported by the Research Foundation Flanders under grant G039923N and KU Leuven under project 3H200414. Artem Polyvyanyy was in part supported by the Australian Research Council project DP220101516. The research by Jan Mendling was supported by the Einstein Foundation Berlin under grant EPP-2019-524 and by the German Federal Ministry of Education and Research under grant 16DII133.

## References


[1] W. M. P. van der Aalst, in: Process Mining: Data Science in Action, Springer, 2016.

[2] C. Di Francescomarino, C. Ghidini, F. M. Maggi, F. Milani, Predictive process monitoring methods: Which one suits me best?, in: BPM, Vol. 11080 of LNCS, Springer, 2018, pp. 462–479.

[3] N. Tax, I. Verenich, M. La Rosa, M. Dumas, Predictive business process monitoring with LSTM neural networks, in: CAiSE, Vol. 10253 of LNCS, Springer, 2017, pp. 477–492.

[4] A. Rogge-Solti, M. Weske, Prediction of Remaining Service Execution Time Using Stochastic Petri Nets with Arbitrary Firing Delays, in: ICSOC, Vol. 8274 of LNCS, Springer, 2013, pp. 389–403.

[5] I. Teinemaa, M. Dumas, M. La Rosa, F. M. Maggi, Outcome-oriented predictive process monitoring: Review and benchmark, ACM Trans. Knowl. Discov. Data 13 (2) (2019) 17:1–17:57.

[6] G. Park, M. Song, Predicting performances in business processes using deep neural networks, Decis. Support Syst. 129 (2020) 113191.

[7] R. Poll, A. Polyvyanyy, M. Rosemann, M. Röglinger, L. Rupprecht, Process forecasting: Towards proactive business process management, in: BPM, Vol. 11080 of LNCS, Springer, 2018, pp. 496–512.

[8] M. L. van Eck, X. Lu, S. J. J. Leemans, W. M. P. van der Aalst, PM$^2$: A process mining project methodology, in: CAiSE, Vol. 9097 of LNCS, Springer, 2015, pp. 297–313.

[9] A. Maaradji, M. Dumas, M. La Rosa, A. Ostovar, Detecting sudden and gradual drifts in business processes from execution traces, IEEE Trans. Knowl. Data Eng. 29 (10) (2017) 2140–2154.

[10] A. Yeshchenko, C. Di Ciccio, J. Mendling, A. Polyvyanyy, Comprehensive process drift detection with visual analytics, in: ER, Vol. 11788 of LNCS, Springer, 2019, pp. 119–135.

[11] A. Yeshchenko, C. Di Ciccio, J. Mendling, A. Polyvyanyy, Visual drift detection for sequence data analysis of business processes, IEEE Trans. Vis. Comput. Graph. 28 (8) (2022) 3050–3068.

[12] W. M. P. van der Aalst, A practitioner's guide to process mining: Limitations of the directly-follows graph, in: CENTERIS/ProjMAN/HCist, Vol. 164 of Procedia Computer Science, Elsevier, 2019, pp. 321–328.

[13] J. Evermann, J.-R. Rehse, P. Fettke, Predicting process behaviour using deep learning, Decis. Support Syst. 100 (2017) 129–140.

[14] I. Verenich, M. Dumas, M. La Rosa, F. M. Maggi, I. Teinemaa, Survey and cross-benchmark comparison of remaining time prediction methods in business process monitoring, ACM Trans. Intell. Syst. Technol. 10 (4) (2019) 34:1–34:34.

[15] W. Kratsch, J. Manderscheid, M. Röglinger, J. Seyfried, Machine learning in business process monitoring: A comparison of deep learning and classical approaches used for outcome prediction, Bus. Inf. Syst. Eng. 63 (3) (2021) 261–276.

[16] D. A. Neu, J. Lahann, P. Fettke, A systematic literature review on state-of-the-art deep learning methods for process prediction, Artif. Intell. Rev. 55 (2) (2022) 801–827.

[17] H. Nguyen, M. Dumas, A. H. M. ter Hofstede, M. La Rosa, F. M. Maggi, Business process performance mining with staged process flows, in: CAiSE, Vol. 9694 of LNCS, Springer, 2016, pp. 167–185.

[18] M. Pourbafrani, S. J. van Zelst, W. M. P. van der Aalst, Semi-automated time-granularity detection for data-driven simulation using process mining and system dynamics, in: ER, Vol. 12400 of LNCS, Springer, 2020, pp. 77–91.

