Process Model Forecasting Using Time Series

Johannes De Smedt[1], Artem Polyvyanyy[2], and Jochen De Weerdt[1]

[1] Department of Decision Sciences and Information Management
Faculty of Economics and Business, KU Leuven, Leuven, Belgium
{johannes.desmedt;jochen.deweerdt}@kuleuven.be
[2] School of Computing and Information Systems
The University of Melbourne, Australia
artem.polyvyanyy@unimelb.edu.au

Abstract. The surge in event-based data that is recorded during the execution of business processes is every-growing and has spurred an array of process analytics techniques to support and improve information systems. A major strand of process analytics encompasses the prediction of the process' future development, mostly focusing on next-step, remaining time, or goal-oriented prediction. The granularity of such approaches lies with the events in the process. This work approaches a process at the abstraction level of the process model, i.e., rather than fine-granular next-step prediction, the process' global behaviour is extrapolated into the future. To this purpose, event data is captured at various intervals and aggregated in the form of directly-follows graphs. Each activity pairs' relation within the graph is monitored over these intervals and their future values predicted using common time series techniques. Experiments show that these techniques are already well-capable of informing process analysts with the future status of the process.

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

1 Introduction

An initial outline of how process model forecasting can be achieved was presented by [10].

Various predictive techniques exist, many coined under the term of predictive process monitoring []...

Contributions:

- Propose the first process model-wide forecasting technique, which fits the the broader area of predictive process monitoring.
- To this purpose, a variety of time series analysis techniques are applied to 3 real-life event logs with two event log aggregations that establish suitable time series. They predict the level of directly-follows occurrences for activity pairs over time.
- A variety of considerations regarding the use of time series analysis for event logs are discussed, as wel as the suitability of other predictive algorithms.

- Results show that simple forecasting techniques often already perform adequately without extensive parameter tuning. More intricate models often fail to report consistent performance.

2 Preliminaries

An event log L 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 powerset over the alphabet of activities Σ which exist in the information system $\langle e_1, ..., e_{|\sigma|} \rangle \subseteq \Sigma^*$. 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 (DF) 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 [12, 13], e.g., predictions at a point in the process rely on prefixes of a certain length, or particular state aggregations [1]. In the proposed forecasting approach, however, not cross-sectional but time series data will be used. Hence, the evolution of the DFGs will be monitored over intervals of the log where multiple aggregations are possible:

- Equitemporal aggregation: each sublog contains a part of the event log of equal time duration. This can lead to sparesely populated sublogs when the events' occurrences are not uniformly spread over time, however, is easy to apply (on new traces).
- Equisized aggregation: each sublog cotnains a part of the event log of similar DF sum. This leads to well-populated sublogs, however, might be harder to apply when new data does not contain sufficient new DF occurrences.

Time series can be obtained for all $>_{Ls}$, $Ls \subseteq L$ by applying the aforementioned aggregations. Tables 1 and 2 provide an example of both.

3 Methodology

This section outlines the forecasting techniques that will be used, as well as their connection with time series extracted from event logs.

3.1 Time series techniques

To model the time series of DFGs, various algoritms are 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 $(y_1, ..., y_T)$ [6]. A

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)).

wide array of time series techniques exist, ranging from simple models such as naive forecasts over to more advanced approaches such as exponential smoothing and autoregressive models. Many also exist in a seasonal variant due to their application in contexts such as sales forecasting. Below, the most-widely used techniques are formalised.

The naive forecast simply uses the last value of the time series T as its prediction:

$$\hat{y}_{T+h|T} = y_T$$

A Simple Exponential Smoothing (SES) model uses a weighted average of past values where their important exponentially decays as they are further into the past:

$$\hat{y}_{T+h|T} = \alpha y_T + (1-\alpha)\hat{y}_{T|T-1}$$

with $\alpha \in [0,1]$ a smoothing parameter where lower values of α allow for focusing on the more recent values more. Holt's models introduce a trend in the forecast, meaning the forecast is not flat:

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

with $l_t = \alpha y_t + (1-\alpha)(l_{t-1} + b_{t-1})$ modelling the overall level of the time series and $b_t = \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1}$ modelling the trend over the series where α is as before and β the smoothing parameter for the trend. Exponential smoothing models often perform very well despite their simple setup.

