Expert Systems With Applications 253 (2024) 124181

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Review

# Machine learning in business process management: A systematic literature review

Sven Weinzierl [a,*], Sandra Zilker [b], Sebastian Dunzer [a], Martin Matzner [a]

[a] Institute of Information Systems, Friedrich-Alexander-Universität Nürnberg-Erlangen, Germany
[b] Professorship for Business Analytics, Technische Hochschule Nürnberg Georg Simon Ohm, Germany

ARTICLE INFO

*Keywords:*
Business process management
BPM lifecycle
Machine learning
Deep learning
Literature review

ABSTRACT

Machine learning (ML) provides algorithms to create computer programs based on data without explicitly programming them. In business process management (BPM), ML applications are used to analyse and improve processes efficiently. Three frequent examples of using ML are providing decision support through predictions, discovering accurate process models, and improving resource allocation. This paper organises the body of knowledge on ML in BPM. We extract BPM tasks from different literature streams, summarise them under the phases of a process's lifecycle, explain how ML helps perform these tasks and identify technical commonalities in ML implementations across tasks. This study is the first exhaustive review of how ML has been used in BPM. We hope that it can open the door for a new era of cumulative research by helping researchers to identify relevant preliminary work and then combine and further develop existing approaches in a focused fashion. Our paper helps managers and consultants to find ML applications that are relevant in the current project phase of a BPM initiative, like redesigning a business process. We also offer – as a synthesis of our review – a research agenda that spreads ten avenues for future research, including applying novel ML concepts like federated learning, addressing less regarded BPM lifecycle phases like process identification, and delivering ML applications with a focus on end-users.

## 1. Introduction

Organisations have made great strides in digitising their business processes (Beverungen et al., 2021), so information systems now produce large amounts of process data (van der Aalst, 2016). BPM research offers approaches to create value from such data (van der Aalst, 2016), including process mining (e.g. van der Aalst et al., 2011), business activity monitoring (e.g. McCoy, 2002), predictive business process monitoring (e.g. Grigori et al., 2004), and anomaly detection (e.g. Bezerra, Wainer, & van der Aalst, 2009). ML is popular as an analytical capability at the core of BPM approaches. Because of the increased availability of event data, the emergence of off-the-shelf ML libraries, and advances in hardware, ML approaches are increasingly used to solve BPM tasks. According to Mitchell (1997, p. xv), "[t]he field of machine learning is concerned with the question of how to construct computer programs that automatically improve with experience". ML provides algorithms that identify and learn patterns from data (e.g. examples or observations) and capture those in mathematical models (e.g. functions) through automated improvement procedures (Bishop, 2006).

Before ML can be used in applications tailored to specific BPM tasks (e.g. anomaly detection), underlying ML models are developed by completing the following four steps: Data input, feature engineering, model building, and model assessment (Goodfellow, Bengio, & Courville, 2016; Janiesch, Zschech, & Heinrich, 2021; Liu & Motoda, 1998). Once these models are developed, ML applications can create value from process data, such as cost reduction or risk mitigation (Márquez-Chamorro, Eduardo and Resinas, & and Ruiz-Cortés, 2017). In short, ML supports organisations in improving their business processes (Mendling, Decker, Hull, Reijers, & Weber, 2018).

While ML in BPM bears considerable potential for improving organisational operations, we believe that an overview of ML applications in BPM can help scholars propose contributions in this research area and practitioners find existing ML applications that address a problem they are facing. This systematic literature review contributes to research and practice in the following five ways:

1. We describe BPM lifecycle phases and define BPM tasks in view of ML applications.

* Corresponding author.
*E-mail addresses:* sven.weinzierl@fau.de (S. Weinzierl), sandra.zilker@th-nuernberg.de (S. Zilker), sebastian.dunzer@fau.de (S. Dunzer), martin.matzner@fau.de (M. Matzner).

https://doi.org/10.1016/j.eswa.2024.124181
Received 29 June 2023; Received in revised form 20 April 2024; Accepted 7 May 2024
Available online 22 May 2024

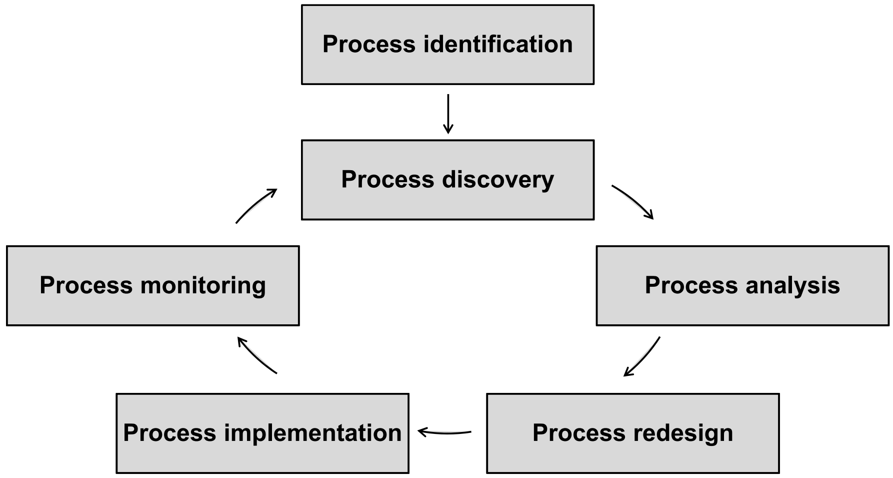


**Fig. 1.** BPM lifecycle based on Dumas et al. (2018).

2. We present an overview of ML applications in the field of BPM.
3. We provide a technical summary of ML applications of each BPM lifecycle phase.
4. We derived findings from our literature review.
5. We set up a future research agenda for advancing research on ML applications in BPM.

The remainder of this paper is structured as follows. After we present the core concepts of BPM and ML in Section 2, we provide an overview of related literature reviews in Section 3. Then, in Section 4, we describe our applied research method, after which we provide an overview of ML applications in BPM, and describe BPM lifecycle phases and define BPM tasks in the view of ML applications (Section 5). Subsequently, in Section 6, we synthesise the results from the previous section and present findings, which we derived from the synthesised results. In Section 7, we present directions for future research, implications for research and practice, and limitations of our literature review.

## 2. Background

This paper focuses on applications based on ML models in the BPM domain. Therefore, in this section, we provide overviews of BPM and its lifecycle, as well as the process of ML model development in general.

### *2.1. BPM and its lifecycle*

BPM is a management discipline aiming to increase a business's competitive advantage by facilitating continuous improvement in organisational operations (Trkman, 2010). To achieve this, BPM proposes management practices that cultivate an end-to-end view of customer-oriented processes across functional boundaries (Trkman, 2010). BPM tasks are classified along lifecycle models to analyse, redesign, implement, and monitor business processes continuously (De Morais & Kazan, 2014). For this paper, we align with Recker and Mendling (2016) in using the BPM lifecycle of Dumas, La Rosa, Mendling, and Reijers (2018) to classify existing ML applications in BPM tasks. We also consider definitions from other BPM lifecycle models in describing each of the phases (e.g. De Morais & Kazan, 2014; Houy, Fettke, & Loos, 2010; zur Muehlen & Ho, 2005). BPM lifecycle model of Dumas et al. (2018) has six phases through which (except *process identification*, the entry point to BPM) business processes pass multiple times to facilitate constant improvement (see Fig. 1).

***Process identification.*** The process identification phase identifies, delimits, and interrelates processes and their stakeholders. Taking a cross-functional perspective allows processes' architectures to be set up (Dumas et al., 2018) that aid in identifying the processes that should be improved along the lifecycle (Houy et al., 2010; Trkman, 2010). An activity at this stage includes analyses of process environment and organisation (zur Muehlen & Ho, 2005).

***Process discovery.*** The process discovery phase documents the current state of processes in the form of as-is process models.[1] To enhance documentation, organisations may, for instance, specify a process in detail, model it in formal process-modelling languages, and conduct process walkthroughs (Houy et al., 2010; zur Muehlen & Ho, 2005; Zairi, 1997).

***Process analysis.*** In the process analysis phase, the resulting process documentations and as-is models are assessed to identify the issues related to a process (Dumas et al., 2018). Examples of activities in this phase are simulating a process, calculating cost and cycle-time, applying process mining, and defining target metrics (De Morais & Kazan, 2014; zur Muehlen & Ho, 2005; Zairi, 1997).

***Process redesign.*** Based on the results of the process analysis phase, the process redesign phase elaborates changes to processes that will improve them (Dumas et al., 2018), resulting in redesigned to-be process models that include information for process operation (De Morais & Kazan, 2014; Houy et al., 2010). Process analysis tools can help to determine process changes that should be implemented (Dumas et al., 2018).

***Process implementation.*** Redesigned processes must be embedded in organisations' information systems (zur Muehlen & Ho, 2005) by using organisational change mechanisms to facilitate work according to redesigned processes and adjusting the IT systems required for the redesigned processes to be executed. In short, the process implementation phase yields executable process models (Dumas et al., 2018).

***Process monitoring.*** During the execution of a redesigned process, conformance and performance are controlled using data about process executions (Houy et al., 2010). Corrective actions can be taken if errors occur or a process faces performance issues. However, if new issues arise, the lifecycle must be repeated (Dumas et al., 2018).

### *2.2. ML model development*

The process of developing an ML model consists of four phases, as depicted in Fig. 2, enriched with relevant concepts (Goodfellow et al., 2016; Janiesch et al., 2021; Liu & Motoda, 1998). In what follows, these four phases are described.

[1] Other lifecycle models refer to this phase as process design (De Morais & Kazan, 2014).

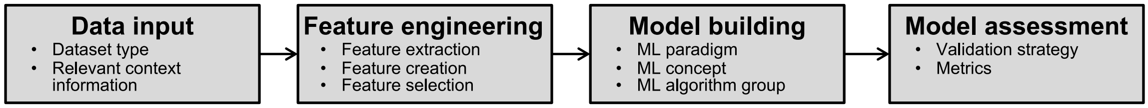


**Fig. 2.** ML model development phases based on Janiesch et al. (2021), Goodfellow et al. (2016), and Liu and Motoda (1998).

***Data input.*** Data of various types can be inputs to ML models. For example, a dataset can consist of tabular, event log, or text data. The perspectives to which the event characteristics refer, such as time, resources, or data flow, can also be included for event logs (de Leoni, van der Aalst, & Dees, 2016).

***Feature engineering.*** Feature engineering, which consists of feature extraction, creation, and selection, is performed to transform and simplify data, thus facilitating model building (Liu & Motoda, 1998). Generally, a feature is an independent variable in a dataset (Hastie, Tibshirani, & Friedman, 2009). Feature extraction retrieves features from the data input and encodes them (e.g. categorical features as one-hot encoded features), (manual) feature creation adds additional features (e.g. temporal features created based on the control-flow and timestamp information of an event log), and feature selection removes features that, for example, have little effect on the learning target.

***Model building.*** ML algorithms are applied to build mathematical models based on data (Bishop, 2006). As the research field ML provides many ML algorithms, we differentiate them based on three dimensions (Janiesch et al., 2021): The *ML paradigm* they follow, a dimension that distinguishes among the approaches supervised, unsupervised, and reinforcement learning and those in-between (i.e. semi- and self-supervised learning); the *ML concept*, which refers to the approach (e.g. deep learning) an ML algorithm uses to address a learning problem in terms of such aspects, as the expected input data, the number of ML models, or the learning target; and the *ML algorithm group*, referring to how the algorithm builds models from data. Algorithms of the same group follow similar principles for model building, including model structure and parameter fitting. Table 1 shows an overview of the ML dimensions.

***Model assessment.*** Several validation strategies can be used to test model generalisability, and metrics can be used to measure model properties (e.g. Hastie et al., 2009). While validation strategies include split validation and cross-validation, metrics can refer to quality (e.g. accuracy for classification models (Ferri, Hernández-Orallo, & Modroiu, 2009)) and runtime (e.g. training time for model training (Wang, Wong, Ding, Guo, & Wen, 2012)).

## 3. Related literature reviews

Previous literature reviews in BPM consider ML an approach to gain insights from process data. These reviews investigate the application of certain ML types for certain BPM lifecycle phases (e.g. process monitoring) and approaches (e.g. process mining). One group of literature reviews focuses on predictive business process monitoring, where ML is an approach for building predictive models. Márquez-Chamorro, Eduardo and Resinas, and and Ruiz-Cortés (2017) provide a global overview of the domain, while Di Francescomarino, Ghidini, Maggi, and Milani (2018) investigate algorithms' tasks, input data, families, and tools. Verenich, Dumas, La Rosa, Maggi, and Teinemaa (2019), Teinemaa, Dumas, La Rosa, and Maggi (2019), and Stierle, Brunk, et al. (2021) focus on techniques for predicting remaining time, techniques for classification-based process outcome prediction, and techniques that use explainable artificial intelligence (XAI) approaches, while Neu, Lahann, and Fettke (2021) and Rama-Maneiro, Vidal, and Lama (2021) conduct a systematic literature review on deep-learning approaches for predictive business process monitoring.

Previous literature reviews also address process mining. Tiwari, Turner, and Majeed (2008) perform a literature review on process mining, specifically process discovery. Maita, Martins, Paz, Peres, and Fantinato (2015) conduct a systematic literature review on applying NNs and support vector machines (SVMs) for data-mining tasks in process mining. Taymouri, La Rosa, Dumas, and Maggi (2021) conduct a systematic literature review on analysis methods for process variants, and consider ML a family of algorithms used in these methods. Folino and Pontieri (2021) perform two systematic literature reviews, one for AI -based process-mining approaches exploiting domain knowledge; second, for process-mining approaches that address auxiliary AI tasks jointly with target process-mining tasks. Both consider ML a subset of AI. Herm, Janiesch, Reijers, and Seubert (2021) conduct a literature review on intelligent robotic process automation (RPA), in which the authors consider ML an approach for transforming symbolic RPA into intelligent RPA. Wanner, Wissuchek, and Janiesch (2020) focus on the combination of ML and complex event processing in their literature review. Finally, Ko and Comuzzi (2023) perform a systematic literature review on anomaly detection for business process event logs. Table 2 provides an overview of related (systematic) literature reviews and positions our work in this context.

Our review has a more general scope and a broader focus than the other reviews we mention. The scope of our review is general, as we do not limit the time horizon because we want to consider all BPM papers that propose ML applications, regardless of when they were published. The focus of our review is broad, as we consider ML applications for BPM that rely on any type of ML and can tackle any BPM task, regardless of the phase of the BPM lifecycle to which they belong. Conducting a literature review with this scope and focus allows us to give an overview of how ML has been applied in BPM, describe BPM lifecycle phases and define BPM tasks in the view of ML applications, provide a technical summary of ML applications of each BPM lifecycle phase, derive findings, and set up an agenda that can advance research on ML applications in BPM.

## 4. Research method

We conducted a systematic literature review using a descriptive perspective and provide guidance based on the review's results (Paré, Trudel, Jaana, & Kitsiou, 2015). We follow vom Brocke et al. (2015) and the concept-centric notion of Webster and Watson (2002). Further, in line with Beese, Haki, Aier, and Winter (2019), we carried out the review in two steps, first applying a structured *search process* to gather relevant papers before developing a coding scheme and using a *coding process* following Recker and Mendling (2016).

### 4.1. Search process

We chose a systematic, sequential procedure to find a representative set of papers for our review (Cooper, 1988; vom Brocke et al., 2015). We used the databases *Scopus*, *IEEE Xplore*, and *Web of Science* to conduct the search process, as the combination of these databases covers a wide range of academic papers in the BPM domain. We defined the keywords for our search string based on our addressed research gap and the dimensions described in Section 2.1 and Section 2.2 combining BPM-related keywords with ML-related keywords (vom Brocke et al., 2015). The search string is depicted in Fig. 3 (see Appendix A for a detailed description of the creation of the search string).

