# Making Sense of Continuous Process Dynamics

Sana Dodangeh[1,2], Martin Kabierski[1], and Han van der Aa[1]

[1] Faculty of Computer Science, University of Vienna, Austria
{sana.dodangeh,martin.kabierski,han.van.der.aa}@univie.ac.at
[2] UniVie Doctoral School Computer Science DoCS, University of Vienna, Austria

**Abstract.** Operating in dynamic and often unpredictable environments, business processes are exposed to changes that cause their execution to fluctuate over time. Since changes are inseparable parts of process execution, capturing them is crucial for gaining reliable insights into how they behave during their execution. Although prior research has studied continuous fluctuations of processes, most studies have focused primarily on visual representations of such changes. However, drawing more actionable insights requires a structured analysis approach that goes beyond visualization. To this end, this paper explores the potential of leveraging established time series analysis techniques—such as decomposition and correlation analysis—to derive meaningful insights into such changes. We evaluate our approach through backward-looking and forward-looking analysis: the former demonstrates the type of insights it can provide into continuous process dynamics by applying it to multiple event logs, while the latter examines whether the approach is beneficial for a downstream process mining task, the prediction of future process behavior.

**Keywords:** process dynamics · time series analysis · continuous fluctuations.

## 1 Introduction

Business processes are typically analyzed using data extracted from information systems that record their execution. These event logs form the foundation of process mining techniques, which aims to understand how processes behave in reality. By analyzing such data, process mining provides valuable insights into process performance and behavioral patterns, supporting various applications [1].

The analysis of such data needs to consider the inherently dynamic nature of business processes, which continuously change over time [2, 3] due to intentional and unintentional adaptations, and are influenced by environmental changes, technological progress, seasonal trends, and other factors [4–6]. A concrete example can be found in the healthcare domain: A hospital check-in process is adjusted annually at the beginning of each flu season by modifying procedures to adapt to this season [7]. Failing to take into account such systematic, non-random process changes during analysis will distort analysis outcomes, leading to inaccurate and unreliable insights, and, ultimately, suboptimal decision-making [8].

Given the dynamic nature of processes, prior work has recognized the need to capture such continuous changes over time. However, this has so far only been addressed through visualization-based approaches [5, 9, 10]. These approaches typically depict the progression of process characteristics over time to reveal phenomena such as gradual changes and recurring patterns. While such approaches provide initial insights, they fall short in two key analytical aspects:[1]

1. They fail to separate systematic, structural changes from random fluctuations, rendering interpretation ambiguous, incomplete, or outright false.
2. They are unable to capture interrelations between process characteristics, making it difficult to understand how changes in one relate to others.

To tackle these limitations, this paper explores the potential of leveraging established time series analysis techniques to gain deeper insights into the continuous dynamics of processes. Recognizing that understanding such dynamics is key to making informed adjustments and decisions in process management, we operationalize these time series analysis techniques for process mining through a structured, three-step algorithmic approach that focuses on (i) the derivation of time series for different process perspectives, (ii) the in-depth analysis of individual time series through decomposition and quantitative characterization (tackling Limitation 1 above), and (iii) the analysis of interrelations between time series through correlation analysis (addressing Limitations 2). This approach provides business analysts with structured information for interpreting continuous process dynamics, supporting downstream tasks in process mining such as analyzing past process dynamics and predicting future process behavior [6]. We evaluate the proposed approach through two complementary analyses. A backward-looking analysis evaluates how the approach supports understanding systematic process dynamics and interrelations between process characteristics. A forward-looking analysis then evaluates how separating these dynamics can support the prediction of future process behavior.

The remainder of this paper is structured as follows: Section 2 presents the motivation behind this work, while Section 3 introduces the proposed algorithmic approach. Section 4 reports on our evaluation. Section 5 provides an overview of related work, and finally, Section 6 concludes the paper.

## 2 Motivation

To motivate our work, we consider current practices for gaining insights into continuous process dynamics. This is so far achieved in a visual manner, especially through time series visualization [10] and drift maps [5]. Figure 1 shows an example of the former idea, highlighting the progression of the case load for the *Road Traffic Fine Management* event log.[2] It shows that the number of new and active cases exhibit temporal fluctuations, with an apparent seasonal pattern. While such visualizations are useful for revealing that process characteristics change over time, deriving more concrete insights from them requires

[1] We illustrate and elaborate on these limitations in Section 2.
[2] Available on `https://data.4tu.nl/categories/13500?categories=13503`

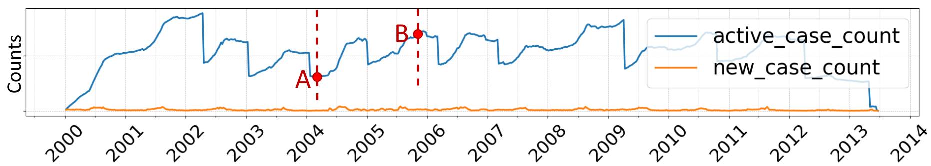



Fig. 1: Time series of new and active case counts in the *Road Traffic Fine Management* process.

analysts to manually inspect and interpret the observed patterns. This makes the analysis highly dependent on visual judgment, which may lead to ambiguous or inaccurate interpretations. Specifically, we recognize that purely visual approaches lack two key aspects that are necessary to properly understand the dynamics of processes:

**C1: Separating systematic from random fluctuations.** Process dynamics can be influenced by various factors, including seasonal patterns, long-term trends, and irregular fluctuations. The interplay of such factors makes interpretations of visualizations particularly problematic, and ignoring these may yield misleading results. For instance, points A and B in Figure 1 mark time points with different active case counts. A direct comparison shows an increase in active cases from point A to B. However, from the visualization alone, it remains unclear whether this increase mainly reflects a persistent upward movement in the underlying trend or the position of the two points within a recurring seasonal pattern. Furthermore, while it is also apparent that not all fluctuations are due to seasonality, it is unclear to what extent factors such as seasonality, long-term trends, or inherent randomness affect the case count, and whether such patterns are consistent over time. All of these insights are critical for properly understanding how a process behaves (and can be expected to behave) over time. Therefore, to support more accurate analysis, it is necessary to systematically separate fluctuations (e.g., seasonality and trends) from changes that occur in a more ad-hoc manner. Moreover, understanding such dynamics is not only important for interpreting past process behavior, but also for some downstream tasks such as predicting how processes may evolve in the future.

