# 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[2]

[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

**Abstract.** The surge in event data recorded during the execution of business processes is ever-growing and has spurred an array of data-driven process analytics to support and improve how the business processes are performed. A major strand of such analytics encompasses forecasting how the business processes will unfold into the future, mainly focusing on the next-step, remaining time, and goal-oriented predictions. The granularity of such approaches lies with the individual events in the process. In contrast, this work approaches process forecasting at a larger scale and proposes a technique to forecast the entire process model from the available historical event data. A forecasted model summarises the anticipated business processes and can be used by process analysts to prepare for the upcoming global and longer-term changes, such as impending process drifts or emerging performance bottlenecks. To this purpose, event data is reshaped into multiple time series, each representing an evolution of a behavioural aspect of the process model, and collectively specifying the historical evolution of the model. We then perform time series forecasting to extrapolate the individual series and, concomitantly, the entire model into the future. Experiments confirm that this approach is capable of informing process analysts about the future evolution of the business processes.

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

## 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. Process mining [1] is one of the most fast-growing fields in information systems analysis and encompasses the wide range of techniques performed on event data generated by these systems, including the visualisation, conformance checking, and enhancement of process models generating these events. More recently, predictive process analysis techniques, often referred to as predictive process monitoring, have surfaced to support the prediction of the next event/activity in the process, the remaining process cycle time, or other goal-oriented process outcomes [4]. Various predictive techniques exist, which make use of various analytics architectures, including neural networks [21], stochastic Petri nets [18], and general classification techniques [22].

These techniques, however, focus on a short time horizon or a well-scoped outcome in terms of prediction [22]. Indeed, it is known that the next-step, or the time-to-next-step, prediction often performs poorly for long time horizons [15]. This, consequently, limits the range of insights that can be obtained. A process (model) is ever-changing. Hence, while the 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 [16], including the stability or change of (parts of) the process model such as process drifts, which can inform improvement ideas [24].

This paper suggests a paradigm shift from the typical predictive process analysis techniques that focus on the level of individual outcome, trace, or event to techniques that focus on the entire process model. This is achieved via an aggregation, both at the input and output of forecasting, of directly-follows (DF) relations over all the process activity pairs as a summary of the process model over various time intervals, which are further extrapolated into the future using time series techniques modelling the number of DF occurrences over time. Given that forecasting over long horizons results in a vast amount of information, inspired by the ideas in [8], we propose an approach for tracking the changes in forecasted models and comparing them with AS-IS models. The idea hence marries the concept of traditional process discovery resulting in descriptive models with a typical machine learning approach, which uses training and test sets in order to infer generalising models that can make quality predictions of the future.

This paper makes four contributions:

1. Proposes the first process model forecasting technique grounded in the time series forecasting;
2. Evaluates the quality of the process models forecasted by our technique using standard conformance checking measures from process mining;
3. Argues that the quality of the produced forecasts are mostly in the range of 5-15% in terms of mean absolute percentage error in terms of conformance with actual models, which makes process model forecasting useful for practical applications; and
4. Presents the Process Change Exploration (PCE) system for comparing AS-IS and forecasted process models.

To this purpose, a variety of time series analysis techniques are applied to three industrial event logs using two aggregations that establish suitable time series. A variety of considerations regarding the use of time series analysis for event logs, and the suitability of other predictive algorithms, are discussed. Results show that simple forecasting techniques often perform adequately without extensive parameter tuning.

This paper is structured as follows...