We also screened the main proceedings of topical conferences (i.e. the *International Conference on Business Process Management (BPM)* and the *International Conference on Process Mining (ICPM)*). Our search and screening, executed in July 2022, retrieved 2,270 papers. Next, we

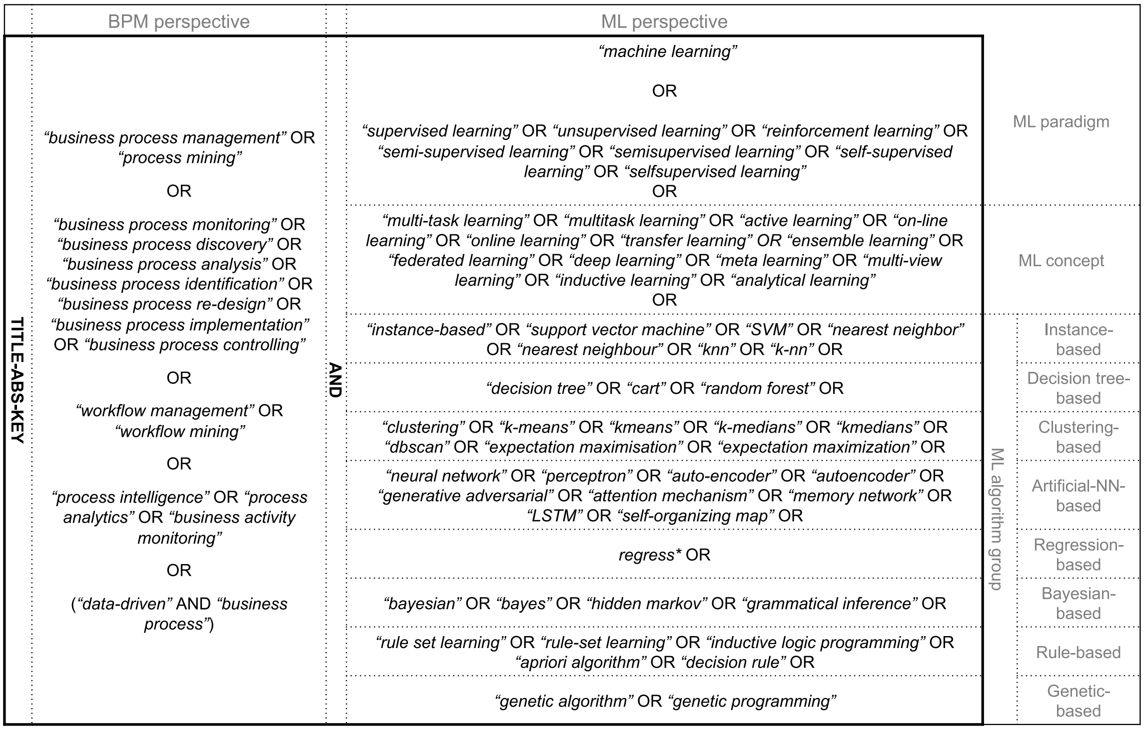


**Fig. 3.** Search string (in Scopus syntax).

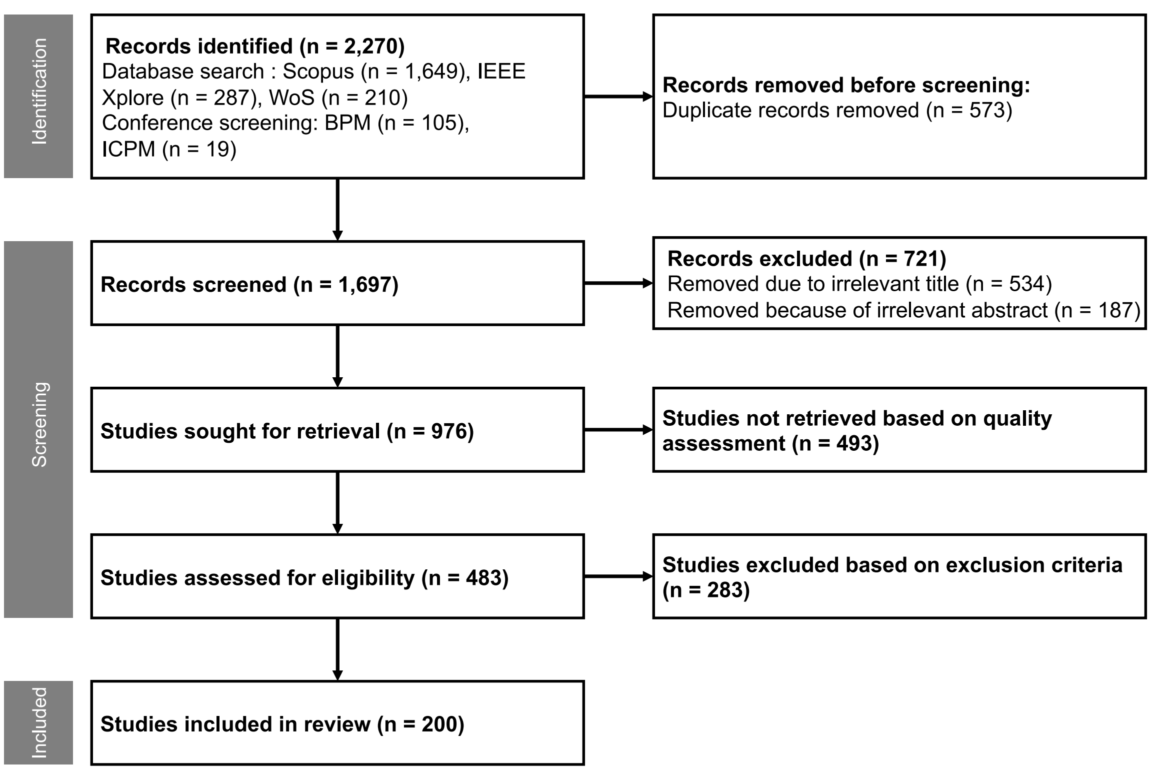


**Fig. 4.** Phases of the search process following the PRISMA framework.

**Table 1**
Overview of ML dimensions.

| | Dimension | Description |
|---|---|---|
| ML paradigm | Supervised learning | The algorithm learns from examples (Alpaydin, 2014); that is, it learns from data instances that are labelled with their correct output (Kotsiantis, Zaharakis, & Pintelas, 2006). |
| | Semi-supervised learning | The algorithm uses labelled input data and unlabelled input data for model training (Zhu, 2005), which is more effective than unsupervised learning and needs less human effort than supervised learning. |
| | Reinforcement learning | The input data for reinforcement learning algorithms is generated through the algorithm's interactions with its environment and changes over time (Alpaydin, 2014). Hence, the algorithm relies on feedback that is received during training and is rewarded for steps towards or achieving a desired objective and punished otherwise (Brynjolfsson & McAfee, 2017; Kaelbling, Littman, & Moore, 1996; Russel & Norvig, 2016). Therefore, the algorithm is like an agent that interacts with a dynamic environment via perception and action (Kaelbling et al., 1996). |
| | Self-supervised learning | Using unlabelled input data (Jaiswal, Babu, Zadeh, Banerjee, & Makedon, 2021; Jing & Tian, 2021) and training an neural network (NN) model are based on pseudo-labels created from input data to solve a "pretext task" (e.g. predicting a word based on its surrounding words in a sentence). Then the model is used to solve a "downstream task" using supervised or unsupervised learning. |
| | Unsupervised learning | Identifying and learning inherent patterns in the input data is based on structural properties (Jordan & Mitchell, 2015). In contrast to supervised learning, there are no labelled examples for model training (Russel & Norvig, 2016). |
| ML concept | Multi-task learning | The algorithm learns a model that addresses multiple related problems (e.g. Zhang & Yang, 2021). |
| | Active learning | An accurate model can be learned with a low volume of labelled training instances if an ML algorithm is allowed to choose the training data from which it learns (e.g. Settles, 2009). |
| | Online/ incremental learning | The algorithm uses the available data and updates the model before a prediction is required or after the last observation is made (e.g. Blum, 1998). |
| | Transfer learning | The algorithm first learns a model on a task. Then part or all of the model is used as the starting point for a related task (e.g. Pan & Yang, 2009). |
| | Ensemble learning | The algorithm learns two or more models on the same training set and the outputs from all models are combined (e.g. Dietterich, 2002). |
| | Deep learning | The algorithm learns multiple representations instead of a single representation to identify complex structures in data (e.g. LeCun, Bengio, & Hinton, 2015). |
| | Meta learning | The algorithm learns from the output of other ML algorithms that learn from a training set (e.g. Nichol, Achiam, & Schulman, 2018). |
| | Federated learning | Multiple data owners collectively learn and use a shared model while keeping all of the local training data private (e.g. Yang, Liu, Chen, & Tong, 2019). |
| | Multi-view learning | The algorithm learns one or more models from multiple views of training data (e.g. Xu, Tao, & Xu, 2013). |
| ML algorithm group | Instance-based | ML algorithms build models directly from the training instances (e.g. support vector machine (Cortes & Vapnik, 1995)). |
| | Decision tree-based | ML algorithms build tree-structured models, in which leaves represent class labels and branches represent conjunctions of the features that lead to these labels (e.g. classification and regression trees (Breiman, Friedman, Olshen, & Stone, 1984)). |
| | Clustering-based | ML algorithms build models to discover natural groups (i.e. clusters) in the data's feature space (e.g. k-means (Forgy, 1965)). |
| | Artificial-neural-network-based | ML algorithms build models consisting of one or more neurons that are connected via edges and structured into one or more layers from input to output (e.g. multi-layer perceptron (Rumelhart, Hinton, & Williams, 1985)). |
| | Regression-based | ML algorithms build models by separating data points through a fitted regression line (e.g. logistic regression (e.g. Hastie et al., 2009)). |
| | Bayesian-based | ML algorithms build models using Bayesian statistics (e.g. naïve Bayes (e.g. Lewis, 1998)). |
| | Rule-based | ML algorithms build models as a set of relational rules that represent knowledge (e.g. RIPPER (Cohen, 1995)).[a] |
| | Reinforcement-learning-based | ML algorithms build models on how an agent should behave in a particular environment by performing actions and observing the results (e.g. Q-learning (Watkins, 1989)). |
| | Genetic-based | ML algorithms build models by iteratively updating a population (pool of hypotheses), evaluating each population member based on a fitness measure, and selecting the best fitting members to produce the next generation (e.g. genetic algorithm (e.g. De Jong, Spears, & Gordon, 1993)). |

[a] In this study, we associate the term "rule-based" with rule-based ML (Fürnkranz, Gamberger, & Lavrač, 2012), where some form of a learning algorithm is applied to automatically identify a set of relational rules for pattern detection or prediction, and not traditional rule-based artificial intelligence (AI) systems (Hayes-Roth, 1985), where a human expert with domain knowledge manually constructs a set of rules for knowledge representation or reasoning.

proceeded in a structured manner to determine the final set of relevant papers. The search process is depicted in Fig. 4.

After removing duplicates, we scanned the title and abstract of each paper for relevance to our topic and checked the papers that remained for whether they met our quality criteria (Okoli, 2015), for example, whether they meet a length requirement of at least four pages. Lastly, to assess eligibility, we screened the full text of each remaining paper in terms of our defined exclusion criteria (vom Brocke et al., 2015). The quality and exclusion criteria can be found in Appendix A. At the end of this process, we had 200 papers.

### 4.2. Coding process

We followed Hruschka et al. (2004) adjusted to three coders by first developing an initial coding scheme, coding the entire sample second, and, third, reconciling the coders' results. In the first step, we adapted Recker and Mendling's (2016) coding scheme and Janiesch et al.'s (2021) ML-model development process to set up the BPM-related and ML-related dimensions of our coding matrix (Webster & Watson, 2002). For the ML-related dimensions, we used the categories as defined in the search string (see Appendix A). Two authors conducted two coding iterations of ten randomly sampled papers per iteration per person. After each iteration, the authors reflected on the coding scheme's sufficiency and completeness and added or adapted dimensions and concepts to better fit the review's objectives (Webster & Watson, 2002). The coders also defined each concept textually to ensure a common understanding, which finalised the coding scheme.

In the second step, three authors conducted the final coding, analysing and coding each of the 200 papers in the final set. The papers were distributed equally among the coders to deal with the large number of papers. To clarify each paper's positioning along the BPM lifecycle and its goal, the paper's BPM aim, such as predicting the next activity or predicting remaining time, was extracted as free text. After all papers were coded, one researcher clustered the tasks (e.g. predictive business process monitoring) per lifecycle phase (Webster & Watson, 2002).

In the third step, the different codes were reconciled in a single table. Since there were multiple coders involved, we conducted an inter-coder reliability analysis using 25 randomly selected papers to ensure consistency among the coders. We calculated the percentage

**Table 2**
Overview of related literature reviews.

| Reference | Scope of review | Focus of review |
|---|---|---|
| Tiwari et al. (2008) | **General:** Papers published (1998) – 2005 | **Broad:** Process mining, specifically process discovery |
| Maita et al. (2015) | **General:** Journal and conference papers published 2004–2014 | **Specific:** NNs and SVMs in process mining |
| Márquez-Chamorro, Eduardo and Resinas, and and Ruiz-Cortés (2017) | **Specific:** Journal and conference papers, and book chapters published 2010–2017 | **Specific:** Predictive business process monitoring |
| Di Francescomarino et al. (2018) | **General:** Journal and conference papers published (2005) – 2018 | **Specific:** Predictive business process monitoring |
| Verenich et al. (2019) | **General:** Papers published 2005–2017 | **Very specific:** Remaining-time-prediction task in predictive business process monitoring |
| Teinemaa et al. (2019) | **General:** Papers published (2005) – 2017 | **Very specific:** Outcome prediction task in predictive business process monitoring |
| Stierle, Brunk, et al. (2021) | **Specific:** Journal and conference papers published (2014) – 2020 | **Very specific:** XAI approaches used in predictive business process monitoring |
| Neu et al. (2021) | **Very specific:** Papers published (2017) – 2020 | **Very specific:** Deep-learning approaches used in predictive business process monitoring |
| Rama-Maneiro et al. (2021) | **Very specific:** Papers published (2017) – 2020 | **Very specific:** Deep-learning approaches used in predictive business process monitoring |
| Taymouri, La Rosa, Dumas, and Maggi (2021) | **General:** Papers published 2003–2019 | **Specific:** Business process variant analysis |
| Folino and Pontieri (2021) | **General:** (i) Journal and conference papers, and book chapters published (2009) – 2020; (ii) journal and conference papers, and book chapters published (2008) – 2020 | **Specific:** (i) AI-based process-mining approaches exploiting domain knowledge; (ii) process-mining approaches addressing auxiliary AI tasks jointly with target process-mining tasks |
| Herm et al. (2021) | **Specific:** Papers published (2015) – 2020 | **Specific:** AI-based RPA |
| Wanner et al. (2020) | **General:** Papers published (2007) – 2018 | **Specific**: Combining complex event processing and ML |
| Ko and Comuzzi (2023) | **General:** Journal and conference papers published (2000) – 2021 | **Very specific**: Anomaly-detection task in pattern detection |
| **Our work** | **General**: Journal and conference papers published (1998) – 2022 | **Broad:** ML applications for tasks in all BPM lifecycle phases |

The year in round brackets indicates the year of the oldest paper in the final coding table.

agreement, Krippendorff's $\alpha$ (Krippendorff, 2018), and Fleiss' $\kappa$ (Fleiss & Cohen, 1973), which are established metrics for three or more coders (Lombard, Snyder-Duch, & Bracken, 2006), and found high inter-coder reliability. Further information on the analysis and the results can be found in Appendix B. The coders pointed out any ambiguities in the coding table during the coding process, which could be resolved later in open discussions to ensure uniformity.

For greater transparency, our coding table can be found in Appendix C and in an interactive concept matrix.

### 4.3. Interactive concept matrix

One aim of our systematic literature review is to make the data coded from the collected papers transparent and accessible in a useful manner. While the coding table in Appendix C fulfils the first aspect, its usability is limited because of its size (number of papers coded and the number of coding categories). Therefore, we created an interactive concept matrix (including a filterable table and two exemplary visualisations of the results)[2] that synthesises the concepts the literature considers from a descriptive perspective. The interactive concept matrix is structured in three sheets, each allowing the user to filter certain concepts to gain insights. On the first sheet "Concept development over time", the user sees how the concepts the literature considers evolve, for example, per BPM lifecycle phase. Using the second sheet, "Intersection of concepts", the user can gain insights into how two concepts intersect. One example can be the intersection of the BPM lifecycle phases and the ML paradigms, where the heatmap colour indicates the number of papers per intersection category, as shown in Fig. 5.

From the third sheet, "Complete coding table", the user can glean a holistic perspective of all concepts the literature considers and filter the complete concept matrix for certain values, such as specific authors, publication years, or lifecycle phases.

[2] The interactive concept matrix can be found here: https://literaturedashboardbpm.herokuapp.com/.

## 5. Results

This section presents an overview of ML applications in BPM along the six phases of the BPM lifecycle and the BPM tasks derived from our literature review. Fig. 6 shows the BPM lifecycle by Dumas et al. (2018), enriched with tasks we identified. In addition, descriptions of BPM lifecycle phases and definitions of BPM tasks in the view ML applications are provided.

### 5.1. Process identification

In the process identification phase, organisations collect information about their business processes. Tasks in this phase are identifying processes in the organisational landscape, determining which business processes can be optimised and whether an optimisation should happen within or outside the organisation. ML applications can support different tasks in this phase, especially by unlocking unstructured data, repairing missing data, and supporting outsourcing decision-making. Moreover, creating data representations facilitates thorough analyses in later phases of the BPM lifecycle. Fig. 7 summarises the tasks in the process identification phase.

#### 5.1.1. Business process outsourcing

A strategic task of process identification is to decide whether a business process or a part of a business process can be optimised regarding time or cost when outsourced to another organisation. ML applications can help identify process indicators that propose a high probability of successful outsourcing. Therefore, Ciasullo et al. (2018) propose a reinforcement-learning-based framework that supports outsourcing decisions of business processes.

#### 5.1.2. Event log creation

After adding process mining to many organisations' business intelligence portfolios, event logs gained importance quickly. Event logs are the standard data form for process-mining applications. While structured data in information systems can directly be compiled into an

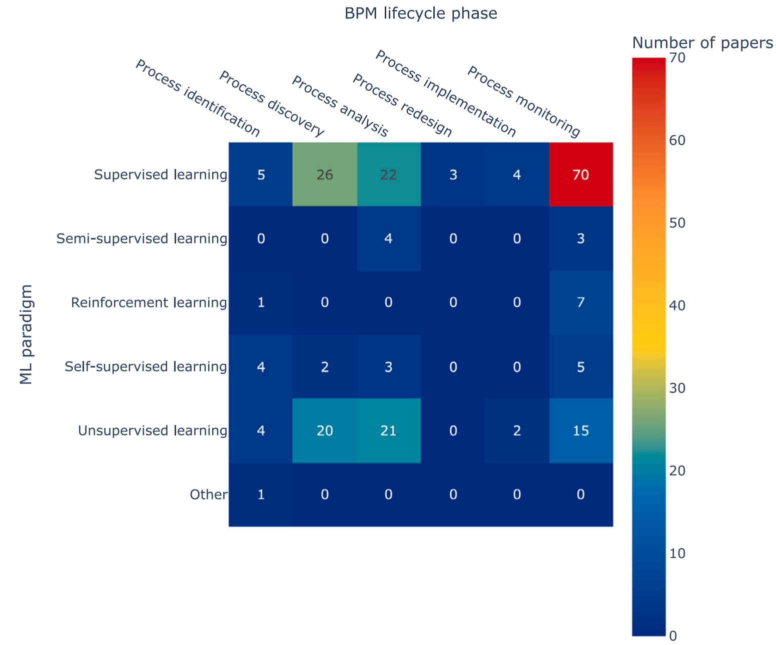


**Fig. 5.** Exemplary figure of the interactive concept matrix for the intersection of the BPM lifecycle phases and the ML paradigms.