AutoRegressive Integrating Moving Average (ARIMA) models are based on autocorrelations within time series. The combine autoregressions with a moving average over error terms.

An AutoRegressive (AR) model of order p uses the past p values in the time series and apply a regression over them. It can be written as:

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

where all ϕ_i, $i \in [1, p]$ can assign different weights to different time lags.

A Moving Average (MA) model of order q can be written as:

$$y_t = c + \epsilon_t + \phi_1 \epsilon_{t-1} + \dots + \phi_q \epsilon_{t-q}$$

with $\phi > 0$ a smoothing parameter, and ϵ_t again a white noise series, hence the model 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. An ARIMA(p, d, q) model can be written as:

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

with y'_t a differenced series of order d. Forecasting is possible by introducing $T+h$ to the equation and iteratively obtaining results by starting off with $T+1$. 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 [3]. 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 over time as follows:

$$\sigma_t^2 = \alpha_0 + \alpha_1 y_{t-1}^2 + \dots + \alpha_q y_{t-q}^2$$

is the variance over time with $\alpha_0 > 0$ and $\alpha_i \geq 0$, $i \in [1, q]$ smoothing parameters, and $y_t = \sigma_t \epsilon_t$ where $\epsilon_t \overset{i.i.d}{\sim} (\mu = 0, \sigma^2 = 1)$. This can be used to capture that the variance gradually increases over time, or has 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:

$$\sigma_t^2 = \alpha_0 + \alpha_1 y_{t-1}^2 + \dots + \alpha_q y_{t-q}^2 + \dots + \beta_1 \sigma_{t-1}^2 + \dots + \beta_p \sigma_{t-p}^2$$

with $\alpha_0 > 0$ and $\alpha_i, \beta_i \geq 0$, $i > 0$.

(G)ARCH models often outperform ARIMA models in contexts such as the prediction of financial indicators of which the variance often changes over time [3].

3.2 Forecasting Directly Follows series

The trade-off exists between approaching DFGs as a multivariate collection of DF time series, or treating each DF separately. The aforementioned time series techniques all use univariate data in contrast with, e.g., Vector AutoRegression (VAR) models, or 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. [8] 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 [9]. 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 [14]. 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.

4 Experimental evaluation

In this section, an experimental evaluation over X real-life event logs is reported.

4.1 Re-sampling and test setup

Time series are obtained by specifying a number of intervals (i.e. time steps in the DF series) using either equitemporal or equidistant aggregation. Time series algorithms are parametric and sensitive to sample size requirements [5]. Depending on the number of parameters a models 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, 50 and 100 intervals will be used.

Three widely-used event logs are used (e.g. see []): the 2012 BPI challenge log[3], the Sepsis cases event log[4], and the Road Traffic Fine Management Process log[5] (RTFMP) event log. Each of these logs has a diverse set of chacteristics 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 equisize or equitemporal aggregation to the event logs with 100 intervals results in the DF time series of Figures 1 to 3 where the most frequently and 50th most frequently occurring activity pairs are included. The equitemporal aggregation is based on a number of intervals over the whole timespan of the event log. When using this aggregation, a noticeable decline of DF pairs is visible towards the end of the series. This phenomenom is typical

[3]https://doi.org/10.4121/uuid:3926db30-f712-4394-aebc-75976070e91f
[4]https://doi.org/10.4121/uuid:915d2bfb-7e84-49ad-a286-dc35f063a460
[5]https://doi.org/10.4121/uuid:270fd440-1057-4fb9-89a9-b699b47990f5

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

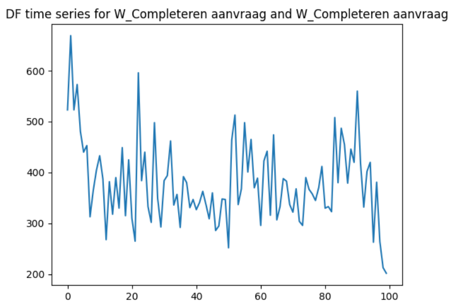
Table 3: Overview of the characteristics of the event logs used in the experimental evaluation.

in event logs, as processes typically have particular endpoint activity, e.g., the closure of a loan application event in the BPI 12 log. The use of a cutoff of 50 for the most-frequently occurring DF pairs results in different time series as well. For the BPI 12 log, the frequency of the 50th pair is still relatively high, while for the other event logs the frequency of the DF pair is low and close to 0, making the series unsuitable for analysis with white noise series analysis techniques that assume stationarity. Ideally, every time series is tested using a stationairty test such as the Dickey-Fuller unit root test [7] and establish an appropriate lag order for differencing. Furthermore for each algorithm, especially ARIMA-based models, (partial) autocorrelation could establish the ideal p and q parameters. However, for the sake of simplicity and to avoid tedious solutions where each activity pair has to have different parameters, various values are used 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.