## 2 Motivation

The shift from fine-granular predictive process monitoring 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[4] 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:

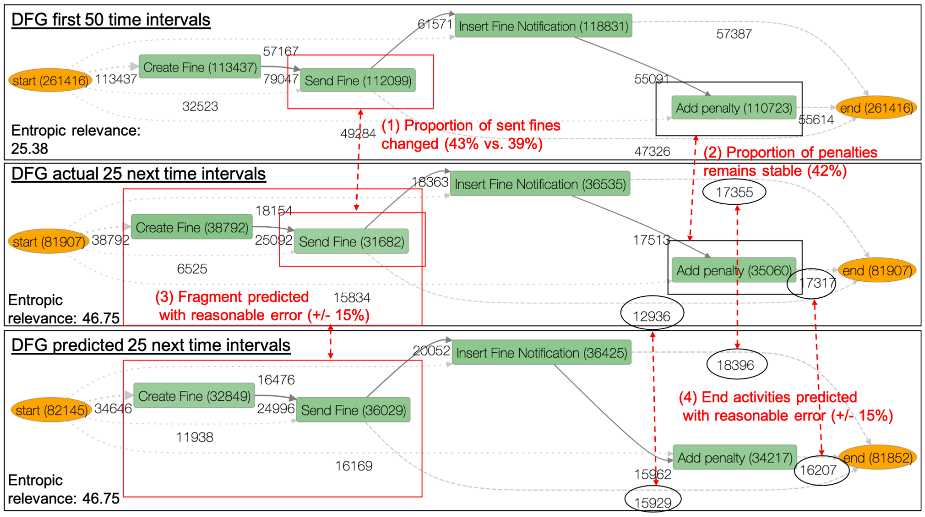


Fig. 1: Directly-follows graphs of the 45 first intervals of the event log, as well as a forecasted and actual DFG of the 25 next intervals.

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

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

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/monitoring focuses on learning the as-is model to reason about a to-be model and suggest potential repairs and improvements, process model forecasting allows to already grasp the future outcomes of the current as-is process, which allows to shortcut potentially wrong outcomes.

We evaluate the forecasts quantitatively using entropic relevance [17] to measure 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. The entropic relevance of the forecasted DFG and the actual future DFG with respect to the test log is the same, suggesting 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 X, suggesting that the future behaviour shifts and the historical DFG does not describe the testing log well enough. Finally, the DFG that describes all and only traces of the testing log with the relative frequencies as they are observed in the log has the entropic relevance with the testing log of Y, the best possible relevance value for this log. As the difference between 4.64 and Y is small, we conclude that the forecasted DFG approximates the actual future behaviour well and much better than the historical DFG.

In this light, we complement the model-level prediction technique with a visualisation system to enable analysts to understand the forthcoming changes to the processes. The authors of [16] discuss the opportunities for process forecasting. They describe that the utility of process forecasting is an understanding of the incremental changes or adaptations that happen to the process model into the future. In terms of visualisation principles, we follow the "Visual Information-Seeking Mantra": *overview first, zoom and filter, then details-on-demand* [20]. 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.

Observe that our process model forecasting technique intrinsically puts forwards an entirely new perspective on predictive process monitoring. The prediction horizon is much longer compared to what existing next-activity prediction models can achieve. Moreover, where next-activity and related predictive algorithms have a strong case-level focus, our technique centres on model-level prediction to obtain a full picture of the future development of a process (model). Please note that process model forecasting could partially assume goal-oriented prediction, as the forecast DFG allows to answer multiple oftentimes used goal statements pertaining to the execution of a particular activity, or a precedence relationship of a particular activity pair [22] at the same time. However, our technique is not intended to "replace" existing techniques, but rather puts forward an entirely new perspective.

## 3 Process model forecasting

This section outlines how directly-follows time series 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 $\Sigma$ 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 $>_L$: $\Sigma \times \Sigma \rightarrow \mathbb{N}$ 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 weights $DFG = (\Sigma, >_L)$.