**C2: Identifying interrelated patterns.** The comprehensive understanding of process dynamics requires insight into the interrelations of fluctuations across different aspects. For example, Figure 1 suggests that increases in case arrivals tend to coincide with higher workloads (i.e., active cases). However, the strength and consistency of this correlation—and thus the extent to which workload aligns with variations in arrival rates—are difficult to assess through visual inspection alone. Similarly, analysts may want to investigate whether periods of increased workload tend to coincide with variations in process performance, for instance, changes in cycle times. This indicates a clear need for analytical support in identifying and quantifying the interrelations between different process dynamics.

## 3 Algorithmic Approach

This section investigates how established time series analysis techniques can be integrated into the analysis of process dynamics, thereby addressing the challenges of purely visual approaches, laid out in Section 2. In particular, we propose a three-step algorithmic approach for sense-making of continuous process dynamics, which is illustrated in Figure 2. Beginning with an event log, the approach consists of the following steps:

1. *Derive time series* that capture the temporal behavior of a range of process characteristics of interest.
2. *Analyze individual time series* by extracting statistical summaries and decomposing the series to separate structured from irregular fluctuations.
3. *Analyze the interrelations among time series* to uncover dependencies and related changes that reveal broader behavioral structures.

These steps result in visualizations of decomposed time series, and statistical summaries and dependency measures of process characteristics. Together, they provide analysts with insights into behavioral patterns, irregular fluctuations, and interrelations between different process characteristics.

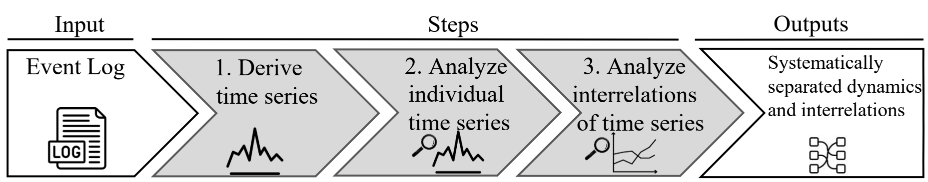



Fig. 2: Overview of the approach for obtaining insights into process dynamics

### 3.1 Derive Time Series

First, the approach transforms the event log into a set of time series, each capturing how a specific process characteristic of interest varies over time. This preserves their temporal evolution and enables analysis of individual series and their interrelations. We operationalize this step based on existing works [10,11].

**Input.** The approach takes an *event log* $L$ as input, defined as a collection of recorded events. Each event $e \in L$ is a tuple $e = (id, act, ts, D)$, where $id$ is the case ID, $act$ the executed activity, $ts$ the recorded timestamp, and $D$ a set of attribute-value pairs capturing data attributes. A trace $\pi = \langle e_1, ..., e_n \rangle$ is a sequence of events that belong to the same case, ordered by their timestamps.

**Process characteristics.** The approach constructs time series for a user-defined set $C$ of process characteristics. A characteristic $c \in C$ can be defined as any aspect of a process that can be measured for a given time window (see also DyloPro for an overview [10]), such as new or active case counts, frequencies of activities, the number of active resources, or the average cycle time of cases. The exact selection for $C$ depends on the analysis goals of a user.

**Windowing.** We segment the timeline of the event log $L$ into equal-length, non-overlapping time windows using time-based tumbling windowing [12, p.354], which results in a sequence of disjoint windows $W_l = \langle w_1, \ldots, w_n \rangle$, where each window $w_t$ covers a fixed duration $l$ (e.g., a day). Each event $e \in L$ is assigned to a window based on its timestamp.

**Time series construction.** For each characteristic $c \in C$, we construct a time series $y_c = \{y_{c,t} \mid t \in \{1, \ldots, |W_l|\}$, where $y_{c,t} \in \mathbb{R}$ reflects the behavior of characteristic $c$ within window $w_t$. Examples are shown in Figure 1. We refer to the collection of time series of all considered characteristics as $Y_C$.

**Warm-up and cool-down elimination.** Event logs may contain incomplete case executions at the beginning or end of the observation period, which leads to distorted observations in the resulting time series of a process characteristic $c \in C$ (i.e., warm-up and cool-down, respectively). To detect such phases, we follow an existing warm-up and cool-down elimination procedure proposed in prior work [11]. In particular, we use the time series of active case counts, as it reflects the overall workload of the process over time. From the distribution of its values, we compute a percentile threshold $\delta$. Subsequently, windows in $W_l$ at the beginning and end of the active case count time series whose values fall below this threshold are identified as warm-up or cool-down periods. These windows are then removed from all time series in $Y_C$ before further analysis.

### 3.2 Analyze Individual Time Series

The goal of this step is to support understanding the dynamics of each process characteristic. To this end, each of the time series in $Y_C$ is analyzed in isolation. The step first uses descriptive statistics to summarize the overall variability of the series and then applies decomposition to separate systematic from random fluctuations through decomposition, thereby addressing C1 in Section 2.

**General statistics.** We first obtain a high-level overview of the magnitude and variability of each process characteristic. For this, we compute descriptive statistics such as the *mean*, *standard deviation*, and *coefficient of variation* (CV), defined as the ratio of the standard deviation to the mean.