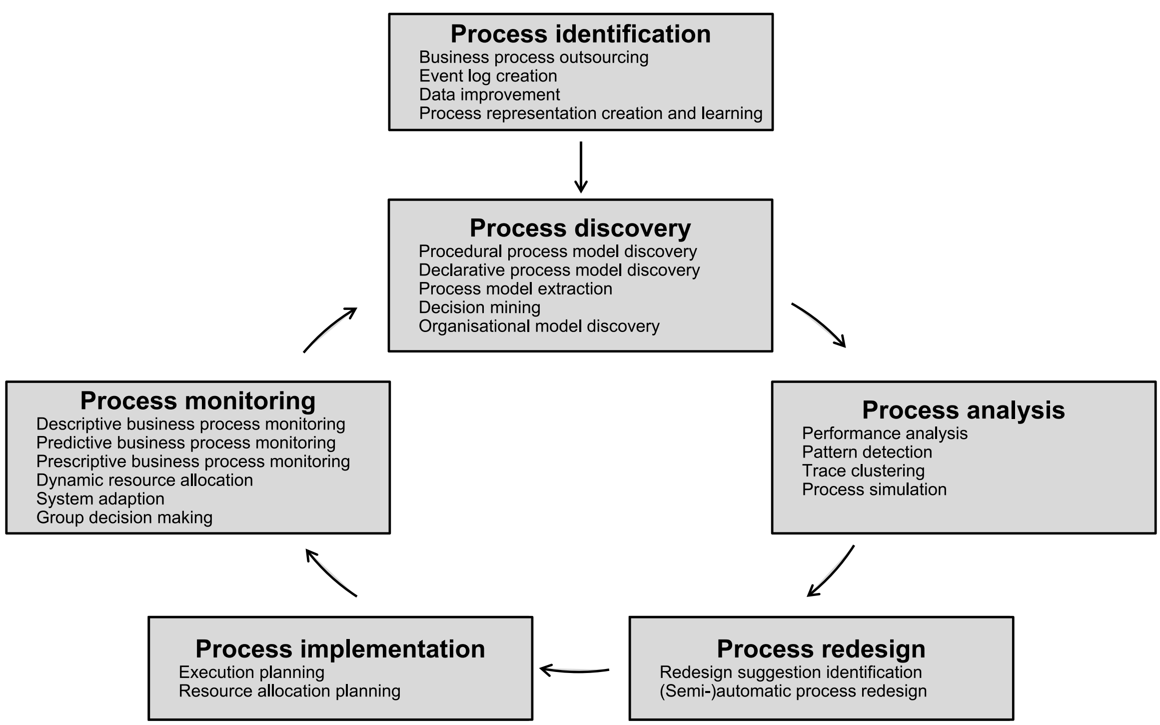


**Fig. 6.** BPM lifecycle of Dumas et al. (2018) enriched with the identified tasks.

event log, ML applications can also turn raw data (e.g. video, image, or plain text) into event logs. Therefore, Sim, Sutrisnowati, Won, Lee, and Bae (2022) propose a method to create an event log from raw event data using convolutional neural network (CNN) models. Tello, Gianini, Mizouni, and Damiani (2019) provide a framework that uses supervised-learning models for mapping low-level event data onto high-level process activities. An event log can also be created for further analysis by detecting process instances (i.e. sequences of events) in e-mails with a clustering model (Jlailaty, Grigori, & Belhajjame, 2017) or processing video data with a computer-vision-based approach (Kratsch, König, & Röglinger, 2022).

*5.1.3. Data improvement*

As part of process identification, process analysts identify process data in sources and sinks. Process data is often represented by event logs that hold a potential value but can lack quality when they include missing or incorrect data (Bose, Mans, & van der Aalst, 2013). ML applications can reconstruct missing values in event data or identify and replace incorrect event data. As ML applications for data improvement increase event-log quality, they leverage accurate analyses. Missing activities in event log data are repaired using a self-organising map model for trace clustering (Xu & Liu, 2019). Nolle, Seeliger, Thoma, and Mühlhäuser (2020) correct anomalies using gated recurrent unit

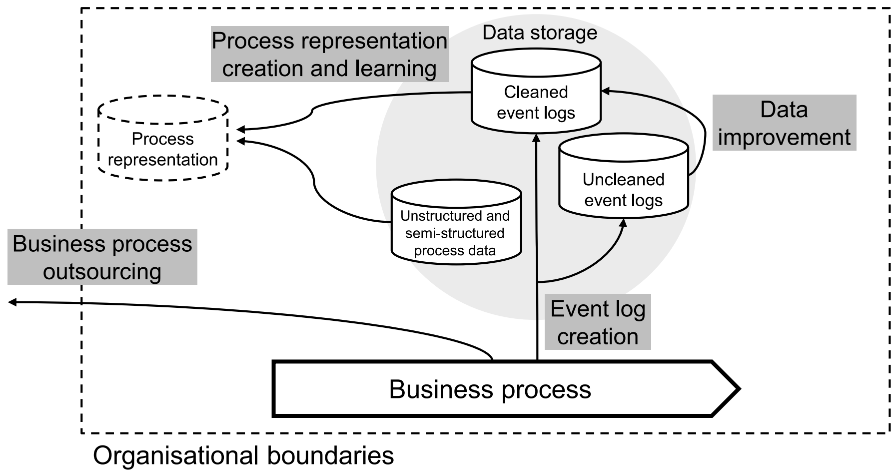


**Fig. 7.** Schematic representation of the tasks in the process identification phase.

(GRU) models with bi-directional beam search. Nguyen, Lee, Kim, Ko, and Comuzzi (2019) clean event log data using an autoencoder model.

#### 5.1.4. *Process representation creation and learning*

Process identification also deals with finding compact, abstract, and computational process representations on which ML can build applications for addressing BPM tasks in subsequent phases of the BPM lifecycle. Therefore, BPM research has developed ML applications to create and learn process representations from process data.

Supervised-learning approaches can produce dense and accurate process representations (Seeliger, Luettgen, Nolle, & Mühlhäuser, 2021) and event abstractions (Tax, Sidorova, Haakma, & van der Aalst, 2016). De Koninck, vanden Broucke, and De Weerdt (2018) propose self-supervised learning techniques to create representations of activities, traces, event logs, and process models. Guzzo, Joaristi, Rullo, and Serra (2021) also use self-supervised learning to create process representations considering multi-dimensional aspects of event log traces.

### 5.2. *Process discovery*

Process discovery focuses on generating as-is models that describe a process as it is executed according to event data. Process models can become complex depending on the number of variants and the length of traces in a process. Unsupervised learning applications can focus on process variants to improve process model comprehension. While ML applications can detect general process model structures in data, they can also mine specific types of process models, such as procedural and declarative models. Moreover, ML applications can optimise discovered process models regarding specific process model quality criteria. Consequently, ML applications support many tasks in the process discovery phase and automate it to a certain extent. Fig. 8 overviews the process discovery tasks.

#### 5.2.1. *Discovery of procedural process model*

Discovering procedural process models is a task to gain transparency over a process as it is executed. Discovered procedural process models can be complex, so they are often simplified using heuristics or filtering options. ML applications can identify process structures or optimise parameters and event-log sampling for discovery algorithms to create accurate and comprehensible procedural process models. Additionally, ML applications can choose the best-suited discovery algorithm based on the input data and produce specific procedural models (e.g. reference models). Early works in ML-based BPM, such as Herbst (2000), Herbst and Karagiannis (2000, 2004), focus on the discovery of process model structures using Bayesian algorithms.

Other studies address trace-clustering-based process model discovery. Greco, Guzzo, Pontieri, and Sacca (2004) propose an algorithm that incorporates a k-means model to cluster workflow executions, where a cluster is a global constraint of a discovered process. In a later work (Greco, Guzzo, Pontieri, & Sacca, 2006), the same authors present an extension of their algorithm for discovering conforming process models. García-Bañuelos, Dumas, La Rosa, De Weerdt, and Ekanayake (2014) propose a trace-clustering-based process discovery technique that allows users to control process model complexity. Closely connected to these studies, Qiao, Akkiraju, and Rembert (2011) present a clustering-based approach for retrieving business process models.

Process discovery applications that rely on genetic algorithms use target metrics like fitness (van der Aalst, De Medeiros, & Weijters, 2005), either fitness, replay, precision, generalisation, or simplicity (Buijs, van Dongen, & van der Aalst, 2012), or completeness, precision, and simplicity together (Vázquez-Barreiros, Mucientes, & Lama, 2014, 2015), to create optimised process models. Genetic algorithms can also be combined with graph-based representations to analyse complex processes (Turner, Tiwari, & Mehnen, 2008).

Other studies that consider discovery of procedural process model address various discovery tasks, such as the creation of process models for unseen event log data using graph neural network (GNN) models (Sommers, Menkovski, & Fahland, 2021), the creation of process models using hidden Markov models containing equations and rules (Sarno & Sungkono, 2016a), and, building on the latter, the inclusion of invisible prime tasks and parallel control-flow patterns (Sarno & Sungkono, 2016b).

Moreover, probabilistic process models using a Bayesian algorithm (Silva, Zhang, & Shanahan, 2005) are discovered. Lu, Zeng, and Duan (2016) discover synchronisation-core-based process models, where a decision-tree model is used to group traces into clusters, from which synchronisation cores of activity dependencies are derived. Ferreira and Gillblad (2009) discover process models from event stream data by estimating model parameters and case assignments using an expectation–maximisation procedure. Ferreira, Szimanski, and Ralha (2013) discover hierarchical process models that capture the relationships between macro-level activities and micro-level events using a hierarchical Markov model with expectation–maximisation-algorithm-fitted parameters.

Another group of studies discovers certain forms of process models, such as process trees using a k-means model to handle context data in addition to control-flow data (Shraga, Gal, Schumacher, Senderovich,

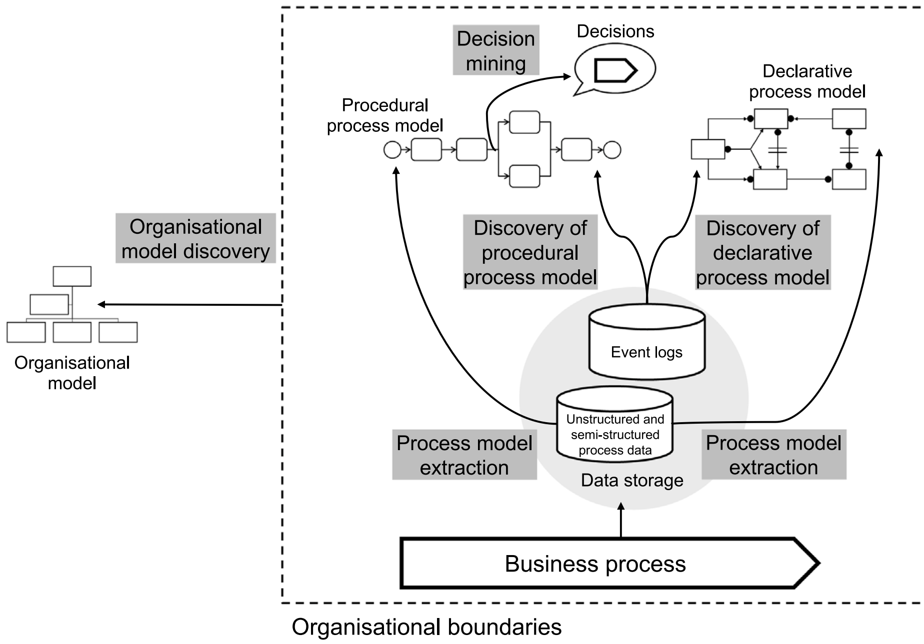


**Fig. 8.** Schematic representation of the tasks in the process discovery phase.

& Weidlich, 2020) or connections between events using a logistic regression model (Măruşter, Weijters, van der Aalst, & Van Den Bosch, 2002). Building on the latter, Măruşter, Weijters, van der Aalst, and Van Den Bosch (2006) propose a method that uses the RIPPER algorithm to create rule sets representing process-activity relations.

Finally, specific procedural process discovery applications are meta-process discovery for process discovery algorithm selection using an SVM model for reference model selection (Wang et al., 2012), reference model discovery using a genetic algorithm (Martens, Fettke, & Loos, 2014) or a clustering algorithm (Li, Reichert, & Wombacher, 2010), behavioural pattern model discovery using a clustering algorithm (Diamantini, Genga, & Potena, 2016), and discovery of configurable process model using a genetic algorithm (Buijs, van Dongen, & van der Aalst, 2013).

### 5.2.2. Discovery of declarative process model

In contrast to procedural process models, declarative models express specific rules, which should not be infringed during execution. The discovery of declarative process models finds constraints in an event log. ML applications support both discovering declarative models and consistency checking of such constraints. Leno, Dumas, and Maggi (2018) facilitate the discovery of multi-perspective declarative process models using a clustering model for selecting traces and a classification model to identify declarative constraints. Leno, Dumas, Maggi, La Rosa, and Polyvyanyy (2020) additionally present a redescription-mining-based approach for declarative constraint identification that uses two decision-tree models. Inductive logic programming allows for the discovery of declarative models (Chesani et al., 2009; Lamma, Mello, Montali, Riguzzi & Fabrizio and Storari, 2007) and learning integrity constraints for declarative modelling (Lamma, Mello, Riguzzi, Storari, Sergio, 2007). Maggi, Bose, and van der Aalst (2012) use the Apriori algorithm to find declarative constraints, and Maggi, Di Ciccio, Di Francescomarino, and Kala (2018) combine the Apriori algorithm with a sequence analysis algorithm to discover declarative constraints.

### 5.2.3. Process model extraction

Process models can be modelled in diagram software or paper drawings, or described in plain text. Interpreting these process models and translating them into computationally interpretable process models can be done with ML applications. Plain-text process descriptions can serve as foundations for extracting procedural process models using a hierarchical multi-grained deep neural network (DNN) (Qian et al., 2020) and declarative process models using a discovery algorithm that incorporates a BERT model (López, Strømsted, Niyodusenga, & Marquard, 2021). Polančič, Jagečič, and Kous (2020) extract digital process models from hand-drawn process diagrams using a CNN model. Kim, Suh, and Lee (2002) propose a document-based discovery approach that applies case-based reasoning to effectively reuse the design outputs.

### 5.2.4. Decision mining

When discovering processes, it is essential to understand decision points where process flows split. Decision mining comprises identifying decision points and conditions, which lead to a course of action. ML applications can automate identifying decision points and decision dependencies. Process models can be enhanced by using decision-tree models to detect dependencies in the data at the decision points in the process model, as Rozinat and van der Aalst (2006) do in one of the first approaches. Bazhenova, Buelow, and Weske (2016) use decision trees to semi-automatically derive decision models from event logs. Both Mannhardt, de Leoni, Reijers, and van der Aalst (2016) and Effendi and Sarno (2017) present a technique for discovering overlapping rules in event logs using decision-tree models. de Leoni, Dumas, and García-Bañuelos (2013) propose a technique for discovering branching conditions by combining decision-tree models with invariant discovery techniques. Lastly, Sarno, Sari, Ginardi, Sunaryono, and Mukhlash (2013) propose a decision-mining approach that uses a decision-tree model and considers multi-choice workflow patterns.

### 5.2.5. Organisational model discovery

Processes are not executed in isolation but in the context of an organisation. Consequently, event data contains information about organisational functions. ML applications can create models of the executing organisation based on event logs using clustering models (Yang, Ouyang, Pan, Yu, & ter Hofstede, 2018).

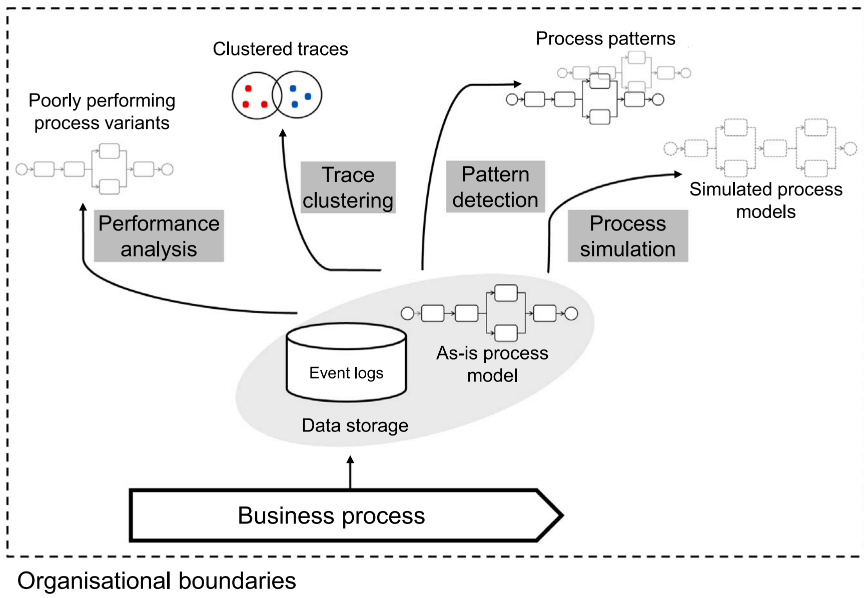


**Fig. 9.** Schematic representation of the tasks in the process analysis phase.

### 5.3. Process analysis

Process analysis assesses as-is processes using process models and process data. Process analysis takes a retrospective perspective in analysing a process. Process analysts determine key performance indicators (KPIs), assess process performance, look for problematic instances, and simulate processes to evaluate process redesigns. All these tasks can be supported with ML applications' pattern recognition capabilities, unsupervised learning capabilities for identifying similar process instances regarding control flow or performance, and predictive capabilities to determine realistic simulation parameters. Fig. 9 shows a schematic representation of the tasks in the process analysis phase.

#### 5.3.1. Performance analysis

When assessing the performance of processes and process instances, organisations determine KPIs to look for poorly performing processes and the cause of their performance failure. In doing this, analysts find starting points for process redesign. ML applications support performance analysis, for example, by repairing scarce performance data, aggregating KPIs, and identifying causal relations between data and process outcome. Berkenstadt, Gal, Senderovich, Shraga, and Weidlich (2020) propose an approach using techniques from queuing theory and supervised learning to predict process performance indicators. Wetzstein et al. (2009) use rule-based learning to recompose low-level process indicators into high-level process metrics. To enable fairness-aware process mining and performance analysis, Qafari and van der Aalst (2019) add situation-specific discrimination to event logs and train decision-tree models. Es-Soufi, Yahia, and Roucoules (2016) propose an approach that detects process patterns using process mining and creates performance-related predictions per activity of a process pattern using a supervised-learning model.

Training logical decision-tree models based on logical event log encoding (Ferreira & Vasilyev, 2015) and uplift tree models (Bozorgi, Teinemaa, Dumas, La Rosa, & Polyvyanyy, 2020) can reveal causal relationships in process performance and specific control-flow actions. Savickas and Vasilecas (2018) use belief network models to analyse process performance with domain-specific data. In contrast to the previously mentioned approaches, Theis and Darabi (2020) compute the generalisability of process models using generative adversarial network (GAN) models. To support performance analyses in general, de Leoni, van der Aalst, and Dees (2014), de Leoni et al. (2016) present a process analysis framework to correlate different process characteristics using regression and decision-tree models.