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 and goal-oriented outcome prediction [21, 22], e.g., predictions at a point in the process rely on prefixes of a certain length, or particular state aggregations [2]. In the proposed forecasting approach, however, time series instead of cross-sectional data will be used. 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. Firstly, 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 [7]. Secondly, 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 [9]. Time series for the DFs $>_{T_{a_1,a_2}} = \langle >_{L_1} (a_1, a_2), \dots, >_{L_s} (a_1, a_2)\rangle, \forall a_1, a_2 \in \Sigma \times \Sigma$ 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 |
|---|---|---|
| $<_{Ls} (A1, A1)$ | (0,1,0) | (1,0,0) |
| $<_{Ls} (A1, A2)$ | (1,1,1) | (1,1,1) |
| $<_{Ls} (A2, A1)$ | (0,1,0) | (0,1,0) |
| $<_{Ls} (A2, A2)$ | (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 $(y_1, ..., y_T)$ [7]. 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 tme series techniques all 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. [13] 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 [14]. 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 [23]. Machine learning models could potentially leverage interrelations between the different DFs but again would require long training times 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 1 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 [7]. 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 [13]. AutoRegressive Integrating Moving Average (ARIMA) models are based on auto-correlations within time series. They combine autoregressions 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 [5]. 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 [5].

### 3.3 Process change exploration

In Section 3.1 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, 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 the task T1, we modify the DFG syntax. In order to display the process model adaptation from time range $T_{i_0,j_0}, i_0 < j_0$ to $T_{i_1,j_1}, i_1 < j_1$ we display the union of the process models of these regions, annotating the places and transitions with the numbers of both ranges. We colour the aDFG as follows: we use colour saturation to show the places with higher values. We colour transitions with a diverging saturation (red-black-green) schema. This colouring applies red colour to transitions that are dominant in the $T_{i_0,j_0}$ range, and green if transitions are dominant in the $T_{i_1,j_1}$ range, otherwise the transition is close to black. For coloring transitions, we used idea of the three colour schema from [10].

**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. Analyst can brush one region in order to zoom in, creating one region of interest $T_{i_0,j_0}, i_0 < j_0$ 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 the red, while the second one is coloured green.

**Activity and path sliders.** We adopt two sliders that are used to simplify the DFG [11] 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 three real-life event logs is reported. The aim of the evaluation is to measure to what extent the forecasted process models/DFGs are capable of correctly reproducing actual future DFGs in terms of their arc weights (measured as deviations or errors), as well as whether they are capable of reproducing the same process model behaviour (measured against the behaviour of the baseline actual DFG). 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. This is done for both the equisized and equitemporal aggregation.

### 4.1 Re-sampling and test setup

In order 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 [6]. 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 $(y_1, \ldots, y_T, \ldots, \hat{y}_{t+h-1})$ [25]. 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 [3]. 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 popular publicly-available event logs are used: the 2012 BPI challenge log[5], the Sepsis cases event log[6], 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.

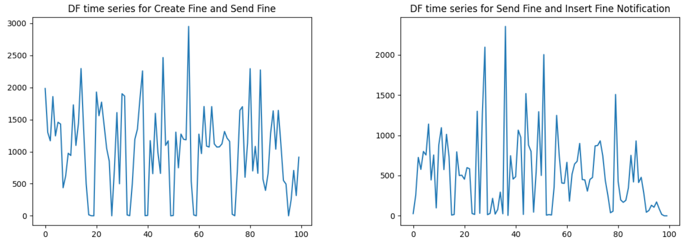
An example of applying the equisized or equitemporal aggregation to the Sepsis event log with 100 intervals results in the DF time series of Figure 2, where the DF occurrences of the most frequently occurring activity pair is included. For the equisized aggregation, the number of DFs is indeed relatively stable over

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

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

| Event log | # cases | # activities | Average trace length |
|---|---|---|---|
| **BPI 12** | 13,087 | 36 | 20.020 |
| **Sepsis** | 1,050 | 16 | 14.490 |
| **RTFMP** | 150,370 | 11 | 3.734 |

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



(a) Most common DF - equisize (b) Most common DF - equitemp

Fig. 2: RTFMP

the log's timeline where for the equitemporal aggregation a noticeable decline of DF pairs is visible towards the end of the series. 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 time line. 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 [12] 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 reported below for comparison with the other time series techniques. The results contain the best-performing representative of each forecasting family.