**Time series decomposition.** To better understand the temporal dynamics of each process characteristic and distinguish the sources of variation discussed in C1, the approach breaks down the time series into components that capture distinct types of variation [13]. For a process characteristic $c \in C$:

- Trend ($T_c$): systematic, long-term changes in the characteristic over time.
- Seasonality ($S_c$): systematic, recurring short-term patterns periodically.
- Residual ($R_c$): random fluctuations not explained by trend or seasonality.

Since time series decomposition is a widely studied task, a range of existing methods can be used. Currently, the approach considers three decomposition methods: classical additive, classical multiplicative, and Seasonal-Trend decomposition using Loess (STL), where Loess stands for locally estimated scatterplot smoothing. In the additive case, the value $y_{c,t}$ of a characteristic $c \in C$ in time window $w_t$ is expressed as the sum of its components, $y_{c,t} = T_{c,t} + S_{c,t} + R_{c,t}$, whereas in the multiplicative case it is expressed as their product, $y_{c,t} = T_{c,t} \times S_{c,t} \times R_{c,t}$ [14, Ch. 3]. While the classical methods estimate the trend and seasonal components using moving averages, STL performs locally weighted regression to capture trend and seasonal patterns [15] In all cases, a seasonal period must be specified to capture recurring patterns in the time series.

Figure 3 shows the multiplicative decomposition of the new case count time series, revealing yearly seasonality, a stable trend, and residual fluctuations that are not explained by other components.

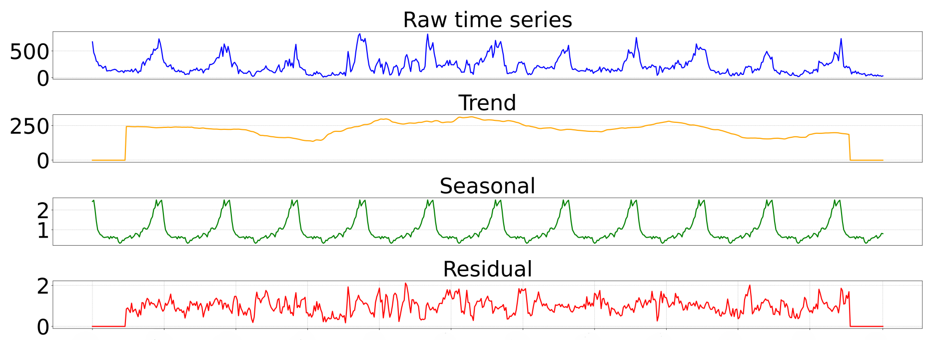



Fig. 3: Decomposed time series for new case count in the *Road Traffic Fine Management* process.

**Decomposition analysis.** Beyond visualizing the decomposed time series, it is crucial to determine to what extent the trends and seasonal patterns explain the observed temporal dynamics, and how much of the variations correspond to random fluctuations. To this end, we use the following analyses:

*Component variance fraction.* To understand how much each component contributes to the overall dynamics of a characteristic, the fraction of total variation associated with the trend, seasonal, and residual components can be quantified. We refer to this measure as the component variance fraction. We use the component variance fraction instead of trend- or seasonality-strength measures because our aim is not to assess whether trend or seasonality is strong relative to random fluctuations. Rather, we aim to determine to what extent each component accounts for the observed dynamics of a process characteristic.

For this, the relative contribution of a given component to the summed variance of all components can be computed. For example, for the trend this yields:

$$\mathrm{vf}(T_c) = \frac{\mathrm{Var}(T_c)}{\mathrm{Var}(T_c) + \mathrm{Var}(S_c) + \mathrm{Var}(R_c)} \tag{1}$$

The formulas for the seasonal and residual components follow the same structure, replacing $T_c$ in the numerator with $S_c$ and $R_c$, respectively. Note that the denominator does not necessarily equal $\mathrm{Var}(y_c)$, as covariances between the components may be non-zero.

Taking the new case counts in Figure 3 as an example, suppose that the trend, seasonality, and residual components explain 60%, 30%, and 10% of the variance, respectively. This would indicate that both long-term shifts and recurring cycles contribute significantly to the dynamics of new cases, whereas irregular fluctuations would play a comparatively minor role.

*Ljung-Box test.* To assess whether any structured patterns remain in the residuals after decomposition, the Ljung-Box test can be applied to each residual time series [14, ch.5]. This test evaluates whether the residuals can be considered white noise at a predefined significance level $\alpha$, i.e., random fluctuations with no autocorrelation. It takes a predefined maximum lag $h$ for which residual autocorrelation is jointly assessed. In this test, the null hypothesis is that the residual autocorrelations up to lag $h$ are jointly zero. It thus complements the component variance fractions by indicating whether the residuals still contain autocorrelated structure after decomposition.

**Decomposition method selection.** As different time series may be better represented by different decomposition methods, the approach selects a decomposition method for each time series $y_c$. To this end, the approach applies the classical additive, classical multiplicative, and STL decomposition methods and computes the variance fractions of the resulting trend, seasonal, and residual components. The method whose decomposition yields the smallest residual variance fraction ($\mathrm{vf}(R_c)$) is then selected. This criterion is aligned with our goal of explaining as much of the observed dynamics as possible through systematic components (trend and seasonality), leaving only a limited share of unexplained fluctuations in the residual component.