#### 5.3.2. Pattern detection

Process analysis further strives to identify favourable and problematic process patterns. Before pattern recognition capabilities of ML applications could be leveraged, process analysts manually compared process variants to find workarounds, anomalous behaviour or time series patterns, which could be improved or transferred to other instances. Process analysts can thereby provide input for redesigning process variants. Analysts can use ML applications with autoencoder models to identify anomalies even when they lack domain knowledge, as Nolle, Luettgen, Seeliger, and Mühlhäuser (2018) do without, and Krajsic and Franczyk (2020) do within, a lambda architecture designed for event steams. Another approach uses a word2vec model to encode activities in event log data and a (one-class) SVM model to detect anomalies (Junior, Ceravolo, Damiani, Omori, & Tavares, 2020). Rogge-Solti and Kasneci (2014) detect temporal anomalies in the activity execution duration using Bayesian models.

Cuzzocrea, Folino, Guarascio, and Pontieri (2015, 2016a) detect deviant process instances using multi-view, multi-dimensional ensemble learning in event logs. Building on their techniques, Cuzzocrea, Folino, Guarascio, and Pontieri (2016b) present a probabilistic-based framework for robust detection of deviance. Stierle, Weinzierl, Harl, and Matzner (2021) derive activity relevance-scores from an attention-layer of a GNN model that detects deviant process instances.

Weinzierl, Wolf, Pauli, Beverungen, and Matzner (2022) present a method for detecting workarounds using an autoencoder model that removes noisy process instances and a CNN model that maps process instances to workaround classes. Yeshchenko, Di Ciccio, Mendling, and Polyvyanyy (2019) detect concept drifts in event log data using visual analytics and time-series-based clustering. Finally, supervised learning and conformance checking (Valdés, Céspedes-González, & Pou, 2022a) or supervised learning and graph kernels (Valdés, Céspedes-González, & Pou, 2022b) can prove useful in identifying time-series patterns in event log data.

#### 5.3.3. Trace clustering

Clustering approaches can automatically group process instances or instance variants to analyse processes quickly. Improvements, problems, and advantages identified in one variant may apply to other instances or instance variants in the same cluster. Therefore, clustering traces can support process analysts.

The most popular trace-clustering approach in BPM is distance-based trace clustering. Distance-graph models (Ha, Bui, & Nguyen, 2016), graph similarity metrics (De Weerdt, vanden Broucke, Vanthienen, & Baesens, 2012), and similarity in heterogeneous information networks (Nguyen, Slominski, Muthusamy, Ishakian, & Nahrstedt, 2016) allow for distance-based trace clustering. Delias, Doumpos, Grigoroudis, and Matsatsinis (2019) take the outranking relations theory into account for trace clustering. Other studies address trace clustering using a co-training-based strategy with multiple trace profiles (Appice & Malerba, 2015), frequent-item-set mining (Seeliger, Nolle, & Mühlhäuser, 2018), and a Levenshtein-distance-based approach (Bose & van der Aalst, 2009) to consider the process context. High-level process analysis is achieved by employing hierarchical clustering (Jung, Bae, & Liu, 2008, 2009) or clustering using graph partitioning (Sarno, Ginardi, Pamungkas, & Sunaryono, 2013) based on graph similarity of different business process models.

Previous research also uses model-based trace clustering and approaches that include expert and domain knowledge via process model metrics (De Koninck, Nelissen, Baesens, Snoeck & Monique and De Weerdt, 2017), the extension with must-link and cannot-link relationships between process instances (De Koninck, Nelissen, Baesens, Snoeck, & De Weerdt, 2021), and active learning (De Weerdt, vanden Broucke, Vanthienen, & Baesens, 2013). Super-instance-based trace clustering enables combining distance- and model-based clustering (De Koninck & De Weerdt, 2019). Boltenhagen, Chatain, and Carmona (2019) use distance-based quality criteria in model-based trace clustering.

Prior research also describes certain applications of trace clustering. Folino, Guarascio, and Pontieri (2015) use a logical trace-clustering model to create comprehensible process models. Varela-Vaca, Galindo, Ramos-Gutiérrez, Gómez-López, and Benavides (2019) apply trace clustering to mine configuration flows optimised for a particular type of user. Yang et al. (2017) use distance-based clustering to set up a recommender system. Ferreira, Zacarias, Malheiros, and Ferreira (2007) apply a mixture of Markov chains learned with the expectation-maximisation algorithm to create traces from identified tasks. De Koninck, De Weerdt, and vanden Broucke (2017) present a trace clustering approach with instance-level explanations.

#### 5.3.4. Process simulation

Process analysis assesses potential process changes by simulating redesigned process models. ML applications can aid in creating realistic simulation conditions, as the conditions are learned from a mass of data instead of hard-coded rules. While Camargo, Dumas, and González-Rojas (2022) facilitate data-driven process simulation using two long short-term memory (LSTM) models, one for processing time prediction and one for waiting time prediction, Khodyrev and Popova (2014) use decision and regression tree models to predict KPI values for short-term process simulation.

### 5.4. Process redesign

Process redesign's goal is to improve an as-is process by applying process changes to the model and creating a to-be process model. Process designers use, for example, experience-based redesign heuristics, describing concrete measures to make a process efficient or less prone to error. ML applications can support process redesigners by automatically suggesting process redesign measures based on data or by performing (semi-)automatic process redesign, where more appropriate or optimised process designs are identified. Fig. 10 provides an overview of these tasks.

#### 5.4.1. Redesign suggestion identification

A task in process redesign is to elaborate redesign suggestions based on ideas, heuristics, or new technology. ML applications can identify unfavourable patterns in process instances and how to redesign them, for example, geared towards a certain target (e.g. customer satisfaction or throughput time). Therefore, research in this area addresses redesign suggestion identification. Mustansir, Shahzad, and Malik (2022) use language models and DNN models to detect suggestions for process redesign in textual customer feedback data.

#### 5.4.2. (Semi-)automatic process redesign

A subset of ML applications in process redesign focuses on semi-automatic process redesign. More concretely, ML applications addressing this task use genetic algorithms to perform multi-objective optimisation of business processes (Vergidis, Tiwari, Majeed, & Roy, 2007) or to determine an objective prioritisation of process redesigns (Afflerbach, Hohendorf, & Manderscheid, 2017).

### 5.5. Process implementation

Redesigned process models must be integrated into an information system before they can be executed. Resources need to be assigned to redesigned tasks, staff needs to be trained, and execution plans need to be rolled out. ML-supported process implementation tasks include creating execution and staffing plans from process models and data and resource allocation planning. Fig. 11 shows the tasks of the process implementation phase.

#### 5.5.1. Execution planning

Transferring from a to-be process model to process implementation requires setting up execution plans. Process models, including decision points and process semantics, can be used to create execution plans. This is tedious work, but ML applications can approximate an optimal execution plan from a detailed model. Therefore, Bae, Lee, and Moon (2014) propose a method to enrich BPM structures with alternative process paths and semantics, after which a genetic-based algorithm can find an execution plan that enhances performance.

#### 5.5.2. Resource allocation planning

After having set execution plans, resources, such as machines, employees, storage spaces or production lines, must be assigned to cases. Process owners define who or what is employed in which process variant. To make decisions during planning, ML applications can provide suggestions that optimise resource utilisation or throughput time. Process owners can then pick a suggested resource allocation plan and adjust it if needed. ML applications thereby eliminate trial and error approaches. Resource allocation planning is realised based on the prediction of execution routes using a naïve Bayes model (Kazakov, Novikov, Kulagina, & Shlapakova, 2018), the resource decision mining from events using classification models and heuristics (Senderovich, Weidlich, Gal, & Mandelbaum, 2014), the examination of process instance scheduling using a genetic-based algorithm (Xu, Liu, Zhao, Yongchareon, & Ding, 2016), or the recommendation of resource allocation candidates using classification models (Liu, Cheng, & Ni, 2012). Finally, Delcoucq, Dupiereux-Fettweis, Lecron, and Fortemps (2022) apply a bi-dimensional clustering approach for resource allocation planning, where resources are clustered based on similar behaviour in the process, and activities are clustered based on the executing resource.

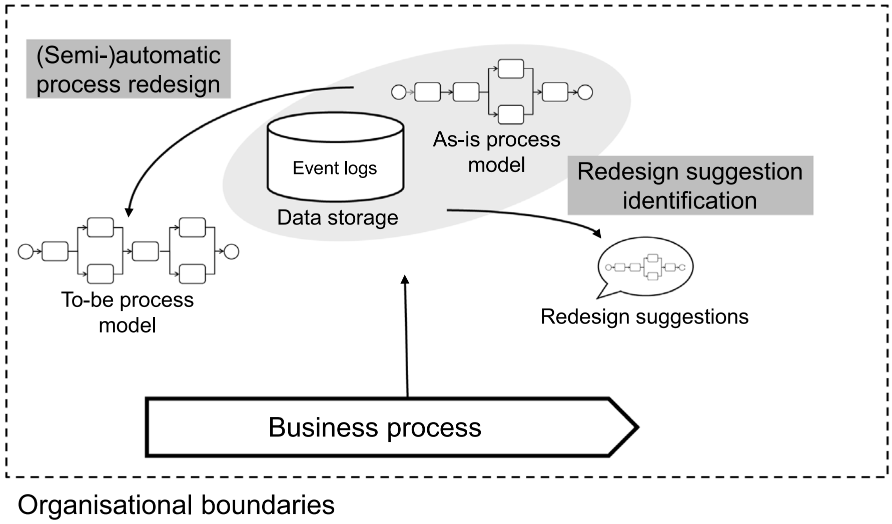


**Fig. 10.** Schematic representation of the tasks in the process redesign phase.

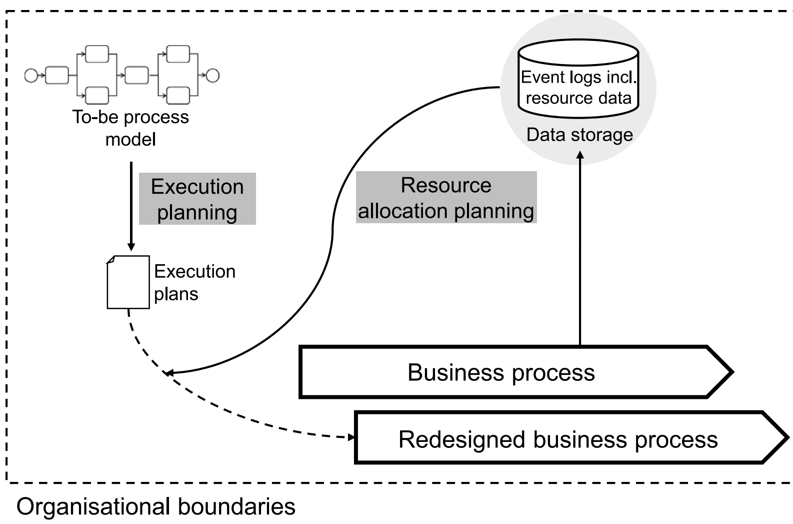


**Fig. 11.** Schematic representation of the tasks in the process implementation phase.

### 5.6. Process monitoring

After the redesigned processes are implemented, they are executed, and how well they perform in terms of their performance and objectives must be determined. Process monitoring challenges analysts because parallel cases must be overseen. Additionally, cases, which look fine momentarily, can turn out problematic later on. ML applications can recognise patterns in event data that indicate problematic outcomes, even when a case is in a good state. ML applications can help analysts identify cases for further analysis in large process data sets or provide decision support to process users, enabling them to execute processes with higher performance. Executing business processes can also be optimised regarding KPIs or process conformance. ML applications in process monitoring can further automate noticing an appropriate point for starting another BPM lifecycle iteration by detecting concept drifts in processes. Fig. 12 provides a schematic overview of the tasks and their relationship.

#### 5.6.1. Descriptive business process monitoring

Descriptive business process monitoring determines KPIs of running process instances and provides real-time information about reactive actions that could increase process performance. It can also detect concept drifts in processes and identify anomalous behaviour in process instances. ML applications can recognise patterns in high-volume data. In doing that, they help to find entry points for reiterating the BPM lifecycle to adapt to concept drifts or positive anomalies, or to counteract poor process behaviour.

Montani and Leonardi (2012) retrieve traces via a k-nearest-neighbour model and cluster them using a hierarchical clustering model to detect changes in event log data.

Online detection of anomalies is addressed using graph autoencoder models (Huo et al., 2021) or supervised learning, specifically, a random forest model, extreme gradient-boosting model, or an LSTM model (Lee, Lu, & Reijers, 2022). Other studies apply a DBSCAN model to cluster process instances into common and anomalous behaviour (Junior, Tavares, da Costa, Ceravolo, & Damiani, 2018), or a DenStream model (Cao, Estert, Qian, & Zhou, 2006) to cluster process instances with a graph-based representation including control flow and time (Tavares, Ceravolo, da Costa, Damiani, & Junior, 2019) and additional attributes (Ceravolo, Damiani, Schepis, & Tavares, 2022). Krajsic and Franczyk (2021) create a latent space representation of event data with a variational autoencoder, cluster the representation with a k-means model, and use either an isolation forest, a local outlier factor, or a one-class SVM model for each cluster to detect anomalies. For

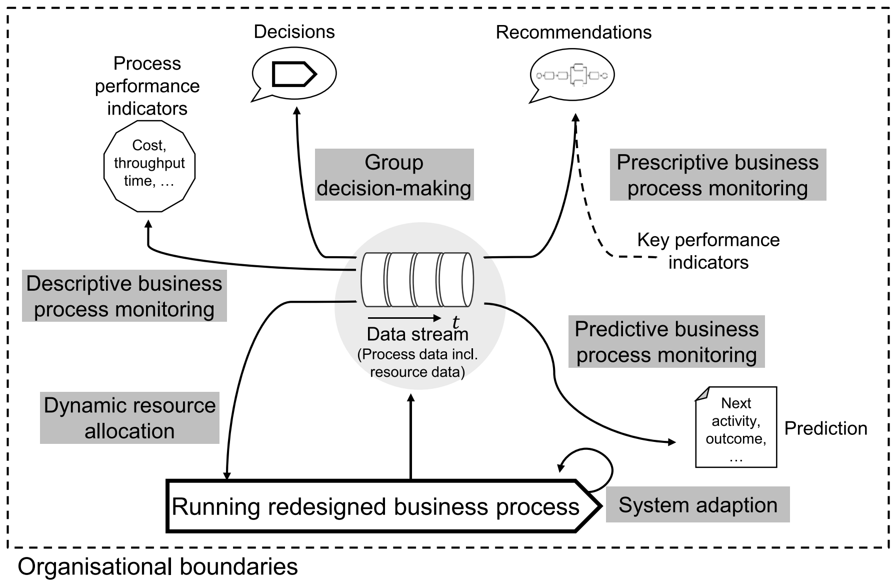


**Fig. 12.** Schematic representation of the tasks in the process monitoring phase.

their part, Nolle, Seeliger, and Mühlhäuser (2018) and Nolle, Luettgen, Seeliger, and Mühlhäuser (2019) determine anomaly scores based on the output of a GRU model and map them onto types of anomalies, while Lee, Burattin, Munoz-Gama, and Sepulveda (2021) check process conformance online using a hidden Markov model.

*5.6.2. Predictive business process monitoring*

In contrast to descriptive monitoring, predictive business process monitoring shifts to an ex-ante perspective by predicting different process properties of running process instances. With these predictions, process users can take proactive or corrective actions to mitigate risks or improve the performance of process executions. Predictive business process monitoring pursues multiple objectives including future activities, outcomes, remaining time, or a combination of objectives. Those approaches facilitate early warning systems.

A common prediction task is the next activity prediction, which deep-learning-based techniques address. For the next activity prediction, studies employ LSTM models (Evermann, Rehse, & Fettke, 2017; Pasquadibisceglie, Appice, Castellano, & Malerba, 2021), CNN models (Di Mauro, Appice, & Basile, 2019; Heinrich, Zschech, Janiesch, & Bonin, 2021; Pasquadibisceglie, Appice, Castellano, & Malerba, 2019, 2020), multi-layer perceptron (MLP) models (Mehdiyev, Evermann, & Fettke, 2020; Theis & Darabi, 2019), and transformer models (Heinrich et al., 2021; Philipp, Jacob, Robert, & Beyerer, 2020). Some works use deep-learning models and provide explainable next-activity predictions. Wickramanayake et al. (2022) (and their earlier version (Sindhgatta, Moreira, Ouyang, & Barros, 2020)) propose LSTM models that differ in the attention mechanism they use for the model explanation, Hsieh, Moreira, and Ouyang (2021) use an NN architecture computing dynamic features through an LSTM and static features through an MLP and generate explanations for predictions via the model-agnostic counterfactual algorithm DiCE (Mothilal, Sharma, & Tan, 2020), Hanga, Kovalchuk, and Gaber (2020) use an LSTM model and infer from it business process models to explain the model's decision-making process, and Gerlach, Seeliger, Nolle, and Mühlhäuser (2022) use a GRU model and infer from it multi-perspective likelihood graphs for the same purpose. A smaller group of studies proposes Bayesian approaches, that is, regularised probabilistic finite automata models (Breuker, Matzner, Delfmann, & Becker, 2016) or Bayesian network models (Brunk et al., 2021; Pauwels & Calders, 2020). Other studies compute transition probabilities in process models using decision-tree models (Lakshmanan, Shamsi, Doganata, Unuvar, & Khalaf, 2015) or propose factorisation machines to predict next activities (Lee, Parra, Munoz-Gama, & Sepulveda, 2018). Pauwels and Calders (2021) introduce incremental learning strategies for updating next-activity-prediction models. Finally, instead of only predicting the next activity, Bernard and Andritsos (2019) address the remaining activity sequence prediction (or suffix prediction) using clustering and SVM models to provide transparent predictions.