### 4.2 Evaluation criteria

Given that we want to evaluate the capability of the approach to accurately predict the evolution of the process model, the combination of all DF predictions is considered to obtain a global DFG prediction. The following two criteria are used:

- **Cosine distance:** measures the distance between two vectors and is often used to compare graph distance. This metric is used to compare the DFGs' edge weight matrices between the actual and predicted number of DF relations.

- **Entropic relevance:** see section 2.

These criteria balance a predictive and structural evaluation of the algorithms and report on both the numeric performance common in a forecasting setting as well as their appropriateness in terms of reproducing a structurally usable process model which allows for the observed process behaviour. In both cases a lower score is better.

### 4.3 Results

All pre-processing was done in Python with a combination of *pm4py*[7] and the *statsmodels* package [19]. The code and full results are available here[8]. Figures 3 and 4 contain visualisations of the cosine distance and entropic relevance for the RTFMP log, and Table 4 contains the mean absolute percentage error (MAPE) of the actual and forecasted DFGs in terms of entropic relevance.

| | | **BPI 12** | | | **Sepsis** | | | **RTFMP** | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | **50** | **75** | **100** | **50** | **75** | **100** | **50** | **75** | **100** |
| **equisize** | **nav** | **2.63** | 2.55 | 2.23 | 8.66 | 7.19 | 5.85 | **35.10** | **19.63** | 17.03 |
| | **arima212** | 4.49 | **2.31** | **2.20** | NA | 8.93 | 6.55 | 40.65 | NA | 20.71 |
| | **ar2** | 2.91 | 2.69 | 2.22 | 10.45 | 6.49 | 5.14 | NA | NA | **14.44** |
| | **hw** | 2.71 | 2.41 | 2.22 | 8.25 | 7.52 | 6.20 | 43.06 | 24.09 | 20.20 |
| | **garch** | 3.02 | 3.02 | 2.26 | **7.66** | **6.14** | **4.95** | 37.65 | 22.93 | 15.12 |
| **equitemp** | **nav** | 3.75 | **3.09** | 3.11 | 9.77 | 6.99 | NA | 20.81 | 12.96 | 15.19 |
| | **arima212** | 5.35 | 4.16 | **2.83** | 14.87 | 7.88 | NA | 33.03 | 13.79 | **14.48** |
| | **ar2** | NA | 3.19 | 3.13 | 15.35 | 6.51 | NA | NA | 13.06 | 15.38 |
| | **hw** | 4.10 | 3.21 | 3.06 | **7.98** | 6.08 | NA | 26.15 | 13.29 | 15.77 |
| | **garch** | **3.74** | 3.19 | 3.17 | 23.00 | **5.68** | NA | **19.68** | **11.68** | 14.64 |

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

The MAPE results show that many techniques are capable of staying well below the 5% difference for BPI 12 (except for ARIMA after 50 intervals) with the naive forecasting baseline performing competitively with the slightly outperforming ARIMA(2,1,2) model in some cases. For the Sepsis dataset, results below the 10% line can be achieved by many techniques, with GARCH performing best for equisized and Holt-Winters for equitemporal aggregation. Given the few DF occurrences in the later part of the event log, no results can be produced for the equitemporal aggregation. Finally, the RTFMP only achieves below 15% MAPE for the 100 intervals or equitemporal aggregation. Figures 3 and 4 show that GARCH indeed performs well in terms of low entropic relevance, but not cosine

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

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

distance. Besides, it shows that most techniques perform similarly in terms of entropic relevance.

Overall, there is no clear forecasting technique that stands out over the whole line, while the naive baseline often performs adequately and GARCH reporting best or close-to-best results in at least half the cases. The errors are higher when fewer intervals are used for training with 100 intervals ($ts = 75$) resulting in the lowest MAPE for all three datasets. Nevertheless, even for 75 intervals ($ts = 50$) the results are closer to the 100 than the 50 interval setup. It also appears that equitemporal aggregation results is lower forecasting errors when the training set is small.