### 3.3 Analyze Time Series Interrelations

Next, we explore how the investigation of multiple time series can be supported beyond visualization. This responds to C2 in Section 2: the need to identify interrelated patterns across different process characteristics. After analyzing each time series individually, this step examines whether pairs of process characteristics move together over the same time windows. To this end, we applyzero-lag Pearson correlation analysis, which quantifies the strength and direction of the linear relationship between two time series $y_a, y_b \in Y_C$ within the same time window. The correlation coefficient $r_{a,b} \in [-1, 1]$ indicates whether the series move in the same direction (positive correlation), in opposite directions (negative correlation), or show no linear relationship (values close to 0). It is computed as follows [14, ch.2]:

$$r_{a,b} = \frac{\sum_t (y_{a,t} - \overline{y_a})(y_{b,t} - \overline{y_b})}{\sqrt{\sum_t (y_{a,t} - \overline{y_a})^2 \sum_t (y_{b,t} - \overline{y_b})^2}} \tag{2}$$

Here $\overline{y_a}$ and $\overline{y_b}$ denote the means of the respective time series. Note that delayed interrelations could be studied with lag-aware methods such as cross-correlation. However, this would require comparing multiple lags and carefully interpreting whether a high correlation at a given lag reflects a meaningful delay or is influenced by temporal patterns such as trend and seasonality. We therefore leave such analysis as future work.

### 3.4 Output

Overall, the approach produces a detailed profile of continuous process dynamics. For each characteristic in $C$, it provides time series, decomposed components, statistical summaries, component variance fractions, and residual diagnostics. Together with correlation measures between characteristics, these outputs help analysts interpret both individual dynamics and how they evolve in relation to each other over time.

## 4 Evaluation

This section presents evaluation experiments to systematically assess the proposed approach in two complementary experiments. In the first experiment, we conduct a backward-looking analysis to examine how the approach supports understanding systematic process dynamics and interrelations between process characteristics. We then perform a forward-looking analysis to evaluate whether separating systematic dynamics improves the accuracy of predicting future process behavior, compared to predicting the raw time series. For reproducibility, all code and supplementary results are available in our project repository.[3]

### 4.1 Experiment 1: Backward-Looking Analysis

In the first experiment, we assess the ability of our approach to make sense of continuous process dynamics. Specifically, we determine the extent to which the systematically separated dynamics explain observed variations, and to what degree changes in one process characteristic relate to changes in others.
**Data.** For this experiment, we use four real-life event logs publicly available through the 4TU Research Data repository[4], namely BPIC18, BPIC15-1, Hospital Billing, and International Declarations (IntDec). They originate from different domains and differ substantially in their characteristics. Specifically, the mean workload across the logs ranges from about 78 to over 14,500 cases per time window. The mean number of arriving cases varies from less than one to more than 40 per time window, while the mean number of active resources ranges from about two to over 60 per time window. This diversity allows us to evaluate the approach across processes with different scales and operational dynamics.
**Setup.** To conduct this experiment, we use the following setup.
*Approach implementation.* We implemented the proposed approach in Python, leveraging the *PM4Py* and *statsmodels* libraries for event log handling and time series analysis, respectively.

[3] `https://git01lab.cs.univie.ac.at/sanad95/cbf`
[4] `https://data.4tu.nl/categories/13500?categories=13503`

*Time series construction.* To conduct the evaluation across different aspects of processes, we construct time series for multiple process characteristics using Step 1 of our approach.

- *New case counts:* A case-level process characteristic indicating the number of new cases in time window $w_t$.
- *Active case counts:* A case-level process characteristic representing the number of cases that have started but have not yet completed during time window $w_t$. This characteristic quantifies the workload of the process.
- *Average throughput time:* A performance-related process characteristic capturing the mean completion time of cases in window $w_t$.
- *Active resource counts:* A resource-related process characteristic representing the number of distinct resources involved in the process during window $w_t$.

*Parameter settings.* For the evaluation, following Step 1, we use daily windowing with a seasonal period of 7 days, and remove warm-up and cool-down phases using a percentile-based threshold $\delta$, computed from the active case counts as the value below which 25% of observations fall. Moreover, in Step 2 of our approach, we use $\alpha = 0.05$ and $h = 10$ for the Ljung–Box test.

**Results.** We first examine the overall variability of individual process characteristics to understand how strongly they fluctuate over time. We then assess how much of their observed variation is explained by the systematically separated dynamics identified by our approach. Finally, we examine interrelations between characteristics to determine whether our approach reveals systematic relationships between their changes.

*High-level variability of process characteristics.* Table 1 summarizes the CVs (coefficients of variation) for all characteristics across the considered event logs. After examining the statistics of all event logs, we observe substantial differences in the degree of variability across logs and process characteristics. Overall, the CVs range from 0.07 to 3.45, indicating that some processes exhibit relatively stable behavior while others exhibit larger dynamics. Among all event logs, the Hospital Billing log consistently shows the most stable behavior across all characteristics, with low CVs (CV = 0.07 for active cases, 0.14 for case arrivals, 0.26 for average throughput time, and 0.12 for active resources). In contrast, several logs exhibit substantially stronger fluctuations in specific characteristics. For example, BPIC18 shows the strongest variability in case arrivals and active resources (CV = 3.45 and CV = 0.94, respectively). Similarly, International Declarations and BPIC15-1 exhibit the highest variability in active cases and average throughput time (CV = 0.63 and CV = 2.41, respectively). As shown in Figure 4, the raw (non-decomposed) time series of active resource counts shows a stable pattern in the Hospital Billing log, while it fluctuates strongly in BPIC18.

*Explained variability through decomposition.* To gain a nuanced understanding of these fluctuations, we analyze the variance fractions of the decomposed components, as reported in Table 1. The table also includes the results of the white noise test of the residuals and the automatically selected decomposition methods.