Another prediction task is outcome prediction, whose outcome can be binary (e.g. the violation of a business rule), multi-class (e.g. multiple conditions), or continuous (e.g. a KPI value). To predict binary process outcomes in the form of conditions, studies use a CNN model (Pasquadibisceglie, Appice, Castellano, Malerba, & Modugno, 2020), an LSTM model (Wang, Yu, Liu, & Sun, 2019) or a random forest or logistic regression model including features created with text models (Teinemaa, Dumas, Maggi, & Di Francescomarino, 2016). Other studies with the same aim use a gated GNN model, from which attention-based explanations are extracted (Harl, Weinzierl, Stierle, & Matzner, 2020), a model built with an evolutionary algorithm and decision rules generated to explain the model's predictions (Márquez-Chamorro, Resinas, Ruiz-Cortés, & Toro, 2017), or a neuro-fuzzy model, from which explainable rules are directly extractable (Pasquadibisceglie, Castellano, Appice, & Malerba, 2021). Rizzi, Di Francescomarino, and Maggi (2020) predict the violation of business rules with a random forest model and use post-hoc explanations created with LIME (Ribeiro, Singh, & Guestrin, 2016) to manipulate data to improve model accuracy. Di Francescomarino, Dumas, Maggi, and Teinemaa (2016) use a DBSCAN model as a density or a model-based clustering approach to cluster partial process instances. Then they apply a decision tree or a random forest model per cluster to predict whether a running process instance violates a business rule. Folino, Guarascio, and Pontieri (2012) and Bevacqua, Carnuccio, Folino, Guarascio, and Pontieri (2013) predict violations of service-level agreements. Folino et al. (2012) first cluster running process instances. Then, with a cluster-dependent finite state machine model, the remaining time is predicted, and a rule is applied to the predictions to determine the violations. Bevacqua et al. (2013) follow a similar approach but apply a rule to remaining-time or remaining-step predictions generated with regression models to determine the violations. Cuzzocrea, Folino, Guarascio, and Pontieri (2019) predict violations of aggregated performance constraints. First, running process instances of a window are clustered into variants. Afterwards, with a cluster-dependent linear regression or a k-nearest neighbour model, a performance indicator is predicted for each running process instance. Then, a Gaussian-process regression model predicts an aggregated performance indicator over future window slots, and a rule is applied to the predictions to determine the violations. As multi-class process outcomes, Folino, Greco, Guzzo, and Pontieri (2011) predict behavioural classes of running process instances using a decision-tree model. Lastly, continuous process outcomes are predicted as customer delays with non-linear regression or regression tree models (Senderovich, Weidlich, Gal, & Mandelbaum, 2015), as delivery times with various regression models and concept-drift handling (Baier, Reimold, & Kühl, 2020), and as process performance at the process-model level using a CNN, LSTM, or long-term recurrent CNN model (Park & Song, 2020).

Prediction of the remaining time is another common prediction task. The remaining time of running process instances is predicted using shallow ML models considering inter-case features (Klijn & Fahland, 2020; Senderovich, Di Francescomarino, Ghidini, Jorbina, & Maggi, 2017), shallow ML models considering news sentiments as external process context (Yeshchenko, Durier, Revoredo, Mendling, & Santoro, 2018), data-aware transition systems annotated with shallow ML models without (Polato, Sperduti, Burattin, & de Leoni, 2014) and with concept-drift adaption (Firouzian, Zahedi, & Hassanpour, 2019b), Bayesian NN models (Weytjens & De Weerdt, 2021), LSTM models (Alves, Barbieri, Stroeh, Peres, & Madeira, 2022), and MLP models with entity-embedding (Wahid, Adi, Bae, & Choi, 2019).

Some studies do not focus on a specific prediction task but address multiple prediction tasks. Appice, Di Mauro, and Malerba (2019) predict next activities, next timestamps, and remaining time using various feature encodings, learning approaches, and shallow ML models. Extending their earlier work (Polato et al., 2014), Polato, Sperduti, Burattin, and de Leoni (2018) present three methods that rely on naïve Bayes and support vector regression models to predict the remaining time, one of which can also be used to predict suffixes. In a pioneering work using deep learning in the domain, Tax, Verenich, La Rosa, and Dumas (2017) predict next activities, next timestamps, remaining times, and suffixes using a multi-task LSTM model. This approach (Tax et al., 2017) has been be extended in various ways, such as with embedding layers for efficient computation (Camargo, Dumas, & González-Rojas, 2019), linear temporal logic rules in the post-processing for accurate suffix predictions (Di Francescomarino, Ghidini, Maggi, Petrucci, & Yeshchenko, 2017), or text models in the pre-processing to compute textual data (Pegoraro, Uysal, Georgi, & van der Aalst, 2021). Galanti, Coma-Puig, de Leoni, Carmona, and Navarin (2020) use an LSTM model to generate remaining time, activity occurrence, and cost predictions, and apply SHAP (Lundberg & Lee, 2017) to create post-hoc explanations for the predictions. Other deep-learning-based approaches predict process failure and next activities using a DNN model with convolutional and recurrent layers (Borkowski, Fdhila, Nardelli, Rinderle-Ma, & Schulte, 2019), or next activities and suffixes using an LSTM-based encoder–decoder model with an attention layer (Jalayer, Kahani, Beheshti, Pourmasoumi, & Motahari-Nezhad, 2020). Pfeiffer, Lahann, and Fettke (2021) encode different event information as gramian angular fields (i.e. 2D-images), learn a representation based on this encoding using a CNN model, and fine-tune the trained model towards various prediction tasks. Taymouri, La Rosa, Erfani, Bozorgi, and Verenich (2020) use a GAN model to predict next activities and next timestamps, and Taymouri, La Rosa, and Erfani (2021) combine these with beam search to predict suffixes and remaining time.

Other studies propose solutions for certain prediction tasks — predicting activity ordering (Verenich, Dumas, La Rosa, Maggi, & Di Francescomarino, 2016), enterprise social networks (Pham, Ahn, Kim, & Kim, 2021), or process outcomes based on scarcely labelled event logs (Folino, Folino, Guarascio, & Pontieri, 2022).

#### 5.6.3. Prescriptive business process monitoring

Prescriptive business process monitoring aims at steering processes towards specific optimisation objectives. To achieve this, such ML applications recommend taking action in process instances. Other approaches warn process owners when a process instance requires human attention; for instance, if a negative outcome is expected. ML applications point analysts' attention to important process instances and provide instructions to process users on how they can carry out their work effectively and efficiently.

One stream of research investigates warnings that are based on process predictions. One indicator of a triggered warning can be the prediction of a process outcome, as Teinemaa, Tax, de Leoni, Dumas, and Maggi (2018) propose using a random forest model or a gradient-boosted tree model in combination with a cost model. Another indicator can be the prediction of a process performance indicator, as suggested by Kang, Kim, and Kang (2012) using an SVM model. Bozorgi, Teinemaa, Dumas, La Rosa, and Polyvyanyy (2021) use an orthogonal random forest model to recommend whether and when to apply an intervention in a process to decrease running process instances' cycle times.

Another stream of research investigates recommendations for actions that need to be taken in response to process predictions. The recommendation of the next best actions is addressed by predicting process risk using a decision-tree model and applying integer linear programming (Conforti, de Leoni, La Rosa, van der Aalst, & ter Hofstede, 2015). Next-best-action recommendations are also determined based on next activity predictions generated by an LSTM model, KPI information, and declarative process model simulation (Weinzierl, Dunzer, Zilker, & Matzner, 2020). Alternatively, they are based on KPI value predictions obtained using a random forest, SVM, or decision-tree model and a transition-system abstraction (de Leoni, Dees, & Reulink, 2020). Khan, Ghose, and Dam (2021) model the next-best-action recommendation as a Markov decision process and apply deep Q-learning to learn the optimal policy for solving this decision problem.

#### 5.6.4. Dynamic resource allocation

While resource allocation planning is an important task in process implementation, dynamic resource allocation is part of daily operations. Because of changes in order situations or resource availability beyond expectation, organisations need to reallocate their resources to activities during process execution. This task can be formulated as an optimisation problem. Therefore, ML applications optimise resource allocation taking run-time constraints into account, and assist process owners in finding optimal resource utilisation. Some papers apply shallow ML models to map activities to appropriate workers (Liu, Wang, Yang, & Sun, 2008), use naïve Bayes models to predict the performance of human resources before incoming jobs are assigned to them based on the predictions (Wibisono, Nisafani, Bae, & Park, 2015), or apply a k-means model to group process instances at critical activities before these are mapped to available resources (Pflug & Rinderle-Ma, 2016). Other works use a DBSCAN model and an ensemble of MLP models to allocate human resources based on team faultiness (Zhao, Pu, & Jiang, 2020), or an agglomerative hierarchical clustering model and a k-means model to recommend resources' task preferences (Zhao, Liu, Dai, & Ma, 2016). Studies applying reinforcement learning, specifically the Q-learning algorithm, allocate resources via policies that consider time and costs (Huang, van der Aalst, Lu, & Duan, 2011) or time and workload balancing (Firouzian, Zahedi, & Hassanpour, 2019a) as optimisation objectives. Park and Song (2019) use an LSTM model to predict the next activities and processing time and use the predictions to build a bipartite graph, laying the basis for solving a minimum-cost and maximum-flow problem for resource allocation.

#### 5.6.5. System adaption

Process and system adaptations during execution may be required to adjust to changing process environments. Such adjustments do not necessarily need to trigger an iteration of the full BPM lifecycle but can be implemented instantaneously. Self-adaptive systems approaches find adaptations, which improve process performance in the monitoring phase. In contrast to prescriptive business process monitoring, self-adaptive systems apply permanent changes to a process instead of improving a process instance. Studies address self-adaptive systems that increase process reliability using an ensemble of LSTM models (Metzger, Neubauer, Bohn, & Pohl, 2019) and reinforcement learning (Metzger, Kley, & Palm, 2020). Saraeian, Shirazi, and Motameni (2019) use an MLP model to estimate uncertain characteristics to control and optimise an autonomous BPM system architecture. Huang, van der Aalst, Lu, and Duan (2010) adapt work distributions based on reinforcement learning, consider process performance goals as optimisation objectives, and learn work distribution policies during process condition changes. Samiri, Najib, El Fazziki, and Boukachour (2017) combine reinforcement learning and forecasting techniques to adapt workflows automatically.

#### 5.6.6. Group decision-making

Group decision-making is only addressed by De Maio, Fenza, Loia, Orciuoli, and Herrera-Viedma (2016), who use reinforcement learning to learn the weighting of decision-makers (e.g. heterogeneous experts) for a decision activity (e.g. select a supplier) based on past process executions considering context and business process performances.

## 6. Synthesis of results and derived findings

In this section, we synthesise the results presented in the previous section for each BPM lifecycle phase. Based on the synthesised results along with the interactive coding table, we derived findings, which we present in Section 6.7.

### 6.1. Process identification

As data input, most ML applications in process identification use event logs, including time or resource information. Unlike other BPM lifecycle phases, input data types include unstructured raw event, video, and text data for event log creation.

Moreover, ML applications that address process representation creation and learning employ language models trained with NNs such as word2vec (e.g. De Koninck et al., 2018) or use embedding layers in DNN models (e.g. Guzzo et al., 2021) for feature extraction.

In process identification, ML applications follow the supervised- and unsupervised-learning approach. Additionally, ML applications deal with self-supervised learning because NNs are employed to create and learn process representations. These NNs are designed to extract a label from underlying data automatically (e.g. Nguyen et al., 2019). While around half of the ML applications in this phase adopt deep learning, almost all use NN models. This is because of the flexibility of designing NNs used in process representation creation and learning, event log creation, and data improvement applications.

To assess the performance of models in the tasks process representation creation and learning and event log creation, clustering models are trained with the created process data instances, and clustering metrics are calculated for the clustered instances (e.g. rand index or mutual information score). For this, mainly publicly available event logs are used such as the ones from the BPI challenges.[3]

[3] For example, see Business Process Intelligence Challenge: https://www.tf-pm.org/resources/logs.

### 6.2. Process discovery

In process discovery, most ML applications use event logs that only include control-flow information as data input to discover process models. Additionally, unstructured data, such as text from e-mails or image data, are used because ML applications can extract process models from unstructured data.

ML applications addressing process model extraction use language models trained with NNs, such as a BERT model (López et al., 2021) for feature extraction from unstructured data. For other tasks of this phase, like discovery of procedural process models, feature extraction refers to transforming the control flow of event logs into instance types, such as activity sequences or activity graphs.

Concerning the ML paradigm, ML applications of this phase follow the supervised-learning and unsupervised-learning approaches at about the same rate. Because of the lack of label information, unsupervised learning, however, is more common. For example, in discovery of procedural process model, process activity relations are mined in an unsupervised manner using the RIPPER algorithm (Măruşter et al., 2006). Regarding ML concepts, deep learning and transfer learning are adopted in process model extraction applications. While deep learning facilitates learning process model patterns from unstructured input data, transfer learning supports learning such patterns through pre-trained models. Moreover, clustering, Bayesian, decision-tree, genetic, and rule-based algorithms are used in process discovery. Clustering algorithms facilitate generating comprehensible process models for further analysis, as, for example, Greco et al. (2006) do. Bayesian-based algorithms can induce process structures from data (e.g. Herbst, 2000). Decision-tree-based algorithms are applied to create decision-tree models in decision-mining applications. Genetic-based algorithms are used to find optimal process models according to one or more selected process model quality criteria, as, for example, Vázquez-Barreiros et al. (2015) do. Rule-based algorithms are applied to learn rules from event logs; such rules represent a declarative process model in process discovery (e.g. Maggi et al., 2012).

The model assessment uses process discovery metrics from process mining, such as fitness, precision or generalisation, indicating how good the ML-created process models are. Often, own event logs are used for model assessment, but also publicly available logs.

### 6.3. Process analysis

Most ML applications use event logs, including time, resource, or data-related context information. Text data or process models are only occasionally used as data input. Some approaches also integrate domain knowledge.

ML applications in process analysis propose individual feature engineering techniques for event log data, and commonly used techniques cannot be observed, even when considering single tasks.

Like in process discovery, the supervised-learning and unsupervised-learning approach is addressed at about the same rate. This ratio can be explained by trace clustering applications, which are prominent in process analysis and address unsupervised-learning problems (e.g. Bose & van der Aalst, 2009). Analysis applications adopt deep, ensemble, active, meta, and multi-view learning. Deep learning is used to learn accurate models for process analysis tasks by considering intricate structures in event log data, as Camargo et al. (2022) do for process simulation. Ensemble learning is adopted to detect patterns based on the outcome of models, which are trained with different supervised ML algorithms (e.g. Cuzzocrea et al., 2016a). Active learning is used to increase ML models' accuracy by selecting process instances from an event log based on a metric, as in De Weerdt et al. (2013) for trace clustering, or human judgement, as in Cuzzocrea et al. (2016b) for pattern detection. A meta-model is learned based on the outcome of several base models to detect patterns in event log data (e.g. Cuzzocrea et al., 2016a). Multi-view learning is adopted to train multiple shallow

ML models based on event logs, where each model refers to a view for accurate pattern detection (e.g. Cuzzocrea et al., 2015) or complexity-reduced trace clustering (Appice & Malerba, 2015). ML applications in process analysis use unsupervised clustering algorithms for trace clustering. Further, as in process discovery, we observe that rule-based algorithms are employed here to learn models as a set of explainable rules from event log data; these rules reveal the cause of delayed processes (e.g. Ferreira & Vasilyev, 2015).

Process discovery metrics (e.g. fitness, precision, or generalisation) or clustering metrics (e.g. cluster set entropy or mutual information score) are calculated to assess the performance of clustering models employed in trace clustering applications. For applications addressing other tasks in process analysis, common supervised ML metrics are calculated for model assessment. Mainly publicly available event logs, but sometimes also own event logs are used.

### *6.4. Process redesign*

ML applications for (semi-)automatic process redesign receive business process designs consisting of activities, connections, and routing decisions (e.g. Afflerbach et al., 2017) as data input, while ML applications for redesign suggestion identification can compute event log data (Mustansir et al., 2022). In ML applications for process redesign, feature engineering is not explicitly described. ML applications for process redesign follow the supervised-learning approach as both tasks in this phase, redesign suggestion identification (e.g. Mustansir et al., 2022) and (semi-)automatic process redesign (e.g. Afflerbach et al., 2017), require guidance for model training. Concerning ML algorithms, NNs are used for detecting redesign suggestions, while (semi-)automatic process redesign is addressed using genetic algorithms. Common supervised ML metrics and KPIs (e.g. customer satisfaction) are used with publicly available event logs for model assessment.

### *6.5. Process implementation*

ML applications for resource allocation receive event logs with resource information as data input (e.g. Liu et al., 2012). Other dataset types can also be found in this phase, such as business process problems (e.g. Bae et al., 2014). Regarding feature engineering, ML applications for process implementation only selectively describe the extraction of features. However, common techniques for feature extraction cannot be observed. ML applications in this phase follow the supervised- and unsupervised-learning approach. However, there is a tendency towards the supervised-learning approach. Resource allocation planning can explain this ratio (e.g. Xu et al., 2016), which is addressed equally using supervised- and unsupervised-learning algorithms. Further, compared to other ML algorithm groups, genetic-based algorithms tend to be used more for process implementation. Resource-specific or time-related measures are calculated using mainly publicly available event logs for model assessment of resource allocation planning applications, such as r-precision (e.g. Delcoucq et al., 2022) or processing time (e.g. Xu et al., 2016).

### *6.6. Process monitoring*

While other BPM lifecycle phases work with event logs of fixed sizes, process monitoring applications receive continuous data streams in practice. Still, most of the recent process monitoring approaches are developed based on fixed-sized event logs (e.g. Brunk et al., 2021; Evermann et al., 2017; Heinrich et al., 2021). Because of their public availability for research purposes and their static properties (e.g. a static number of activities), event logs are preferred over event streams in research. These event logs are generally not limited to the control flow and include time, resource, or data-related context information. Additionally, some approaches integrate domain knowledge.

Monitoring applications rely on established techniques to extract features from event log data. Examples of such encoding techniques are the boolean, frequency, or index-based technique (e.g. Leontjeva, Conforti, Di Francescomarino, Dumas, & Maggi, 2016). Monitoring applications incorporate features that are created manually from event log data that describe certain aspects of a business process, facilitating the learning of accurate ML models. For instance, temporal features can be created based on timestamps of events, such as the time from the first event of a process instance to the current event of the same process instance (e.g. Tax et al., 2017). Another example is creating inter-case features, such as the number of resources that perform a certain activity (e.g. Senderovich et al., 2017). Feature selection is applied in this phase. For example, to learn accurate models by choosing an event log's most relevant context features (e.g. Alves et al., 2022).