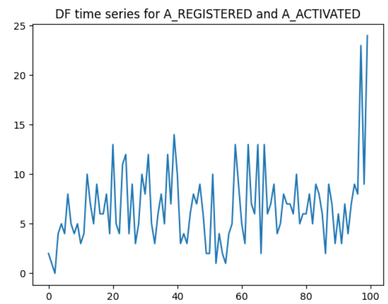
Resampling is based on a 10-fold cross-validation constructed following a rolling window approach for various horizon values $h \in [1, 5]$ where a recursive strategy is used to iteratively obtain $\hat{y}_{t+h|T_{t+h-1}}$ with $(y_1, \dots, y_T, \dots, \hat{y}_{t+h-1})$ [15]. The 10 training sets exist from $(y_1, \dots, y_{T-h-f})$ and the test sets from $(y_{T-h-f+1}, \dots, y_{T-f})$ with $f \in [0, 9]$ the fold index [2]. 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.

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 to obtain a global DFG prediction is considered. The following two criteria are used:

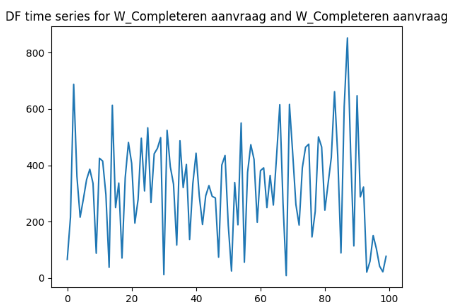
- Graph edit distance (GED) [4]: measures the similarity between two graphs based on the number of operations required to render the graphs isomorphic using edge and node insertions, deletion, or substitution. In this evaluation, it is used to check whether the forecasts are at least capable of returning similar structures by discerning whether a DF edge should exist between two activities regardless of how many directly follows relations occur.
- Cosine distance: measures the distance between two vectors and are often used to compare graph distance . This metric is used to compare the DFG's edge weight matrices between the actual and predicted number of DF relations.



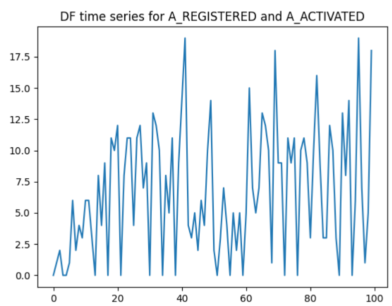
(a) Most common DF - equisize



(b) 50th most common DF - equisize



(c) Most common DF - equitemp



(d) 50th most common DF - equitemp

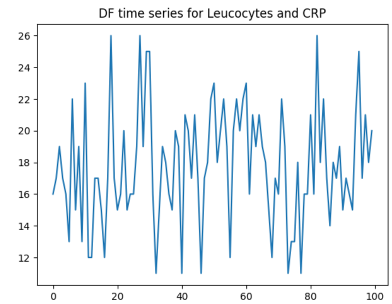
Fig. 1: BPI 12

4.2 Results

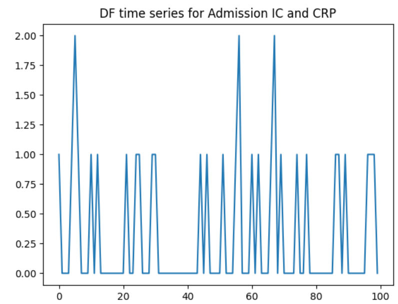
All pre-processing was done in Python with a combination of pm4py[6] and the statsmodels package [11]. The code is available