The results reveal a clear pattern: almost all characteristics with low variability (CV) are largely explained by the trend component. We can observe this

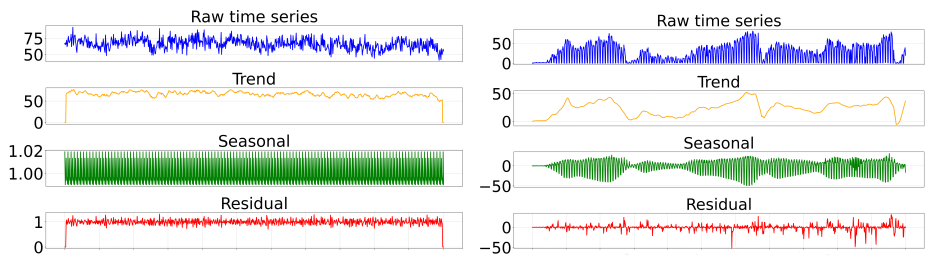
| Log | Characteristic | CV | $vf(T_c)$ | $vf(S_c)$ | $vf(R_c)$ | WN | DM |
|---|---|---|---|---|---|---|---|
| BPIC18 | Active case count | 0.35 | 1.00 | 0.00 | 0.00 | No | Multi. |
| | New case count | 3.45 | 0.76 | 0.24 | 0.00 | No | STL |
| | Avg throughput time | 1.18 | 0.36 | 0.46 | 0.18 | No | STL |
| | Active resource count | 0.94 | 0.31 | 0.59 | 0.10 | No | STL |
| BPIC15-1 | Active case count | 0.14 | 1.00 | 0.00 | 0.00 | No | Multi. |
| | New case count | 1.52 | 0.18 | 0.13 | 0.69 | No | Additive |
| | Avg throughput time | 2.41 | 0.04 | 0.25 | 0.71 | Yes | STL |
| | Active resource count | 0.83 | 0.10 | 0.67 | 0.23 | No | STL |
| Hospital Billing | Active case count | 0.07 | 1.00 | 0.00 | 0.00 | No | Multi. |
| | New case count | 0.14 | 1.00 | 0.00 | 0.00 | No | Multi. |
| | Avg throughput time | 0.26 | 1.00 | 0.00 | 0.00 | No | Multi. |
| | Active resource count | 0.12 | 1.00 | 0.00 | 0.00 | No | Multi. |
| IntDec | Active case count | 0.63 | 1.00 | 0.00 | 0.00 | No | Multi. |
| | New case count | 1.13 | 0.46 | 0.34 | 0.20 | No | STL |
| | Avg throughput time | 1.42 | 0.10 | 0.42 | 0.48 | Yes | STL |
| | Active resource count | 0.30 | 0.17 | 0.48 | 0.35 | Yes | STL |

Table 1: Variance fractions of trend ($T_c$), seasonal ($S_c$), and residual ($R_c$) components for the considered process characteristics. WN indicates if the residuals are white noise, and DM the selected decomposition method (STL, Additive, Multi. for multiplicative).

pattern in Hospital Billing log, where all characteristics exhibit very small CVs. For these series, the approach selects the classical multiplicative decomposition and attributes nearly all variance to the trend component (rounded to 1.00 for all four characteristics), while seasonal and residual contributions remain negligible. As illustrated in Figure 4a, although the approach separates seasonal patterns and random fluctuations of the active resource counts in this event log, their magnitudes remain much smaller than the trend. A similar pattern can also be observed in the International Declarations log, where active case counts are almost entirely explained by the trend component (1.00), suggesting that fluctuations are primarily driven by long-term developments. This can encourage analysts to focus on long-term trends rather than short-term variations.

In contrast, BPIC18 exhibits more diverse dynamics across most characteristics, where in active case counts the trend dominates, while in others multiple components contribute substantially to the observed variability. For instance, new case counts are explained by a combination of trend (0.76) and seasonal (0.24) components. Similarly, as shown in Figure 4b, for active resource counts the seasonal component (0.59) dominates the trend (0.31), with both seasonal and residual components visibly contributing to the observed fluctuations. This suggests that fluctuations are largely driven by systematic seasonal effects, which may encourage analysts to explicitly consider such patterns in resource planning.

Finally, a different pattern emerges when the residual component dominates the variability. Specifically, we observe this in BPIC15-1 for average throughput time, where the residual explains most of the variance (0.71), followed by smaller contributions from the seasonal (0.25) and trend (0.04) components. In this case, the residual behaves as white noise, indicating that most systematic dynamics are effectively captured by the decomposition, suggesting that the remaining fluctuations are largely irregular and therefore difficult for analysts to anticipate.




(a) Hospital Billing (b) BPIC18

Fig. 4: Extracted components of active resource counts in two event logs.

*Interrelations between process characteristics.* To complement the analysis of individual dynamics, we next examine how process characteristics relate to one another through correlation analysis, as summarized in Table 2.

Across all event logs except for the International Declarations log, the correlation between new case counts and active case counts is relatively weak, despite the intuitive expectation that increases in arrivals should immediately lead to more active cases. For example, in the BPIC18 log, where new case counts are highly dynamic, this correlation is slightly negative (-0.17), suggesting that case arrivals and workload do not necessarily increase or decrease simultaneously. This likely reflects temporal delays between incoming cases and their accumulation. In contrast, in the International Declarations log, where active case counts fluctuate strongly, the correlation becomes much stronger (0.57), indicating a more immediate accumulation of active cases. This can enable analysts to identify whether increases in case arrivals are instantly reflected in the workload or whether delays occur before cases accumulate in the system, helping them better understand and address such delays.