Regarding model building, monitoring applications cope primarily with supervised learning, perhaps because ML is used to learn mappings from process instances or prefixes (i.e. subsequences of process instances) to targets (e.g. Maggi, Di Francescomarino, Dumas, & Ghidini, 2014). Monitoring applications adopt deep learning, ensemble learning, multi-task learning, online learning, and transfer learning. In predictive and prescriptive business process monitoring, process instances from event logs are converted into prefixes to create predictions or prescriptions as early as possible, increasing the volume of data in an event log. As deep learning profits from large data volume, ML applications in this area often use deep-learning models, such as for predicting the next activity, as Evermann et al. (2017) do. Ensemble learning is adopted for addressing all tasks of the monitoring phase using the outcome of several models, which are trained with supervised-learning algorithms (e.g. Metzger et al., 2019). The concept of multi-task learning is used when several process indicators should be monitored, leading to the use of, for instance, DNN models that are designed to approximate multiple learning targets simultaneously (e.g. Tax et al., 2017). The concept of incremental learning assumes a continuous data stream as input, so the concept is present in the process-monitoring phase. Concept-drift detection, in particular, appears with incremental learning (e.g. Baier et al., 2020). Because concept-drift detection aims to identify changes in labels or distributions in event stream data, it is a good match for these approaches. Only a few studies use transfer learning and all these studies are realised with DNNs that distinguish between representation learning and fine-tuning (e.g. Pfeiffer et al., 2021). After learning a generic representation from scratch, the resulting model can be fine-tuned for a desired task. A pre-trained model can also be fine-tuned to transfer from the general representation to the task. Instance, tree, regression, and NN-based algorithms are often employed in the process-monitoring phase because they include popular supervised ML algorithms that can be used for a variety of prediction tasks (e.g. the tree-based algorithm C4.5). Bayesian algorithms are also common in predictive business process monitoring. This is because transition probabilities are often learned with a Bayesian algorithm for a given process model and a certain prediction task before the process model is augmented with the learned probabilities. In doing so, the process model is transformed into a predictive model. Reinforcement-learning algorithms are used for a few process-monitoring tasks, such as next-best-action recommendations in prescriptive business process monitoring, as in Khan et al. (2021), or dynamic resource allocation, as in Firouzian et al. (2019a). Because of the required mapping between actions and a certain state or change of state in a system or an environment, reinforcement learning is particularly well suited to these tasks. Clustering algorithms are also used in predictive business process monitoring to train accurate predictive models per cluster, as Di Francescomarino et al. (2016) do. Additionally, some studies focus on the explanation of ML applications, as Galanti et al. (2020) do by applying SHAP for predictive business process monitoring.

For some tasks, such as predictive business process monitoring, the quality of ML models can be directly assessed. Because of the availability of ground-truth labels in event logs (e.g. next activities), established metrics from supervised ML research can be calculated. For other tasks, model assessment must consider metrics other than these standard metrics. For example, prescriptive monitoring applications take KPIs, often defined based on context features, like cost savings (e.g. Bozorgi et al., 2021) or expected throughput-time (e.g. Weinzierl, Dunzer, et al., 2020) into account for model assessment. Also, measuring time-related metrics is important in this phase as models are applied in running business processes. For example, prediction time (Zhao et al., 2016) or earliness (Teinemaa et al., 2016) are such metrics. For model assessment mainly publicly available event logs are used. Little approaches use own event logs. Further, monitoring applications use a split validation strategy more often than a cross-validation strategy compared to other BPM lifecycle phases. This preference towards split validation is due to retaining the process data's natural structure (Weytjens & Weerdt, 2021).

### 6.7. Derived findings

The results described in the previous sections are summarised in Table 3 for each BPM lifecycle phase and ML model development phase. Based on these synthesised results, we derived overarching findings.

***Finding 1: ML applications mainly address the event-data-intensive BPM lifecycle phases process discovery, analysis, and monitoring.*** A look into the number of reviewed papers shows that most ML applications address process monitoring, analysis and discovery. Due to the availability of input data, especially process data in the form of event log data, ML applications prevail in these phases. Therefore, various prediction and prescription tasks have been addressed in predictive and prescriptive business process monitoring, various pattern detection and trace clustering tasks in process analysis, and various model discovery tasks in process discovery. On the contrary, for example, it is more challenging to develop ML applications in non-event-data-intensive phases, where process identification relies on documentation, process redesign on redesign heuristics and practices, and process implementation on work plans.

***Finding 2: ML applications along the BPM lifecycle mainly use event logs with limited process context.*** Most ML applications in BPM are developed based on publicly available event logs. These event logs document a business process or its sub-types and include a limited amount of context data. However, in organisations, information systems (e.g. ERP systems) can add more context data to event logs than publicly available event logs. That is important for ML applications of BPM lifecycle phases, where context information plays a role, such as process monitoring or analysis.

***Finding 3: New forms of process data enable new use cases.*** New process data forms have led to the development of ML applications that address new BPM tasks. However, new BPM tasks can already be addressed if an event log includes a certain context attribute, such as the resource attribute for the tasks resource allocation planning in process identification or dynamic resource allocation in process monitoring. In the case of unstructured forms of data, event log creation in process identification is possible with video or text data, and process model extraction in process discovery can be addressed with text or image data. Other BPM tasks, such as (semi-)automatic process redesign, are enabled with more specific forms of data like business process designs.

***Finding 4: Integrating domain knowledge can advance ML applications.*** Domain knowledge in various forms has been integrated into ML applications to fulfil different purposes. For example, process models are integrated into prescriptive-business-process-monitoring applications to ensure the conformance of next best actions (e.g. Weinzierl, Dunzer, et al., 2020), constraints integrated into trace-clustering applications to control the clustering of traces (e.g. De Koninck et al., 2021), or logical rules integrated into predictive-business-process-monitoring applications to improve the prediction performance (e.g. Di Francescomarino et al., 2017).

***Finding 5: Common feature extraction techniques are only used for a few BPM tasks.*** Considering feature extraction techniques of ML applications across BPM tasks, we observe that ML applications apply the same feature extraction techniques for a few tasks. In contrast, specific feature extraction techniques were proposed for the other tasks. For example, ML applications for predictive business process monitoring often use the same techniques to realise feature extraction from event log data. These techniques facilitate the development of predictive-business-process-monitoring applications, as finding an appropriate technique to extract features from a given event log can be challenging. Examples of such techniques are boolean, frequency, or index-based sequence encoding (Leontjeva et al., 2016).

***Finding 6: Deep learning is the dominant ML concept in BPM.*** Deep learning is the most adopted ML concept in BPM. The main reason for this is that the architecture of DNN models can be constructed flexibly depending on the desired tasks (Goodfellow et al., 2016). For example, autoencoder models reproducing the input are constructed in process identification to create process representations (e.g. Guzzo et al., 2021). In contrast, GNN models considering event dependencies in the form of edges are constructed in process discovery to detect generalisable process models (e.g. Sommers et al., 2021).

***Finding 7: ML applications mainly focus on improving performance.*** ML applications in BPM are developed to achieve a performance improvement. A performance improvement can be a higher prediction or detection performance, a lower training or inference time, or a trade-off of various performance criteria.

***Finding 8: Explainability of ML applications is primarily addressed in process monitoring.*** Although BPM research already develops explainable ML applications based on approaches from the field of XAI,[4] these ML applications are mainly developed for predictive business process monitoring (Stierle, Brunk, et al., 2021). Specifically, when DNN models are used for predictive business process monitoring, which are perceived as black boxes, ML applications aim to make model decisions transparent for process users or other process stakeholders.

***Finding 9: Model assessment of ML applications is mainly done with data-based metrics.*** ML applications in BPM are typically assessed with data-based metrics. Common supervised ML metrics, refinements of such, or KPIs are used in BPM tasks, based on supervised learning. In contrast, model assessment is often realised indirectly for BPM tasks where no supervised learning is addressed. That is, a not-supervised-learned ML model is applied, and then clustering or process discovery metrics are calculated based on the model output. In doing that, model assessment can be data-based, even if no label information is available.

***Finding 10: Most ML applications along the BPM lifecycle use benchmark event logs.*** Our reviewed papers show that ML applications are mostly evaluated using benchmark experiments. Further, most benchmark experiments use publicly available event logs. BPM research, in particular predictive business process monitoring research, has elaborated on many ways to address prediction tasks with those event logs. Therefore, identifying accurate ML models for a business process captured in one of the publicly available event logs is consequently quite simple. When addressing a different business process, it is hard to identify promising approaches based on those benchmarks, as they only cover a selection of possible process types and characteristics. Consequently, transferring the acquired findings and insights into practice is limited.

[4] The focus of this paper is on ML, a sub-field of AI. However, as some concepts or research areas are branded with the term “AI”, we use “AI” rather than “ML”.

**Table 3**
Overview of synthesised results and derived findings.

| | Data input | Feature engineering | Model building | Model assessment |
|---|---|---|---|---|
| **Process identifica-tion** (12 paper) | • Mostly event logs with time- and resource-related context information<br>• Other dataset types for *event log creation* (e.g. raw event, video, and text data) | • For *process representation creation and learning* language models based on artificial NNs (e.g. word2vec or doc2vec) or artificial NNs with embedding layer | • Besides supervised and unsupervised learning relatively often self-supervised learning for *process representation creation and learning*<br>• Around half of the ML applications adopt deep learning<br>• Almost all ML applications use an artificial NN | • For *process representation creation and learning* and *event log creation* task clustering models (k-Means) and clustering metrics (e.g. rand index or mutual information score)<br>• Mainly publicly available event logs (e.g. BPI challenge event logs) |
| **Process discovery** (45 paper) | • Mostly event logs without context information<br>• Sometimes other dataset types for *process model extraction* (e.g. text or image data) | • For *process model extraction* language models based on artificial NNs (e.g. word2vec or BERT)<br>• For other tasks, feature engineering refers to the transformation of control-flow data into certain instance types (e.g. activity sequences or activity graphs) | • Supervised and unsupervised at about the same rate; unsupervised more common because of lack of label data (e.g. *discovery of declarative process model*)<br>• Occurrence of deep learning and transfer learning for *process model extraction* more than once<br>• Relatively often Bayesian-based, clustering-based, rule-based, and genetic-based | • Often process discovery metrics (e.g. precision, generalisation, or fitness)<br>• Often own event logs, but also publicly available logs |
| **Process analysis** (44 paper) | • Mostly event logs with time-, resource-, or data-related context<br>• Rarely other data types (e.g. text data or set of process models)<br>• Sometimes domain knowledge integrated | • Different approaches; no standards regarding feature extraction, creation, or selection | • Supervised and unsupervised at about the same rate<br>• Deep Learning (various tasks), ensemble learning (*pattern detection*), active learning (*pattern detection and trace clustering*), meta Learning (*pattern detection*), and multi-view learning (*pattern detection*) more than once<br>• Often clustering; relatively often rule-based (as in process discovery) | • Often common performance metrics from supervised ML<br>• Metrics for *trace clustering* often process discovery metrics (e.g. fitness, precision, or generalisation) or clustering metrics (e.g. cluster set entropy or mutual information score)<br>• Mainly publicly available event logs, but sometimes also own event logs |
| **Process redesign** (3 paper) | • Event logs for *redesign suggestion detection*<br>• Business process designs for *(semi-) automatic process redesign* | • Feature engineering generally not explicitly described | • Only supervised learning<br>• Genetic-based for *(semi-) automatic process redesign* and artificial-NN-based for *redesign suggestion detection* | • Common performance metrics from supervised ML and KPIs (e.g. customer satisfaction)<br>• Mainly publicly available event logs |
| **Process implementa-tion** (6 paper) | • Event logs with resource information for *resource allocation planning*<br>• Other data set types can be found (e.g. set of business process problems) | • Feature extraction sometimes described, but common patterns cannot be observed | • Only supervised and unsupervised learning with a tendency to supervised learning as *resource allocation planning* is addressed about the same rate with supervised and unsupervised learning | • Resource-specific measures (e.g. r-precision) or time-specific measures (e.g. processing time)<br>• Mainly publicly available event logs |
| **Process monitoring** (90 paper) | • Continuous data streams in theory, but in practice mostly event logs with fix size<br>• Event log often with time-related, resource-related, or data-related context<br>• Sometimes domain knowledge integrated | • Certain techniques for feature extraction from event logs (e.g. event- or sequence-encoding techniques)<br>• Manual feature creation from event log data (e.g. temporal or inter-case features)<br>• Feature selection sometimes applied (e.g. to select relevant context features) | • Mostly supervised learning<br>• Occurrence of deep learning (various tasks), ensemble learning (various tasks), multi-tasking learning (various tasks), incremental learning (various tasks), and transfer learning (*predictive business process monitoring*) more than once<br>• Often NN-based, instance-based, and tree-based, regression-based, Bayesian-based ML applications for *predictive business process monitoring*, reinforcement-learning-based for few process monitoring tasks (e.g. *prescriptive business process monitoring* or *dynamic resource allocation*), and clustering-based for *predictive business process monitoring*<br>• Explainability of ML applications is sometimes focused | • Often established performance metrics from supervised ML research (for some tasks)<br>• Often metrics defined based on context features for *prescriptive business process monitoring* (e.g. cost savings)<br>• Relatively often time-related metrics (e.g. prediction time or earliness)<br>• More often split validation than cross-validation compared to other phases<br>• Mainly publicly available event logs; seldomly own event logs |
| **Derived findings** | **F1:** ML applications mainly address the event-data-intensive BPM lifecycle phases process discovery, analysis, and monitoring<br>**F2:** ML applications along the BPM lifecycle mainly use event logs with limited process context<br>**F3:** New forms of process data enable new use cases<br>**F4:** Integrating domain knowledge can advance ML applications | **F5:** Common feature extraction techniques are used for a few BPM tasks | **F6:** Deep learning is the dominant ML concept in BPM<br>**F7:** ML applications mainly focus on improving performance<br>**F8:** Explainability of ML applications is primarily addressed in process monitoring | **F9:** Model assessment of ML applications is mainly done with data-based metrics<br>**F10:** Most ML applications along the BPM lifecycle use benchmark event logs |

**Table 4**
Future research agenda.

| Research direction | | Based on |
|---|---|---|
| **Data input** | | |
| 1 | ML applications for event-data-intensive BPM lifecycle phases (i.e. process discovery, analysis and monitoring) received much attention. Exhibiting unstructured data enables the development of ML applications for less event-data-intensive BPM lifecycle phases, where especially process data in the form of event log data are not or only to a limited extent available (i.e. process identification, redesign, and implementation). | F1 |
| 2 | Most ML applications along the BPM lifecycle mainly use event logs with limited process context. However, in real-world cases, event logs contain more information about processes; for example, machine parameters in manufacturing. To overcome that, future research should develop ML applications considering the enterprise process network and their data. | F2 |
| 3 | New forms of process data enable new use cases. As object-thinking is gaining attention in practice and academia, process data is increasingly stored as object-centric event logs. Object-centric event logs include object dependencies and thus enable new use cases. Therefore, future research should investigate the development of ML applications for object-centric event logs. | F3 |
| 4 | Business processes depend on their execution domain and integrating domain knowledge can advance ML applications. Process rules and heuristics may be known in advance. Therefore, in future research, the integration of domain knowledge into ML applications should be further investigated. | F4 |
| **Feature engineering** | | |
| 5 | Most feature extraction techniques used in ML applications for BPM are created paper for paper and common feature extraction techniques are only used for a few BPM tasks. Therefore, the development of general feature extraction techniques and the use of those is a direction for future research. For this, generative AI approaches, such as large language models, are promising. | F5 |
| **Model building** | | |
| 6 | Deep learning is the dominant ML concept in BPM research. However, beyond deep learning, there are ML concepts that have received little attention in BPM research but are promising to advance ML applications and unlock future research. For example, such an ML concept is transfer learning in the context of standard business processes. | F6 |
| 7 | ML applications strongly emphasise improving process performance, which can be unsuitable for successful use in practice. Therefore, incorporating aspects of ethical AI in into ML applications, including transparency, justice, fairness, non-maleficence, responsibility and privacy, is a promising direction for future ML research in BPM. | F7 |
| 8 | While explainability is primarily addressed in the development of ML applications for process monitoring tasks, future research should consider explainability in novel ML applications for other tasks along the BPM lifecycle. The use of XAI approaches and the symbiosis with domain knowledge are important aspects in this future research direction. | F8 |
| **Model assessment** | | |
| 9 | Model assessment in ML applications in BPM is mostly done using standard data-based metrics. Beyond data-based assessment, human-centric and economic metrics related to the BPM domain may fit the model assessment more accurately. Therefore, future research should develop such metrics for model assessment for BPM tasks (e.g. regarding efficiency and explainability of ML models). | F9 |
| 10 | Even though experiments with benchmark event logs are the de facto standard for assessing the utility of ML applications in BPM, generalisability and transfer of findings and insights can be limited. Therefore, future research should propose new approaches to support assessing the utility of ML applications. For example, promising approaches include the elaboration of methodological and technical guidance for benchmark experiments or the application of qualitative and quantitative methods to gain further insights from practitioners. | F10 |

## 7. Discussion

This section discusses future research directions we derived from our findings, implications of our literature review for research and practice, and limitations of our study.

### 7.1. Future research directions

We propose ten future research directions to advance research on ML applications in BPM, as summarised in Table 4. We identified these research directions based on the findings from our literature review and structured them along the phases of the ML model development process.

***Research direction 1:*** According to Finding 1, ML applications in BPM mainly address tasks in the event-data-intensive BPM lifecycle phases of process discovery, analysis and monitoring, where process data in the form of event log data is typically available. Therefore, we propose as future research direction the examination of other data sources from different enterprise systems and the investigation of approaches to induce structure into unstructured and semi-structured data. Consequently, future research can develop novel ML applications for the less event-data-intensive BPM lifecycle phases (i.e. process identification, redesign, and implementation).

To make unstructured process data in these lifecycle phases usable, techniques from the natural language processing (NLP) (e.g. Otter, Medina, & Kalita, 2020) or computer vision (e.g. Voulodimos, Doulamis, Doulamis, & Protopapadakis, 2018) field, which are deeply interconnected with ML approaches, could facilitate the development of novel ML applications. For example, setting up organisational business process landscapes in process identification with NLP techniques based on textual documentation (e.g. van der Aa, Carmona Vargas, Leopold, Mendling, & Padró, 2018), suggesting and testing redesign opportunities in process redesign with computer vision techniques based on video recordings (e.g. Mustansir et al., 2022), and supporting the implementation of business processes in suitable information systems with NLP techniques based on text-based work plans (e.g. Xu et al., 2016).