The cosine distance is reported in Tables 4 to 7 where grayscale is used per event log to indicate the relative performance of each algorithm. Generally, it can be found that naive forecasts are performing the worst overall, meaning that the DF time series forecasting does enjoy performance gains from a more intricate use of autocorrelations and/or decomposition. Both decomposition techniques (SES and HW) perform similarly and adequately and consistently report the best result for each dataset for both interval and aggregation settings. The same holds for simple AR models, where the lag-parameter q occassionally offers slightly different results although this does not seem to be related with the number of intervals. Results for ARIMA and GARCH models are mixed. For ARIMA, the 3 best performing parameter sets are reported (with q and p used with values [1,4] and d with values [0,1,2]). They do not seem to be capable of outperforming

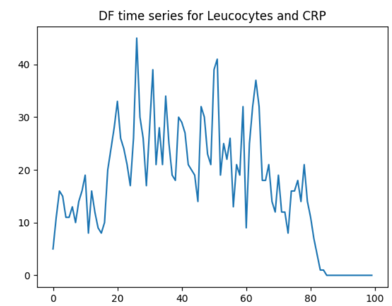
[6]https://pm4py.fit.fraunhofer.de



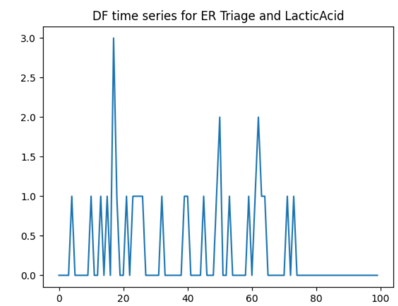
(a) Most common DF - equisize



(b) 50th most common DF - equisize



(c) Most common DF - equitemp



(d) 50th most common DF - equitemp

Fig. 2: Sepsis

simpeler AR models. Overall, the simplest models seem to perform best and do not require extensive parameter tuning. This might indicate that the time series do not exhibit intricate autocorrelations (ARIMA) and/or changing variance over time (GARCH).

Finally, it can be noted that the cosine distance is slightly higher for longer horizon values, especially for the BPI 12 event log for which Figure 1 showed stronger activity towards the end of the series still. This phenomenom is typical of forecasting results as such predictions accumulate forecast errors.

References

1. 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)
2. Bergmeir, C., Benítez, J.M.: On the use of cross-validation for time series predictor evaluation. Information Sciences 191, 192–213 (2012)
3. Francq, C., Zakoian, J.M.: GARCH models: structure, statistical inference and financial applications. John Wiley & Sons (2019)

	h	AR (1)	AR (2)	AR (4)	ARIMA (2,1,1)	ARIMA (2,1,2)	ARIMA (4,1,1)	GARCH	HW	NAÏVE	SES
RTFMP	1	1.873	1.819	1.892	1.920	1.956	1.989	2.553	1.804	2.821	1.804
	2	2.047	1.970	2.028	2.125	2.141	2.172	3.590	2.017	2.998	2.017
	3	1.965	1.907	1.928	1.973	1.998	2.031	3.705	1.963	3.082	1.963
	4	2.005	1.931	1.957	2.001	2.017	2.068	3.807	2.034	2.896	2.034
	5	2.060	1.993	1.986	2.092	2.081	2.103	3.946	2.084	2.775	2.084
BPI12	1	0.216	0.212	0.218	0.242	0.244	0.244	0.225	0.235	0.398	0.235
	2	0.224	0.229	0.238	0.273	0.283	0.268	0.226	0.243	0.365	0.243
	3	0.229	0.228	0.240	0.281	0.294	0.284	0.231	0.248	0.383	0.248
	4	0.416	0.410	0.423	0.481	0.474	0.466	0.416	0.437	0.608	0.437
	5	0.697	0.681	0.690	0.737	0.715	0.703	0.693	0.691	0.871	0.691
Sepsis	1	0.996	1.023	1.060	1.088	1.093	1.107	0.980	0.981	1.727	0.981
	2	0.940	0.965	1.003	1.039	1.005	1.030	0.946	0.955	1.743	0.955
	3	0.967	0.975	0.990	1.055	1.018	1.037	0.974	0.973	1.775	0.973
	4	0.938	0.940	0.949	1.023	0.995	1.040	0.940	0.943	1.719	0.943
	5	0.936	0.937	0.952	0.996	0.976	0.996	0.932	0.939	1.564	0.939

Table 4: Cosine distance results for $h = 5$ and 50 intervals with equisized aggregation.