At the same time, active resource counts often show stronger associations with incoming cases than with the current number of active cases. In the Hospital Billing log, for instance, the correlation between new case counts and active resource counts reaches 0.67, while the correlation with active case counts remains much weaker (0.18). This suggests that resource availability fluctuates more in line with case arrivals than with the current workload. Specifically, such patterns may help analysts understand whether resource allocation is primarily driven by incoming demand rather than by the number of ongoing cases.

| Log | $r_{NC,AC}$ | $r_{NC,AR}$ | $r_{AC,AR}$ | $r_{TT,AC}$ | $r_{TT,NC}$ | $r_{TT,AR}$ |
|---|---|---|---|---|---|---|
| BPI Challenge 2018 | -0.17 | -0.15 | 0.17 | 0.03 | 0.04 | 0.50 |
| BPIC15-1 | 0.15 | 0.51 | 0.23 | -0.02 | 0.07 | 0.27 |
| Hospital Billing | 0.12 | 0.67 | 0.18 | 0.31 | -0.35 | -0.18 |
| International Declarations | 0.57 | 0.41 | 0.34 | 0.18 | 0.16 | 0.29 |

Table 2: Pairwise correlations between process characteristics across the analyzed event logs. NC = new case count, AC = active case count, AR = active resource count, and TT = average throughput time.

Moreover, average throughput time is often more strongly associated with active resource counts than with new or active case counts. In the BPIC15-1 log, where throughput time is highly dynamic, the correlation with active resource counts is 0.27, while correlations with new case counts (0.07) and active case counts (-0.02) remain negligible. This allows analysts to identify whether fluctuations in average throughput time co-occur with changes in resource availability rather than with variations in case arrivals or workload, providing insights into operational factors associated with performance dynamics.

Overall, the backward-looking analysis demonstrates that the proposed approach can support analysts in obtaining insights into both the individual dynamics of process characteristics and their interrelations. Such insights can in turn guide further investigations into potential process improvements.

### 4.2 Experiment 2: Forward-Looking Analysis

This experiment further evaluates the usefulness of our approach by assessing whether systematically separating process dynamics through decomposition improves predictions of future process behavior. Specifically, we compare the accuracy of predictive models trained on decomposed time series to those trained on the original series, determining which models provide better insights into the future values of characteristics such as case arrivals, workload, resource availability, and process performance.

**Setup.** We use the same data and setup as for Experiment 1, unless stated otherwise below. We further consider the following aspects in Experiment 2:

*Prediction models.* We consider two widely-used time series models: AutoRegressive Integrated Moving Average (ARIMA) and Seasonal Autoregressive Integrated Moving Average (SARIMA). ARIMA captures non-seasonal patterns such as trends, whereas SARIMA additionally models seasonality [13, Chs. 3,7,8]. In our experiment, both models are configured with one seasonal cycle, while all other parameters follow their default settings.

*Train-test split.* After removing the warm-up and cool-down phases in Step 1 of our approach, we split each time series chronologically into training and test segments. We train the models on the first 80% of observations and evaluate them on the remaining 20%, i.e., the test segment.

*Baseline prediction models.* We establish baselines by applying ARIMA and SARIMA directly to the original time series using the train-test split described above. This allows us to assess how well commonly used time series models capture process dynamics without separating systematic and irregular fluctuations.

*Decomposition-based prediction models.* For decomposition-based prediction models, we decompose the training segment into trend, seasonal, and residual components. We employ STL decomposition, as the other decomposition methods produce undefined trend values at the boundaries of the series, which prevents generating predictions over the full time horizon.

After decomposition, we represent each time series by combinations of its three component series. We can model these components either separately or in a grouped form, which leads to multiple possible configurations. In this study, we focus on three configurations that reflect different assumptions about the role of the residual component: we either model it independently, combine it with the trend, or exclude it as an irregular fluctuation. For each configuration, we train one model per component group on the training segment and predict the test segment. We then combine these predictions to reconstruct the original series.

– *Independent residual* $(T_c, S_c, R_c)$*:* We predict the components independently: trend and residual using ARIMA and the seasonal component using SARIMA.
– *Residual merged with trend* $(T_c{+}R_c, S_c)$*:* We aggregate the residual with the trend and predict them jointly using ARIMA, while the seasonal component is predicted separately using SARIMA. This configuration aims to reduce error accumulation from predicting the residual independently.
– *Residual excluded* $(T_c, S_c)$*:* We predict only the systematic components: the trend using an ARIMA model and the seasonal component using a SARIMA model, while the residual is not modeled.

*Evaluation metric.* To compare the prediction accuracy of the baselines and decomposition-based predictions, we use Root Mean Squared Error (RMSE), which captures deviations at individual time points and emphasizes larger errors [13, Ch. 5]. This metric is defined as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_{c,t} - \hat{y}_{c,t})^2} \tag{3}$$

where $y_{c,t}$ denotes the observed value of time series $c$ in time window $w_t$, $\hat{y}_{c,t}$ the corresponding prediction, and $n$ the number of test observations.

**Results.** Table 3 reports the RMSE obtained by predicting the process characteristics across all event logs using the raw baselines as well as the decomposition-based predictions. Out of the 16 time series, decomposition outperforms both baselines in six and improves over at least one baseline in nine. This indicates that decomposition can improve prediction accuracy in some settings, but does not consistently outperform the raw-series baselines. Overall, the results show that when decomposition improves over at least one baseline, the reduction in RMSE is often substantial: for the $(T_c{+}R_c, S_c)$ configuration, the average improvement is 24.03%. In contrast, when it does not outperform the baselines, the