[19] A. Polyvyanyy, A. Moffat, L. García-Bañuelos, An entropic relevance measure for stochastic conformance checking in process mining, in: ICPM, IEEE, 2020, pp. 97–104.

[20] M. Reichert, P. Dadam, Adept$_{\text{flex}}$-supporting dynamic changes of workflows without losing control, J. Intell. Inf. Syst. 10 (2) (1998) 93–129.

[21] M. Rosemann, J. Recker, Context-aware process design exploring the extrinsic drivers for process flexibility, in: BPMDS, Vol. 236 of CEUR Workshop Proceedings, CEUR-WS.org, 2006.

[22] H. Schonenberg, R. Mans, N. Russell, N. Mulyar, W. M. P. van der Aalst, Process flexibility: A survey of contemporary approaches, in: CIAO!/EOMAS, Vol. 10 of LNBIP, Springer, 2008, pp. 16–30.

[23] B. Shneiderman, The eyes have it: A task by data type taxonomy for information visualizations, in: VL, IEEE Computer Society, 1996, pp. 336–343.

[24] W. M. P. van der Aalst, V. A. Rubin, H. M. W. Verbeek, B. F. van Dongen, E. Kindler, C. W. Günther, Process mining: a two-step approach to balance between underfitting and overfitting, Softw. Syst. Model. 9 (1) (2010) 87–111.

[25] R. J. Hyndman, G. Athanasopoulos, Forecasting: principles and practice, OTexts, 2018.

[26] R. M. Kil, S. H. Park, S. Kim, Optimum window size for time series prediction, in: EMBS, Vol. 4, IEEE, 1997, pp. 1421–1424.

[27] S. Makridakis, E. Spiliotis, V. Assimakopoulos, Statistical and machine learning forecasting methods: Concerns and ways forward, PloS one 13 (3) (2018) e0194889.

[28] S. Makridakis, E. Spiliotis, V. Assimakopoulos, The M4 competition: 100,000 time series and 61 forecasting methods, Int. Journal of Forecasting 36 (1) (2020) 54–74.

[29] D. D. Thomakos, J. B. Guerard Jr, Naive, ARIMA, nonparametric, transfer function and VAR models: A comparison of forecasting performance, Int. J. Forecast. 20 (1) (2004) 53–67.

[30] H. Lütkepohl, Vector autoregressive models, in: Handbook of Research Methods and Applications in Empirical Macroeconomics, Edward Elgar Publishing, 2013, pp. 139–164.

[31] C. Francq, J.-M. Zakoian, GARCH models: structure, statistical inference and financial applications, John Wiley & Sons, 2019.

[32] J. D. Hamilton, Time series analysis, Princeton University Press, 2020.

[33] R. A. Davis, P. Zang, T. Zheng, Sparse vector autoregressive modeling, J. Comput. Graph. Stat. 25 (4) (2016) 1077–1096.

[34] S. Kriglstein, S. Rinderle-Ma, Change visualizations in business processes – requirements analysis, in: GRAPP/IVAPP, SciTePress, 2012, pp. 584–593.

[35] S. J. J. Leemans, E. Poppe, M. T. Wynn, Directly follows-based process mining: A tool, in: Proceedings of the ICPM Demo Track, 2019, pp. 9–12.

[36] J. E. Hanke, A. G. Reitsch, D. W. Wichern, Business forecasting, 9th Edition, Prentice Hall New Jersey, 2001.

[37] A. S. Weigend, Time series prediction: forecasting the future and understanding the past, Routledge, 2018.

[38] C. Bergmeir, J. M. Benítez, On the use of cross-validation for time series predictor evaluation, Inf. Sci. 191 (2012) 192–213.

[39] S. J. Leybourne, et al., Testing for unit roots using forward and reverse dickey-fuller regressions, Oxf. Bull. Econ. Stat. 57 (4) (1995) 559–571.

[40] S. Seabold, J. Perktold, Statsmodels: Econometric and statistical modeling with python, in: Python in Science Conference, Vol. 57, 2010, p. 61.

[41] F. D. Davis, Perceived usefulness, perceived ease of use, and user acceptance of information technology, MIS Quarterly (1989) 319–340.

[42] V. Venkatesh, M. G. Morris, G. B. Davis, F. D. Davis, User acceptance of information technology: Toward a unified view, MIS Quarterly (2003) 425–478.