***Research direction 2:*** In accordance with Finding 2, ML applications along the BPM lifecycle mainly use event logs with limited process context. Therefore, we propose as future research direction the development of novel ML applications, which consider the enterprise process network instead of isolated business processes (Oberdorf et al., 2023). In doing that, ML applications receive input data from multiple data sources, including control-flow information from different business processes and process context information related to the business processes. As more context is considered in such ML applications, the selection of valuable context information via feature selection increases in importance. Finally, as business processes can span multiple organisations, process networks are not limited to one organisation. ML applications could thus span across multiple organisations, providing insights and recommendations for supply chain processes.

***Research direction 3:*** Based on new forms of process data (Finding 1), including object-centric event logs, future research should explore further use cases to develop novel ML applications. Currently, both process-mining vendors and academia are adjusting to object-thinking in BPM (e.g. Li, de Murillas, de Carvalho, & van der Aalst, 2018), which originated in artefact-centric workflow modelling (Nigam & Caswell, 2003). Consequently, process data are more and more stored as object-centric event logs. "Object-centric" is a process paradigm, after which an instance of a business process is not performed in isolation like in the process-instance-centric process paradigm, but interacts via objects with other instances of business processes (van der Aalst, 2019). According to the object-centric process paradigm, an object-centric event log stores object dependencies, which are neglected in traditional event logs (van der Aalst, 2016). Therefore, object-centric event logs reveal new use cases, such as predicting remaining object interactions in the next 24 h, detecting incorrect object interactions, or repairing missing object interactions in object-centric event log data.

***Research direction 4:*** Derived from Finding 4, integrating domain knowledge into model building can advance ML applications. Therefore, the integration of domain knowledge into ML applications should be further investigated. For example, this includes (i) the integration of multiple entities of domain knowledge, (ii) the amount and form of integrated domain knowledge required for solving certain learning problems, and (iii) the integration of profound domain knowledge from BPM.

Concerning (i), each entity can express its domain knowledge as "logical rules, constraints, mathematical equations [...], probability distributions, similarity measures, knowledge graphs or ontologies" (Folino & Pontieri, 2021, p. 86). These entities can also be related to each other in diverse ways or be structured into various areas or levels. Predicting future process behaviour could benefit from integrating domain knowledge as employee-related rules and a company-related ontology into ML applications. Concerning (ii), ML research contains work that proposes approaches that allow controlling which domain knowledge should be integrated into a specific situation into an ML algorithm. For example, Maier et al. (2019) propose operators to incorporate prior knowledge into ML algorithms in a controlled manner. The development of future ML applications could include control mechanisms to handle intended exceptional process flows that, for example, mitigate risks. Concerning (iii), profound domain knowledge from the BPM domain could be present as redesign heuristics, or topical and organisational constraints (e.g. Dumas et al., 2018). The integration of BPM knowledge could aid in automatically suggesting improved process models and definitions.

***Research direction 5:*** According to Finding 5, common feature extraction techniques are only used for a few BPM tasks. Therefore, future research should propose general feature extraction techniques for BPM. Particularly interesting are tasks in BPM lifecycle phases for which various specific techniques have been proposed (e.g. pattern detection in process analysis). The common feature extraction techniques should also consider the extraction of features from dataset types apart from event logs (e.g. text or video) (Kratsch et al., 2022).

To realise common feature extraction techniques, the use of language models seems to be a promising direction. In our reviewed papers, we observed that (large) language models (e.g. word2vec and BERT) are increasingly used in ML applications either as primary components to solve BPM tasks (e.g. process representation creation and learning (e.g. De Koninck et al., 2018) or process model extraction (e.g. Qian et al., 2020)) or as secondary components to support other BPM tasks through learning precise representations of event log or text data (e.g. predictive business process monitoring (e.g. Teinemaa et al., 2016) or pattern detection (e.g. Junior et al., 2020)).

When pre-trained large language models (e.g. the generative pre-trained transformer 4 (GPT-4) (OpenAI, 2023)) are fine-tuned and evolve from the pure ML model to a system (e.g. a conversational agent like ChatGPT (Guo et al., 2023)) or application (e.g. content generation like search engine optimisation) (Feuerriegel, Hartmann, Janiesch, & Zschech, 2023), they open up new possibilities for ML-based BPM research (Vidgof, Bachhofner, & Mendling, 2023). For example, for predictive business process monitoring, conversational agents can be used during process execution, and process users can ask questions to these agents about different aspects of a running process instance. Another example refers to redesign suggestion identification, for which conversational agents can be asked to reveal innovation opportunities for process improvement. Moreover, for process implementation, large-language-model-based applications solving the problem of routine task automation are a new opportunity for ML-based BPM research. Finally, as large language models refer to a type of model used in the context of generative AI (Feuerriegel et al., 2023), other model types, such as diffusion probabilistic models, GANs models, or variational autoencoder models, are promising for developing novel ML applications in BPM. Additionally, models considering data modalities beyond text (e.g. image, audio, and code) or combinations of them (e.g. image-to-text or audio-to-text) are promising for future BPM research on ML applications.

***Research direction 6:*** In accordance with Finding 6, deep learning is the dominant ML concept in BPM. However, beyond deep learning, we propose as future research direction the development of novel ML applications considering ML concepts that have received little attention in BPM research but are promising to advance ML applications. This includes in particular the ML concepts transfer learning, federated learning, causal learning, and neuro-symbolic AI.

The concept of transfer learning (e.g. Pan & Yang, 2009) is promising for BPM research as BPM software provides reference models for many standard business processes (e.g. order to cash) (van Dongen, Jansen-Vullers, Verbeek, & van der Aalst, 2007), and such standard processes are similar in many organisations and organisational units. For example, in such a homogeneous scenario, a model for predicting remaining time could be trained in an organisational unit before this unit could pass the pre-trained model on to other units that do not have enough process data to train models themselves. First works in BPM research make use of transfer learning to address multiple prediction targets given an event log for comprehensive predictive business process monitoring (e.g. Pfeiffer et al., 2021) or to integrate additional context information for accurate discovery of declarative process models (e.g. López et al., 2021).

The concept of federated learning (e.g. Yang et al., 2019) is worth investigating for BPM research as processes can be distributed and go beyond the boundaries of one organisation. For such business processes, federated learning provides promising approaches to training shared ML models using data from multiple owners – while keeping all training data local and private – to train powerful ML models. Multi-organisational process optimisation could also be addressed using federated learning, and small companies with only a small amount of event data could federate to learn predictive models for common processes while keeping their data private.

The concept of causal learning (or causal ML) is promising for BPM research as it aids in better understanding business process improvement by formalising the data-generation process as a structural causal model and enabling to reason about the effects of changes to this process (intervention) and what would happened in hindsight (counterfactuals) (Kaddour, Lynch, Liu, Kusner, & Silva, 2022). Some research in process analysis (e.g. Bozorgi et al., 2020) and process monitoring (e.g. Bozorgi et al., 2021) already demonstrates the benefit of using causal supervised ML. However, the high number of additional ideas in causal ML, including causal generative modelling, causal explanations, causal fairness, and causal reinforcement learning (Kaddour et al., 2022), provide promising avenues for future research.

The concept of neuro-symbolic AI (or neuro-symbolic computation) is worth investigating for BPM research as it provides new principles, concepts, and methods for integrating domain knowledge into ML applications. In fact, neuro-symbolic AI emerged in AI research to overcome the challenge of integrating learning and reasoning (Garcez et al., 2015). For this purpose, neuro-symbolic AI combines robust connectionist machines (e.g. neural networks) with sound, logical abstractions (e.g. logical rules) (Garcez et al., 2022). For example, Pasquadibisceglie, Castellano, et al. (2021) do so as one of the first with a neuro-fuzzy model in predictive business process monitoring.

***Research direction 7:*** According to Finding 7, ML applications mainly focus on improving performance. However, a pure consideration of performance can be insufficient for successful use in practice, and future BPM research should address the development of novel ML applications that incorporate aspects of ethical AI. The field of AI ethics, often called trustworthy AI, has emerged in response to growing concerns about the impact of ML applications among other issues (Kazim & Koshiyama, 2021). AI ethics comprises such aspects as transparency, justice, fairness, non-maleficence, responsibility, and privacy (Feuerriegel, Dolata, & Schwabe, 2020; Jobin, Ienca, & Vayena, 2019; van der Aalst, Bichler, & Heinzl, 2017), and its application regularises the impact of ML applications.

However, despite AI ethics' importance for ML applications on the individual level, BPM research has discussed AI ethics only in theory (Mendling et al., 2018). Further, BPM research addresses fairness-aware process mining (Qafari & van der Aalst, 2019) and demonstrates how process mining can be used to analyse and ensure AI's ethical compliance (Pery, Rafiei, Simon, & van der Aalst, 2022), but those approaches consider fairness and ethical compliance from a process-instance perspective in specific situations. Aspects of AI ethics on the individual level are not yet incorporated in developing ML applications for BPM.

One way to address this gap could be to consider fairness in developing an ML application predicting process outcomes, which can be robust to biases. For example, a credit application might be rejected because of discrimination in a loan application process. Another example refers to privacy, which should be considered in developing an ML application for detecting deviations. Deviations may present an incorrect behaviour of a specific worker to every process analyst without restrictions.

***Research direction 8:*** In accordance with Finding 8, explainability is primarily addressed in the development of ML applications for process monitoring tasks. As explainability is useful and not restricted to process monitoring tasks, we propose as a future research direction the consideration of it in the development of novel ML applications for other tasks along the BPM lifecycle.

For example, trace clustering, to determine why certain traces are assigned to clusters. De Koninck, De Weerdt, et al. (2017) already propose one of the first approaches in this direction. Following Ko and Comuzzi (2023), explainability is also relevant for anomaly detection, as explanations aid in understanding why a detected instance is an anomaly. Organisations can also be interested to understand why an ML model for (semi-)automatic redesign decides that a process should change. As an additional point of this research direction, we subscribe to Bauer, Hinz, van der Aalst, and Weinhardt (2021)'s recommendation of merging XAI with domain knowledge using human-in-the-loop approaches (Wu et al., 2022; Zanzotto, 2019), as first studies in the BPM domain demonstrate (e.g. Junior et al., 2018).

***Research direction 9:*** According to Finding 9, model assessment of ML applications is mainly done with data-based metrics. Therefore, we propose as future research direction the development of additional human-centric and economic metrics that measure BPM-specific aspects of ML applications, that are relevant for process stakeholders but cannot be directly obtained from the underlying data.

One of these aspects is the effectiveness of ML applications, that is, how useful the output of an ML application is for process users and other process stakeholders. For example, in prescriptive business process monitoring, some approaches recommend the next best actions and report KPI improvements (e.g. cost or time savings). However, these approaches do not consider how useful the recommended actions are for process users or other stakeholders (e.g. Weinzierl, Zilker, Stierle, Matzner, & Park, 2020).

Another aspect is the explainability of ML applications, that is, how explainable ML applications are for process users or other stakeholders. For predictive business process monitoring, where explainability of ML applications is mostly addressed in BPM research, the quality of explanation is often assessed via a demonstration of created explanations without any reference to process users or other stakeholders (Stierle, Brunk, et al., 2021). However, the focus of the explanations should be on the humans who execute the processes rather than those who develop them. This focus is important as developers may lack the knowledge to understand such explanations (Bauer et al., 2021). Therefore, metrics are required to measure the explainability of ML applications for process users or other stakeholders.

***Research direction 10:*** According to Finding 10, most ML applications along the BPM lifecycle use benchmark event logs for assessing the utility. However, as generalisability and transfer of findings and insights can be limited using benchmark event logs, future research should investigate new approaches to support assessing the utility of ML applications in BPM.

One approach could be elaborating methodological and technical guidance for benchmark experiments, including event logs, which represent different process types and characteristics. While there is such guidance at pre-print stages for general ML benchmarks, including a framework and data sets (e.g. Romano et al., 2021), BPM requires such methodological approaches for developing ML applications.

Another approach could be assessing the utility of ML applications qualitatively or quantitatively. Involving practitioners in early-stage problem formulation and in the evaluation of an ML application's utility would improve both model assessment and transfer to practice. For example, the utility of an ML application for an organisation can be evaluated in case studies, like Stierle, Weinzierl, et al. (2021) do in the context of process analysis.

### 7.2. Implications

Our literature review has implications for both research and practice. For research, our literature review provides a knowledge base of ML applications structured into BPM lifecycle phases and BPM tasks. The phases and tasks of the knowledge base are described and defined in the view of ML applications. With this knowledge base, we structure the discipline of ML applications in BPM and show how, where, and when ML applications can be used in BPM. While researchers from the BPM domain can use and extend the knowledge base for their research purposes, researchers new to the field can use it as an entry point. Finally, with an interactive coding table, we give researchers a tool that supports them in analysing ML applications in BPM according to concepts, which are structured into the ML model development process phases.

We stimulate more ML research outside the predominant BPM lifecycle phases. In doing that, we contribute to ML applications developed and used in the field of BPM. Nevertheless, this review covers ML applications from all BPM lifecycle phases, as we found that a comparison of ML applications across all lifecycle phases is essential, as the obtained insights from predominant phases can be transferred to less investigated phases.

With our findings and research directions addressing the utility of ML applications, we propose shifting the focus from prediction-performance-oriented ML research in BPM to more business-value-oriented ML research. In doing that, we aim to improve the transfer of developed ML applications, concepts, or ideas from research into practice.

Our literature review also has implications for practice. We show practitioners various ways and potentials to develop and use ML applications to improve business processes and ultimately create value. For example, ML can be used in process identification to create event logs, which are then analysed with process mining applications to detect bottlenecks in business processes. As another example, ML can be used in process monitoring to predict the outcome of running business processes. This enables process users to intervene in running business processes if the predicted process outcomes cause problems. These examples illustrate the benefits, which ML applications from BPM research can have for practice.

Moreover, we recorded the existence of implementations of ML applications in our coding table. Therefore, practitioners can use our coding table to find the source code of an ML application that addresses a desired BPM task. As the source code from a paper can be used as a basis for implementation, practitioners can build on existing knowledge.

### *7.3. Limitations*

Like all studies, this research project has limitations. First, as with any literature review, a natural limitation is the investigated time period. Thus, derived findings may change over time and new findings can evolve. To omit this issue, we provide the interactive coding table alongside this publication. The content can be updated periodically and new findings may be derived.

Second, the number of BPM-related and ML-related keywords in our search string is unbalanced, as many papers do not use the keyword "machine learning" itself but the name of a specific ML algorithm or ML concept. Including more specific BPM-related keywords (e.g. the BPM lifecycle phase process monitoring) resulted in many research papers retrieved.

Third, because of the high number of papers examined in this review, the papers were divided among three researchers for reading and coding. By applying an iterative approach with discussions, as described in Section 4, we tried to ensure that the researchers had a common understanding of the concepts to be coded. Our inter-coder reliability analysis results (Appendix B) confirm that the researchers used a common understanding in coding the concepts. Still, we cannot guarantee that the coding table is free of inconsistencies.

Fourth, we suppose that some papers do not report all of the concepts we are interested in even when those concepts are part of the underlying ML application.

Fifth, assigning each paper to a BPM lifecycle phase was not trivial. Identifying a clear assignment to a BPM lifecycle phase was challenging when the authors of a paper did not mention a particular phase. In addition, a few papers address more than one phase, and we assigned those papers based on the papers' primary focus.

Finally, we used the ML model development process as a basis on which to structure ML applications, but some ML applications use more than one ML model, which we considered in our coding. However, the composition of ML models in or across ML applications goes beyond the scope of this review.

## CRediT authorship contribution statement

**Sven Weinzierl:** Conceptualization, Data curation, Investigation, Writing – original draft, Visualization. **Sandra Zilker:** Conceptualization, Data curation, Investigation, Writing – original draft, Visualization. **Sebastian Dunzer:** Conceptualization, Data curation, Writing – original draft, Visualization. **Martin Matzner:** Supervision, Funding acquisition.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

The developed coding table is included in the paper.

## Acknowledgements

We want to thank Ludwig Grobmeyer for this support in the eary stages of this research project.

## Appendix A. Search process specifications

The search process consisted of three steps: First, we developed a *search string* (Appendix A.1). Then, we conducted the collection of papers based on different search activities depending on the used databases Appendix A.2. Lastly, we derived the final set of papers in the (*assessment phase*), applying various *quality and exclusion criteria* (Appendix A.3).

### *A.1. Search string development*

The search string was constructed from two perspectives: the *BPM perspective* and the *ML perspective*.

To cover the *BPM perspective*, we used the search terms *"business process management"* and *"process mining"* (van der Aalst, 2016), as well as *"workflow management"* and *"workflow mining"* for a general view. Then, we used the phases of the BPM lifecycle by Dumas et al. (2018), combined with the term *"business"* to strictly consider business processes (e.g. *"business process monitoring"* or *"business process discovery"*). We also included related concepts like *"process intelligence"* (van der Aalst, 2016). Lastly, we combined the keywords *"data-driven"* and *"business process"* to include publications that address business processes from a data-driven perspective.

For creating the ML perspective of our search string, we first screened fundamental works on ML to get a comprehensive overview of ML approaches. However, none of the investigated works provided an overview with a suitable degree of detail for our work. Consequently, we set out to develop a new overview following fundamental works. First, we defined levels of the overview, that is, ML paradigm, ML concept, and ML algorithm group following Janiesch et al. (2021). Second, we initially defined the categories for the level *ML paradigm* following Mohri, Rostamizadeh, and Talwalkar (2018) and refined these based on further works (see listed works under ML paradigm in Table 1). Third, we initially defined the categories for the level *ML concept* following Goodfellow et al. (2016), Murphy (2012) and Mohri et al. (2018) and refined these based on further works (see listed works under ML concept in Table 1). Fourth, we initially defined the categories for the level *ML algorithm group* and the corresponding representatives for each category following Mitchell (1997), Bishop (2006) and Mohri et al. (2018) and refined these based on further works (see listed works under ML algorithm group in Table 1).