| Log | Characteristic | Baselines | | Decomposition-based | | |
|---|---|---|---|---|---|---|
| | | ARIMA | SARIMA | (T,S,R) | (T+R,S) | (T,S) |
| BPIC18 | new case counts | 1.07 | 48.12 | 0.47 | **0.44** | 0.47 |
| | active case counts | **4573.30** | 4583.98 | 4596.81 | 4643.27 | 4593.68 |
| | avg throughput time | 7948.89 | 7707.44 | 5866.70 | **5789.62** | 5944.21 |
| | active resource counts | 34.95 | **19.95** | 27.73 | 20.31 | 27.75 |
| BPIC15-1 | new case counts | 0.84 | **0.83** | 1.01 | 0.85 | 0.91 |
| | active case counts | 10.31 | 10.36 | 7.97 | 9.66 | **7.95** |
| | avg throughput time | **2466.50** | 2471.61 | 2529.38 | 2470.77 | 2599.60 |
| | active resource counts | 1.54 | 1.52 | 1.66 | **1.21** | 1.68 |
| Hospital Billing | new case counts | **12.66** | 12.67 | 24.48 | 13.36 | 24.55 |
| | active case counts | 1884.04 | 1903.66 | 1765.75 | 1917.35 | **1758.21** |
| | avg throughput time | **631.71** | 632.19 | 647.93 | 645.30 | 649.90 |
| | active resource counts | 8.06 | **8.05** | 9.26 | 8.92 | 9.34 |
| IntDec | new case counts | **10.13** | 10.19 | 15.91 | 10.83 | 15.82 |
| | active case counts | **749.46** | 756.96 | 795.31 | 764.50 | 796.25 |
| | avg throughput time | 1411.54 | 1400.61 | 1442.48 | **1369.33** | 1473.47 |
| | active resource counts | 0.66 | **0.65** | 0.67 | 0.66 | 0.71 |

Table 3: RMSE values for baselines and decomposition-based predictions across all event logs and process characteristics.

accuracy decrease remains limited, with an average degradation of only 3.48% for this configuration. This can be illustrated by individual examples. For instance, in the BPIC18 log, the $(T_c{+}R_c, S_c)$ configuration reduces the RMSE for the average throughput time from 7707.44 (SARIMA) to 5789.62. Conversely, for the active case count in the International Declarations log, its RMSE (764.50) is only slightly higher than that of ARIMA (749.46).

To understand when decomposition improves prediction accuracy, we analyze the variance fractions of the decomposed components. The corresponding values are available in the accompanying repository. Among time series with substantial seasonal fluctuations ($vf(S_c) > 0.30$), decomposition outperforms at least one baseline in six out of seven time series. This indicates that decomposition tends to improve prediction when seasonal patterns contribute substantially to the dynamics. For instance, Figure 5 shows the average throughput time in the BPIC18 log, where strong seasonal dynamics are present ($vf(S_c)$=0.41). Here, neither ARIMA (RMSE = 7948.89) nor SARIMA (RMSE = 7707.44) captures the recurring fluctuations effectively. In contrast, the $(T_c{+}R_c, S_c)$ configuration (RMSE= 5789.62), shown as a representative decomposition-based prediction, captures the seasonal structure more accurately and therefore produces substantially better predictions.

The grouping configurations also exhibit different behaviors. Overall, the $(T_c{+}R_c, S_c)$ configuration performs best among the evaluated decompositions in four time series and outperforming at least one baseline in eight. This suggests

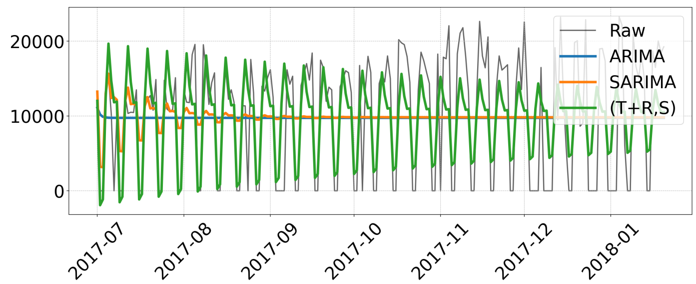



Fig. 5: Predictions for average throughput time in BPIC18 log for baselines and (T+R,S) decomposition-based prediction in the test segment.

that combining the residual with the trend while modeling the seasonal component separately often improves prediction accuracy. In contrast, the $(T_c, S_c, R_c)$ configuration does not achieve the best overall result for any time series, although it still improves prediction accuracy over at least one baseline in five. Similarly, the $(T_c, S_c)$ configuration exhibits the best performance in two time series and also outperforms at least one baseline in five. A closer inspection shows that both $(T_c, S_c, R_c)$ and $(T_c, S_c)$ mainly improve prediction when the residual variance fraction is very small and the trend variance fraction is large. For example, for the active case count in the Hospital Billing log (trend 0.99, residual 0.01), $(T_c, S_c, R_c)$ yields an RMSE of 1765.75, while $(T_c, S_c)$ achieves a slightly lower RMSE of 1758.21, indicating that explicitly modeling the systematic components helps capture the main structure of the series.

Overall, decomposition is most beneficial for predicting future behavior of a process when seasonality explains a large share of the dynamics. The $(T_c{+}R_c, S_c)$ grouping often performs best, highlighting the importance of appropriate component groupings. Moreover, decomposition rarely harms prediction accuracy and can yield substantial improvements when strong seasonal dynamics are present.

## 5 Related Work

Our work relates to research on the analysis of process dynamics and the use of time series in process mining.

**Analysis of process dynamics.** As the dynamic nature of business processes has gained increasing attention, several approaches have emerged to support the analysis and visualization of behavioral changes over time. Process Evolution Analysis (PEA) [5] was among those that track how clusters of process executions emerge and shift across the timeline. Building on this, Visual Drift Detection (VDD) [9] facilitates the visual identification of behavioral drifts by highlighting changes in constraints over time. More recently, DyLoPro [10] was introduced as a visual analytics tool that allows for the exploration of changes across multiple process aspects. While these approaches offer support for analyzing process dynamics, they primarily rely on visual inspection and lack systematic, data-driven analysis of continuous fluctuations. However, current data-driven analyses remain largely descriptive and focus on aggregated behavior, offering limited support for understanding fine-grained, temporal dynamics.