	h	AR (1)	AR (2)	AR (4)	ARIMA (2,1,1)	ARIMA (2,1,2)	ARIMA (4,1,1)	GARCH	HW	NAÏVE	SES
RTFMP	1	2.232	2.177	2.105	2.334	2.318	2.288	2.602	2.515	2.721	2.515
	2	2.459	2.426	2.352	2.554	2.532	2.499	2.499	2.459	3.807	2.459
	3	2.540	2.489	2.426	2.608	2.586	2.577	2.540	2.535	4.175	2.535
	4	2.497	2.488	2.438	2.533	2.515	2.553	2.518	2.464	4.188	2.464
	5	2.685	2.693	2.655	2.724	2.722	2.723	2.743	2.651	4.222	2.651
BPI12	1	0.427	0.429	0.435	0.457	0.464	0.459	0.420	0.447	0.626	0.447
	2	0.504	0.503	0.504	0.533	0.538	0.535	0.496	0.527	0.837	0.527
	3	0.768	0.763	0.762	0.795	0.799	0.811	0.763	0.794	1.008	0.794
	4	1.025	1.017	1.011	1.055	1.067	1.094	1.016	1.056	1.401	1.056
	5	1.304	1.288	1.264	1.287	1.300	1.350	1.291	1.297	1.582	1.297
Sepsis	1	1.169	1.176	1.201	1.220	1.211	1.247	1.185	1.173	1.880	1.173
	2	1.169	1.175	1.214	1.228	1.237	1.267	1.171	1.169	2.134	1.169
	3	1.165	1.162	1.203	1.227	1.232	1.269	1.172	1.176	2.185	1.176
	4	1.177	1.179	1.216	1.220	1.249	1.271	1.185	1.177	2.108	1.177
	5	1.205	1.203	1.212	1.248	1.257	1.249	1.214	1.213	2.178	1.213

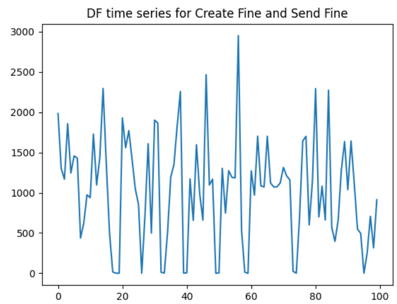
Table 5: Cosine distance results for $h = 5$ and 100 intervals with equisized aggregation.

		AR			ARIMA						
	h	**(1)**	**(2)**	**(4)**	**(2,1,1)**	**(2,1,2)**	**(4,1,1)**	**GARCH**	**HW**	**NAÏVE**	**SES**
RTFMP	**1**	1.884	1.917	1.877	1.896	1.965	1.966	2.011	1.898	2.494	1.898
	2	1.864	1.877	1.815	1.871	1.903	1.970	1.892	1.857	3.276	1.857
	3	1.904	1.881	1.815	1.874	1.865	2.025	1.937	1.888	2.953	1.888
	4	2.026	1.999	1.841	1.980	1.923	2.072	2.039	1.996	2.738	1.996
	5	2.239	2.214	2.175	2.229	2.246	2.398	2.219	2.278	3.469	2.278
BPI12	**1**	0.313	0.280	0.277	0.309	0.304	0.304	0.291	0.332	0.561	0.332
	2	0.398	0.402	0.402	0.411	0.414	0.422	0.402	0.402	0.467	0.402
	3	0.688	0.634	0.645	0.674	0.672	0.693	0.678	0.682	1.032	0.682
	4	0.949	0.901	0.904	0.891	0.905	0.901	0.920	0.857	0.952	0.857
	5	1.212	1.091	1.095	1.097	1.110	1.094	1.177	1.102	1.488	1.102
Sepsis	**1**	0.887	1.020	1.102	1.206	1.362	1.358	0.899	1.091	1.508	1.091
	2	0.727	0.832	0.911	1.070	1.219	1.201	0.771	0.909	1.514	0.909
	3	0.637	0.665	0.732	0.907	1.168	1.022	0.681	0.787	1.421	0.787
	4	0.598	0.601	0.659	0.871	1.064	0.947	0.639	0.720	1.464	0.720
	5	0.541	0.526	0.570	0.777	1.017	0.862	0.580	0.636	1.232	0.636

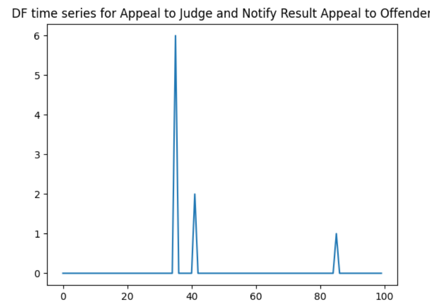
Table 6: Cosine distance results for $h = 5$ and 50 intervals with equitemporal aggregation.