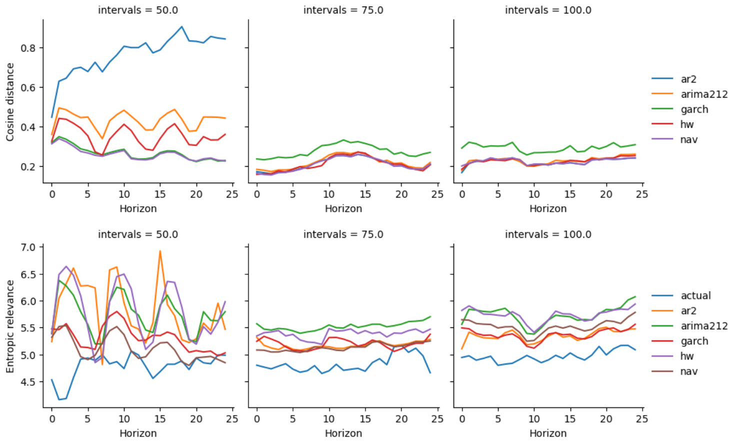


Fig. 3: Results for equisize aggregation for the RTFMP event log.

## 4.4 Managerial implications

Depending on the event log, it is feasible to obtain good (<15%) to very good (<5%) forecasting results in terms of the MAPE and entropic relevance against actual DFGs. The experiments show that the number of intervals used for training vastly impacts results, which is expected given that with 10-fold cross-validation, this results in only 15 interval being used for the tenth fold to predict the next 25 intervals. Given that GARCH models often perform strongly suggests that DF time series benefit from the use of models which allow for a varying level of variance. This makes sense in an event log context, as activities generating the DF occurrences do not occur uniformly during the execution of a process.

## 4.5 Visualising Process Model Forecasts

In Section 4 we evaluated forecasting results, ensuring the relevance of the predicted process models. To that end, gaining actual insights from such predicted

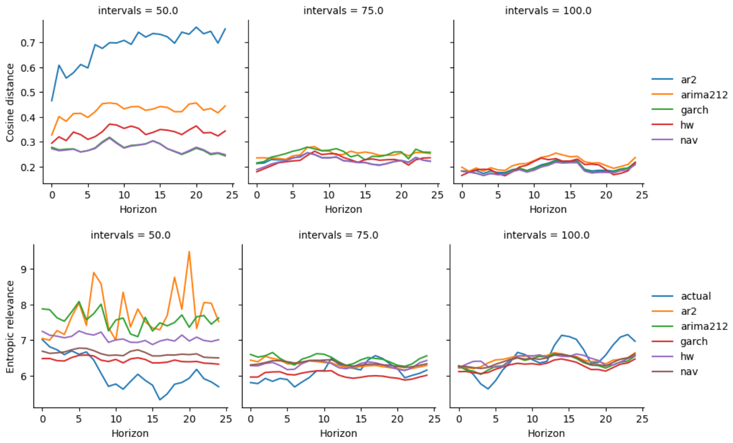


Fig. 4: Results for equitemporal aggregation for the RTFMP event log.

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 5. It shows an interactive visualisation system with several connected views. The system is implemented with D3.js JavaScript library and is available at [9].

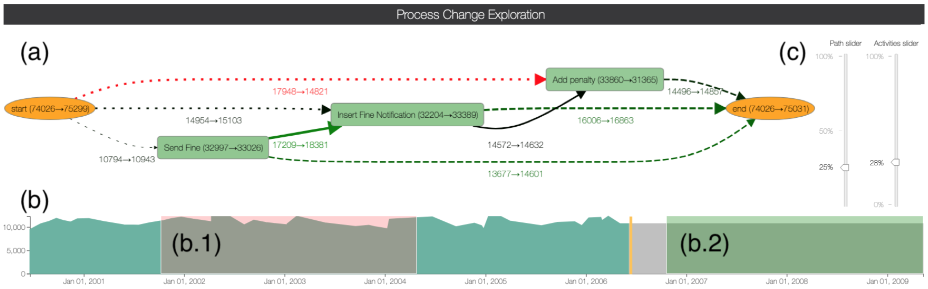


Fig. 5: Process Change Exploration (PCE) system. *(a)* shows *Adaptation Directly-Follows Graph (aDFG)* view. *(b)* shows the *Timeline view with brushed regions* view. Users can brush one or more regions on this graph in order to filter the scope of the analysis *(b.1*, and *b.2)*. Two additional views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

[9] google.com

## 5 Conclusion

In this paper we investigated the potential of using time series forecasting for process model monitoring.