To address these limitations, our work provides a systematic, time-series-based perspective for capturing continuous behavioral change in business processes, moving beyond visual exploration toward deeper analytical insight.

**The use of time series in process mining.** Time series analysis has become a valuable technique in process mining, as it enables the investigation of temporal patterns across various process characteristics. Many studies have transformed event logs into time series that track the evolution of metrics such as activity frequencies, case volumes, process variants, and event counts [2, 10, 16]. Initially, early efforts explored the use of such temporal representations to monitor behavioral changes, treating event logs as time-ordered sequences of traces to detect significant shifts over time [16].

Beyond descriptive monitoring, time series have served as the basis for more advanced forms of analysis, including steady-state detection [2] and visual exploration of evolving process aspects [10]. Furthermore, they have enabled predictive tasks—for instance, by forecasting structural changes in control-flow by modeling temporal patterns of DFG relations [17]. In addition, multivariate time series analysis has supported the examination of interdependencies across perspectives such as workload, performance, and resource usage, shedding light on how process dimensions influence one another during short-term disruptions [11].

While prior work applies time series techniques in process mining to raw time series, our approach is the first to leverage time series decomposition to detect and analyze structured changes in process behavior.

## 6 Conclusion

This paper explores how time series analysis can provide deeper insights into the continuous dynamics of processes. Compared to prior work, our approach focuses on separating systematic from random fluctuations and identifying interrelated dynamics. Our evaluation shows that this can reveal the sources of temporal fluctuations and dependencies between process characteristics. In particular, a forward-looking analysis indicates that separating systematic dynamics can expose informative predictive signals: decomposition-based predictions are often comparable to or better than predictions on the raw time series, especially when seasonality explains a substantial share of the observed variance.

These findings should be interpreted in light of several threats to validity. First, the evaluation is based on four event logs, and the findings may therefore not generalize to all types of processes and process dynamics. Second, the results depend on several design choices, including the selected process characteristics, the window size, and other parameter settings for both backward-looking and forward-looking analyses. Third, in the forward-looking analysis, the choice of prediction models itself may influence the comparison. Other prediction approaches, such as SARIMAX, VAR, VARIMA, etc., may yield different results

Finally, although the approach can provide meaningful insights into dynamic patterns, it has not yet been tested by users. We aim to address this in future work to evaluate its practical usefulness. Moreover, future work could extend the predictive analysis by incorporating uncertainty estimates, for example by using prediction intervals instead of point predictions to capture uncertainty in future process dynamics. Overall, our results demonstrate that integrating time series decomposition and correlation analysis into process mining provides a structured perspective for understanding and anticipating continuous process dynamics.

## References

1. W. Van Der Aalst, *Process Mining: Data science in action*. Springer, 2016.
2. A. Kraus, K. A. Elyasi, and H. van der Aa, "On the use of steady-state detection for process mining: Achieving more accurate insights," in *CAiSE*. Springer, 2025, pp. 204–220.
3. T. Grisold, C. Janiesch, M. Röglinger, and M. T. Wynn, "Managing dynamics in and around business processes," *BISE*, vol. 66, no. 5, pp. 533–540, 2024.
4. R. J. C. Bose, W. M. Van Der Aalst, I. Žliobaitė, and M. Pechenizkiy, "Dealing with concept drifts in process mining," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 1, pp. 154–171, 2013.
5. A. Yeshchenko, D. Bayomie, S. Gross, and J. Mendling, "Visualizing business process evolution," in *BPMDS*. Springer, 2020, pp. 185–192.
6. S. Franzoi, S. Hartl, T. Grisold, H. van der Aa, J. Mendling, and J. vom Brocke, "Explaining process dynamics: a process mining context taxonomy for sensemaking," *Process Science*, vol. 2, no. 1, p. 2, 2025.
7. B. Pentland, E. Vaast, and J. R. Wolf, "Theorizing process dynamics with directed graphs: A diachronic analysis of digital trace data," *MIS Quart*, vol. 45, no. 2, 2021.
8. C. Browne, "A change dynamic model for the online detection of gradual change," *arXiv preprint arXiv:2205.01054*, 2022.
9. A. Yeshchenko, C. Di Ciccio, J. Mendling, and A. Polyvyanyy, "Visual drift detection for event sequence data of business processes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 8, pp. 3050–3068, 2021.
10. B. Wuyts, H. Weytjens, S. vanden Broucke, and J. De Weerdt, "Dylopro: Profiling the dynamics of event logs," in *BPM*. Springer, 2023, pp. 146–162.
11. A. Kraus, J.-R. Rehse, and H. van der Aa, "Data-driven assessment of business process resilience," *Process Science*, vol. 1, no. 1, p. 4, 2024.
12. W. M. Van Der Aalst and J. Carmona, *Process mining handbook*. Springer, 2022.
13. R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*, 2nd ed. OTexts, 2018.
14. ——, *Forecasting: Principles and Practice*, 3rd ed. OTexts, 2021.
15. R. B. Cleveland, W. S. Cleveland, J. E. McRae, and I. Terpenning, "Stl: A seasonal-trend decomposition procedure based on loess," *JOS*, vol. 6, pp. 3–73, 1990.
16. R. J. C. Bose, W. M. van der Aalst, I. Žliobaitė, and M. Pechenizkiy, "Handling concept drift in process mining," in *CAiSE*. Springer, 2011, pp. 391–405.
17. J. De Smedt, A. Yeshchenko, A. Polyvyanyy, J. De Weerdt, and J. Mendling, "Process model forecasting and change exploration using time series analysis of event sequence data," *Data & Knowledge Engineering*, vol. 145, p. 102145, 2023.