		AR			ARIMA						
	h	**(1)**	**(2)**	**(4)**	**(2,1,1)**	**(2,1,2)**	**(4,1,1)**	**GARCH**	**HW**	**NAÏVE**	**SES**
RTFMP	**1**	1.833	1.814	1.969	1.927	1.945	2.062	1.863	1.801	2.457	1.801
	2	1.893	1.806	1.879	1.855	1.718	1.879	1.822	1.965	2.855	1.965
	3	1.844	1.782	1.823	1.799	1.601	1.828	1.778	2.049	3.305	2.049
	4	1.848	1.803	1.802	1.847	1.574	1.759	1.806	2.093	3.229	2.093
	5	2.057	2.031	2.016	2.051	1.855	2.015	2.042	2.244	3.254	2.244
BPI12	**1**	1.179	1.275	1.155	1.190	1.235	1.133	1.182	1.118	1.282	1.118
	2	1.432	1.566	1.310	1.445	1.443	1.259	1.427	1.327	1.654	1.327
	3	1.619	1.592	1.341	1.437	1.426	1.278	1.603	1.470	1.839	1.470
	4	1.759	1.720	1.378	1.551	1.498	1.352	1.741	1.558	1.405	1.558
	5	1.934	1.916	1.810	1.784	1.791	1.802	1.916	1.697	2.002	1.697
Sepsis	**1**	0.126	0.112	0.187	0.242	0.346	0.282	0.393	0.395	0.000	0.395
	2	0.171	0.111	0.172	0.278	0.401	0.336	0.396	0.397	0.000	0.397
	3	0.221	0.129	0.169	0.313	0.440	0.374	0.400	0.397	0.000	0.397
	4	0.259	0.158	0.157	0.321	0.438	0.418	0.406	0.396	0.000	0.396
	5	0.286	0.189	0.156	0.325	0.450	0.434	0.411	0.394	0.000	0.394

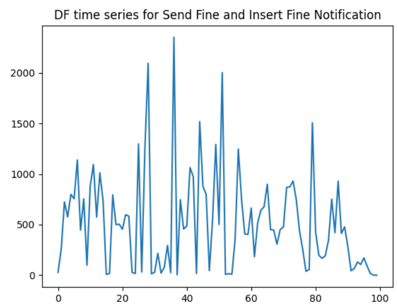
Table 7: Cosine distance results for $h = 5$ and 100 intervals with equitemporal aggregation.



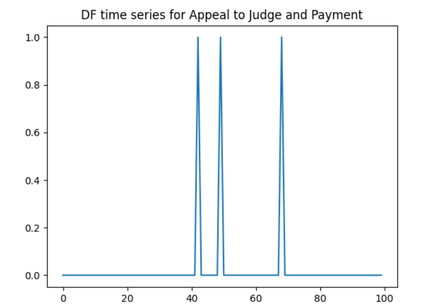
(a) Most common DF - equisize



(b) 50th most common DF - equisize



(c) Most common DF - equitemp



(d) 50th most common DF - equitemp

Fig. 3: RTFMP

4. Gao, X., Xiao, B., Tao, D., Li, X.: A survey of graph edit distance. Pattern Analysis and applications 13(1), 113–129 (2010)
5. Hanke, J.E., Reitsch, A.G., Wichern, D.W.: Business forecasting, vol. 9. Prentice Hall New Jersey (2001)
6. Hyndman, R.J., Athanasopoulos, G.: Forecasting: principles and practice. OTexts (2018)
7. 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)
8. Makridakis, S., Spiliotis, E., Assimakopoulos, V.: Statistical and machine learning forecasting methods: Concerns and ways forward. PloS one 13(3), e0194889 (2018)
9. 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)
10. 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)
11. 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)

12. 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)
13. 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)
14. 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)
15. Weigend, A.S.: Time series prediction: forecasting the future and understanding the past. Routledge (2018)