In future research we will cover more intricate forecasting techniques and compare with machine learning-based models. Furthermore, we will perform an extensive prediction confidence interval analysis.

## References

1. van der Aalst, W.: Data Science in Action. In: Process Mining. Springer (2016)
2. 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)
3. Bergmeir, C., Benítez, J.M.: On the use of cross-validation for time series predictor evaluation. Information Sciences 191, 192–213 (2012)
4. 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)
5. Francq, C., Zakoian, J.M.: GARCH models: structure, statistical inference and financial applications. John Wiley & Sons (2019)
6. Hanke, J.E., Reitsch, A.G., Wichern, D.W.: Business forecasting, vol. 9. Prentice Hall New Jersey (2001)
7. Hyndman, R.J., Athanasopoulos, G.: Forecasting: principles and practice. OTexts (2018)
8. Kabicher, S., Kriglstein, S., Rinderle-Ma, S.: Visual change tracking for business process models. In: ER. Lecture Notes in Computer Science, vol. 6998, pp. 504–513. Springer (2011)
9. Kil, R.M., Park, S.H., Kim, S.: Optimum window size for time series prediction. In: Proceedings of the 19th Annual International Conference of the IEEE Engineering in Medicine and Biology Society.'Magnificent Milestones and Emerging Opportunities in Medical Engineering'(Cat. No. 97CH36136). vol. 4, pp. 1421–1424. IEEE (1997)
10. Kriglstein, S., Rinderle-Ma, S.: Change visualizations in business processes - requirements analysis. In: GRAPP/IVAPP. pp. 584–593. SciTePress (2012)
11. Leemans, S., Poppe, E., Wynn, M.: Directly follows-based process mining: A tool. In: Proceedings of the ICPM Demo Track 2019 (CEUR Workshop Proceedings, Volume 2374). Vol. 2374. pp. 9–12. Sun SITE Central Europe (2019)
12. 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)
13. Makridakis, S., Spiliotis, E., Assimakopoulos, V.: Statistical and machine learning forecasting methods: Concerns and ways forward. PloS one 13(3), e0194889 (2018)
14. Makridakis, S., Spiliotis, E., Assimakopoulos, V.: The m4 competition: 100,000 time series and 61 forecasting methods. International Journal of Forecasting 36(1), 54–74 (2020)
15. Park, G., Song, M.: Predicting performances in business processes using deep neural networks. Decision Support Systems 129, 113191 (2020)

16. 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)
17. Polyvyanyy, A., Moffat, A., García-Bañuelos, L.: An entropic relevance measure for stochastic conformance checking in process mining. In: ICPM. pp. 97–104. IEEE (2020)
18. 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)
19. Seabold, S., Perktold, J.: Statsmodels: Econometric and statistical modeling with python. In: Proceedings of the 9th Python in Science Conference. vol. 57, p. 61. Austin, TX (2010)
20. Shneiderman, B.: The eyes have it: A task by data type taxonomy for information visualizations. In: VL. pp. 336–343. IEEE Computer Society (1996)
21. Tax, N., Verenich, I., Rosa, M.L., Dumas, M.: Predictive business process monitoring with LSTM neural networks. In: CAiSE. Lecture Notes in Computer Science, vol. 10253, pp. 477–492. Springer (2017)
22. 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)
23. 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)
24. Van Eck, M.L., Lu, X., Leemans, S.J., Van Der Aalst, W.M.: PM$^2$: a process mining project methodology. In: International Conference on Advanced Information Systems Engineering. pp. 297–313. Springer (2015)
25. Weigend, A.S.: Time series prediction: forecasting the future and understanding the past. Routledge (2018)