Further, concerning the ML perspective of our search string, we included, where applicable keywords using alternate spellings (e.g. *"semisupervised learning"*, *"semi-supervised learning"*) or keywords and their abbreviations (e.g. *"support vector machine"* and *"SVM"*). For most keywords, we used the keyword as a fixed term, but for some, we used the word stem to cover a broader spectrum of papers (e.g. "regress*" to find results for *"regression"* and *"regressor"* among other).

We combined the search strings using Boolean operators. The keywords were combined with "OR" within one perspective, whereas two perspectives were combined with "AND" to ensure that both ML and BPM are covered in the results. The search was conducted for the title of the publication, its abstract, and its keywords.

**Table A.5**
Quality and exclusion criteria applied in the search process.

| ID | Criterion |
|---|---|
| Q1 | The paper meets a length requirement of four pages to ensure a profound contribution. |
| Q2 | The paper is accessible to enable further screening and potential coding. |
| Q3 | The paper has been published at a conference or journal (except predatory journals). |
| Q4 | The paper meets the citation of at least two citations per year on average for papers published in 2021 or earlier, while younger papers were included regardlessly. |
| E1 | The paper does not treat a business process, e.g. social network analyses. |
| E2 | The paper presents an ML application for addressing a domain-specific problem, e.g. the prediction of a certain disease in the healthcare domain. |
| E3 | The paper only considers optimisation algorithms (i.e. solvers) without a reference to optimising parameters of an ML model, e.g. linear programming. |
| E4 | The paper uses classical AI approaches, e.g. logic programming. |
| E5 | The paper uses a clustering algorithm without an automated improvement procedure, e.g. a rule-based clustering approach for process model discovery. |
| E6 | The paper uses ML solely for an evaluation purpose, e.g. linear regression used in a quantitative study. |
| E7 | The paper does not provide a novel approach, but has a comparative nature, e.g. comparison of different approaches or is a literature review. |

### A.2. Conducted search activities per database

To conduct the search and retrieve the relevant documents, three different databases including a wide range of academic publications were used.

**Scopus** For the identification of all relevant records in Scopus[5], the search string as depicted in Fig. 3 was used in the "Advanced document search". After the documents were found, we refined the search by only including records in English. The results were then exported as a csv file including all bibliometric information.

**IEEE Xplore** For the database IEEE Xplore[6], the search string in Fig. 3 was used with the "command search" function where only the data fields were updated to fit the syntax. As there is no possibility to search for a paper's title, abstract, and keywords in one search, three individual searches were conducted. Once all records were found for one of the three searches, the items per page was set to the maximum of 100 records. All results per page were then selected using the "select all on page" function. The results and the bibliometric information were exported as a csv file. The same was done for the remaining pages until all records were exported. The search procedure was then repeated for all three data fields (title, abstract, and keywords). The individual csv files were consolidated in one file and all duplicates were removed.

**Web of Science** For Web of Science[7], the "advanced search" was used for the search string shown in Fig. 3. To search for title, abstract and keywords, the "topic" field was applied. After the search was conducted, the records were filtered to only include English publications. The results were then exported with the full records as an Excel file.

All individual files retrieved from the three databases were then consolidated using MS Excel to all follow the same structure. Lastly, a unique identifier was assigned to each entry of the consolidated list. This list was also used to filter duplicates and to keep track of the papers during the screening phases.

[5] https://www.scopus.com/search/form.uri?display=advanced
[6] https://ieeexplore.ieee.org/search/advanced/command
[7] https://www.webofscience.com/wos/woscc/advanced-search

### A.3. Quality and exclusion criteria

In line with Okoli (2015) and vom Brocke et al. (2015), we created quality and exclusion criteria for the full-text assessment of the papers, summarised in Table A.5.

## Appendix B. Inter-coder reliability analysis

Table B.6 shows the results of the inter-coder reliability analysis for all binary and categorical dimensions. The dimensions that required free text were all checked by one coder at the end of the coding phase to ensure consistency, for which we used a random sample of 25 papers (approximately 12.5% of the set of papers). The coding was done independently by the three coders based on the concept matrix that we developed during the coding process.

We used three types of metrics to measure the inter-coder reliability. First, we calculated the percentage agreement representing the relative number of papers, for which all coders set the same code (Lombard et al., 2006). We also calculated Krippendorff's $\alpha$, which is appropriate for two or more coders and can deal with diverse types of data and missing values (Krippendorff, 2018). Lastly, we calculated Fleiss' $\kappa$ (Fleiss & Cohen, 1973), which is a common metric for three or more coders. We used R (version 4.2.1) with the *DescTools* library for the calculation.

The percentage agreement among the coders is high for all dimensions and concepts (greater than 0.8); only one dimension (*Number of data sets*) is below 0.8, so a single coder coded all papers for this dimension again to ensure consistency in the coding.

The percentage agreement does not account for agreements by chance (Lombard et al., 2006), so we calculated Krippendorff's $\alpha$ and Fleiss' $\kappa$, following Neuendorf (2017). Both metrics correct for the probability of agreement by chance (Landis & Koch, 1977). These metrics reveal that almost all concepts are within a very good range, as most values are above 0.8, which resembles a near-perfect agreement (Landis & Koch, 1977), or are between 0.61 and 0.8, resembling substantial agreement (Landis & Koch, 1977).

We observed divergent behaviour for only three concepts: The values for *Feature creation*, *Self-supervised learning*, and *Online/incremental learning* are between 0.45 and 0.5. The results for *Rule-based* and *Other* in the algorithm group section, are not representative due to strongly imbalanced data. However, to ensure the reliability of our coding, one of the researchers checked at the end again the complete coding.

**Table B.6**
Inter-coder reliability analysis results.

| Dimension | | % Agreement | Krippendorff's $\alpha$ | Fleiss' $\kappa$ |
|---|---|---|---|---|
| Overall | | **0.954** | | |
| Evaluation | **Overall** | **0.867** | | |
| | Reference | 0.958 | 0.785 | 0.787 |
| | Data type | 0.917 | 0.884 | 0.883 |
| | Number of data sets | 0.625 | 0.686 | 0.684 |
| | Implementation available | 0.833 | 0.774 | 0.773 |
| BPM lifecycle phase | **Overall** | **0.972** | | |
| | Process identification | 1.000 | 1.000 | 1.000 |
| | Process discovery | 0.917 | 0.840 | 0.839 |
| | Process analysis | 0.958 | 0.907 | 0.906 |
| | Process redesign | 1.000 | 1.000 | 1.000 |
| | Process implementation | 1.000 | 1.000 | 1.000 |
| | Process monitoring | 0.958 | 0.944 | 0.944 |
| Data input | **Overall** | **0.958** | | |
| | Type of dataset | 0.958 | 0.740 | 0.738 |
| Feature engineering | **Overall** | **0.900** | | |
| | Feature creation | 0.875 | 0.454 | 0.454 |
| | Feature selection | 0.917 | 0.639 | 0.636 |
| ML paradigm | **Overall** | **0.937** | | |
| | Supervised learning | 0.833 | 0.761 | 0.759 |
| | Semi-supervised learning | 1.000 | 1.000 | 1.000 |
| | Reinforcement learning | 1.000 | 1.000 | 1.000 |
| | Self-supervised learning | 0.917 | 0.474 | 0.470 |
| | Unsupervised learning | 0.875 | 0.814 | 0.813 |
| | Other | 1.000 | 1.000 | 1.000 |
| ML concept | **Overall** | **0.983** | | |
| | Multi-task learning | 1.000 | 1.000 | 1.000 |
| | Active learning | 1.000 | 1.000 | 1.000 |
| | Online/incremental learning | 0.958 | 0.489 | 0.486 |
| | Transfer learning | 1.000 | 1.000 | 1.000 |
| | Ensemble learning | 0.958 | 0.737 | 0.735 |
| | Deep learning | 0.958 | 0.937 | 0.936 |
| | Meta learning | 1.000 | 1.000 | 1.000 |
| | Federated learning | 1.000 | 1.000 | 1.000 |
| | Multi-view learning | 1.000 | 1.000 | 1.000 |
| | Other | 0.958 | 1.000 | 1.000 |
| ML algorithm group | **Overall** | **0.954** | | |
| | Instance-based | 1.000 | 1.000 | 1.000 |
| | Decision tree-based | 0.917 | 0.840 | 0.839 |
| | Clustering-based | 1.000 | 1.000 | 1.000 |
| | Artificial-neural-network-based | 0.875 | 0.830 | 0.829 |
| | Regression-based | 0.958 | 0.737 | 0.735 |
| | Bayesian-based | 0.917 | 0.840 | 0.839 |
| | Rule-based[a] | 0.958 | NA | NA |
| | Reinforcement learning-based | 1.000 | 1.000 | 1.000 |
| | Genetic-based | 1.000 | 1.000 | 1.000 |
| | Other[a] | 0.917 | NA | NA |
| Model assessment | **Overall** | **0.833** | | |
| | Validation strategy | 0.833 | 0.822 | 0.821 |

[a] Only the percentage agreement is reported due to strongly imbalanced data.

## Appendix C. Coding table

The coding table is shown in Table C.7.

**Table C.7**
Coding table for ML applications in BPM.

| Paper | Evaluation | | | | | BPM lifecycle phase | | | | | | | BPM task category | Data input | | Feature engineering | | | Model building | | | | | | | | | | | | | | | | | | | | | | | | | | Model assessment | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | | | | ML paradigm | | | | | | ML concept | | | | | | | | | | ML algorithm group | | | | | | | | | | | |
| | Data type | Number of datasets | Name of datasets | Statistical method | Further | Implementation available | Process identification | Process discovery | Process analysis | Process redesign | Process implementation | Process monitoring | BPM task category | Dataset type | Rel. context information | Feature extraction | Feature creation | Feature selection | Supervised learning | Semi-supervised learning | Reinforcement learning | Self-supervised learning | Unsupervised learning | Other | Multi-task learning | Active learning | Online/increm. learning | Transfer learning | Ensemble learning | Deep learning | Meta learning | Federated learning | Multi-view learning | Other | Instance-based | Decision tree-based | Clustering-based | Artificial-NN-based | Regression-based | Bayesian-based | Rule-based | Reinf.-learn.-based | Genetic-based | Other | Metrics | Validation strategy |
| Afflerbach et al. (2017) | Real-world, Synthetic | 2 | Travel agent process, synthetic logs | | |||||x||| (Semi-)automatic process redesign | Business process designs (activities, connections, routing decisions) | | 5 matrices (activity-attribute, object-attribute, the activity-input, activity-output, activity-process-attribute) |||x||||||||||||||||x|||||||||x|| Fitness | |
| Alves et al. (2022) | Real-world | 3 | Incident, BPIC2013, WFM | | |x||||||x| Predictive business process monitoring | Event log | time | one-hot |x|x|x|||||||||||x||||||||x||||||| MAE | split |
| Appice et al. (2019) | Real-world | 2 | Helpdesk, BPIC2012 | | |||||||x| Predictive business process monitoring | Event log | time | index-based, window-based, frequency of activities |x||x||||||||||x||||||x|x|||x|||||| MAE, Accuracy | split, cross |
| Appice and Malerba (2015) | Real-world, Synthetic | 8 | BPIC2013, etm, hospital, isbpm, photo, review, repair, claims | ANOVA | ||||x||| Trace clustering | Event log | | frequency-based |||||||x||||||||||x||||x|||||||| Mean of silhouette width, Computation time, Number of learning iterations, Fitness, Complexity of model | |
| Bae et al. (2014) | Synthetic | 5 | Randomly generated processes | | ||||||x|| Execution planning | Set of BP problems | | sequence priority, resource assignment |||x||||||||||||||||x|||||||||x|| Fitness | |
| Baier et al. (2020) | Real-world | 1 | P2P | | |||||||x| Predictive business process monitoring | Event log | | one-hot ||x|x||||||||x||||||||x|x||x||x||||| Accuracy | split |
| Junior et al. (2018) | Real-world | 1 | Hospital Billing | | |x||||||x| Descriptive business process monitoring | Event log | time | edit weighted distance, time-weighted distance, global time |x||||||x||||x||||||||||x|||||||| | |
| Bazhenova et al. (2016) | Real-world | 1 | Bank system | | |x||x||||| Decision mining | Event log | data | |||x||||||||||||||||x||x||||||||| | |
| Berkenstadt et al. (2020) | Real-world, Synthetic | 2 | DFCI, MM1 | | |x|||x||| Performance analysis | Event log | | |||x||||||||||||||||||||x|||||| RMSE, MAE, MARE | split |
| Bernard and Andritsos (2019) | Real-world, Synthetic | 4 | Helpdesk, BPIC2012, BPIC2013, envPermit | | |x||||||x| Predictive business process monitoring | Event log | | |||x||||x|||||||||||x|x||x|||||||| Damerau similarity, Average execution time, Accuracy | split |
| Bevacqua et al. (2013) | Real-world | 1 | Transshipment process | | |||||||x| Predictive business process monitoring | Event log | time, data | bag, set encoding |x||x|||||||||||||||x|x|x|x||x||x|||| RMSE, MAE, MAPE | cross |
| Boltenhagen et al. (2019) | Synthetic | 3 | Synthetic event logs | | |x|||x||| Trace clustering | Event log | | pseudo-Boolean formula |||||||x|||||||||||x|||x|||||||| Execution time, Maximum distance between a trace and the centroid of its cluster, Number of clusters, Ratio of clustered traces, Inter-cluster distance | |
| Borkowski et al. (2019) | Real-world, Synthetic | 4 | Bosch, Cargo 2000, BPIC2017, CFS | | |||||||x| Predictive business process monitoring | Event log | data | |||x||||||||x|||x||||||||x||||||| Precision, Recall, Matthews correlation coefficient | split, cross |
| Bose and van der Aalst (2009) | Real-world | 1 | Telephone repair process | | ||||x||| Trace clustering | Event log | | depending on clustering technique |||||||x|||||||||||x|||x|||||||| Fitness, Avg weighted fitness, Complexity | |
| Bozorgi et al. (2020) | Real-world | 1 | BPIC2017 | | |x|||x||| Pattern detection | Event log | | |||x||||||||||||||||x||x||||||||| Kullback–Leibler divergence, Action rules | |
| Bozorgi et al. (2021) | Real-world | 2 | Own | | |x||||||x| Prescriptive business process monitoring | Event log | time, re-source, data, inter-case | aggregation encoding (activity, resource), last-state (data), one-hot encoding |x||x|||||||||||||||x||x||||||||| Gini, Net value | split |

(continued on next page)

**Table C.7** (*continued*).

| Paper | Evaluation | | | | | | BPM lifecycle phase | | | | | | BPM task category | Data input | | Feature engineering | | | Model building | | | | | | | | | | | | | | | | | | | | | | | | | | Model assessment | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | | | | | ML paradigm | | | | | ML concept | | | | | | | | | | ML algorithm group | | | | | | | | | | | |
| | Data type | Number of datasets | Name of datasets | Statistical method | Further | Implementation available | Process identification | Process discovery | Process analysis | Process redesign | Process implementation | Process monitoring | BPM task category | Dataset type | Rel. context information | Feature extraction | Feature creation | Feature selection | Supervised learning | Semi-supervised learning | Reinforcement learning | Self-supervised learning | Unsupervised learning | Other | Multi-task learning | Active learning | Online/increm. learning | Transfer learning | Ensemble learning | Deep learning | Meta learning | Federated learning | Multi-view learning | Other | Instance-based | Decision tree-based | Clustering-based | Artificial-NN-based | Regression-based | Bayesian-based | Rule-based | Reinf.-learn.-based | Genetic-based | Other | Metrics | Validation strategy |
| Breuker et al. (2016) | Real-world, Synthetic | 2 | BPIC2012, BPIC2013 | Kruskal–Wallis test | | x | | | | | | x | Predictive business process monitoring | Event log | | n-gram | | | | | | | x | | | | | | | | | | | x | | | x | | | x | | | | | Cross entropy, Accuracy, TP, TN | split |
| Brunk et al. (2021) | Real-world | 2 | BPIC2012, BPIC2013 | | | x | | | | | | x | Predictive business process monitoring | Event log | re-source, data | | | | | | | | x | | | | | | | | | | | x | | | x | | | x | | | | | Accuracy, F1-Score, Mean, StDev, AD | cross |
| Buijs et al. (2013) | Real-world, Synthetic | 9 | CoSeLoG, loan application process | | | | | x | | | | | Discovery of procedural process model | Collection of event logs | | | | | x | | | | | | | | | | | | | | | x | | | | | | | | | x | | Fitness, Precision, Simplicity, Generalisation, Similarity, Size, Number of configuration points | |
| Buijs et al. (2012) | Synthetic | 3 | al2, al2All5pcNoise, HerbstFig6p34 | | | | | x | | | | | Discovery of procedural process model | Event log | | | | | x | | | | | | | | | | | | | | | x | | | | | | | | | x | | Overall fitness, Replay fitness, Precision, Simplicity, Generalisation | |
| Camargo et al. (2019) | Real-world | 4 | BPIC2012, BPIC2013, BPIC2015, Helpdesk | | | x | | | | | | x | Predictive business process monitoring | Event log | time, re-source | embedding, normalisation | x | | x | | | | | | x | | | | | x | | | | | | | | x | | | | | | | Accuracy, Damerau–Levenshtein distance, MAE | split |
| Camargo et al. (2022) | Real-world, Synthetic | 9 | BPIC2017 W, BPIC2012 W, INS, ACR, MP, CVS, CFM, CFS, P2P | | | x | | | x | | | | Process simulation | Event log | time, re-source, data, inter-case | embedding | x | | x | | | | | | | | | | | x | | | | | | | | x | | | | | | | MAE, EMD, RMSE, SMAPE | split |
| Ceravolo et al. (2022) | Real-world, Synthetic | 271 | Helpdesk, synthetic logs | | | x | | | | | | x | Descriptive business process monitoring | Event log | time, cost, re-source | graph-based | | | | | | | x | | | | x | | | | | | | | | | x | | | | | | | | Graph Distance, RMSLE, F-Score, Standard deviation | |
| Chesani et al. (2009) | Synthetic | NA | Synthetic event logs | | | | | x | | | | | Discovery of declarative process model | Event log | data | | | | x | | | | | | | | | | | | | | | x | | | | | | | x | | | | Errors, Accuracy | split |
| Ciasullo et al. (2018) | Real-world | 1 | Italian footwear company | | | | x | | | | | | Business process outsourcing | NA | | | | | | | x | | | | | | | | | | | | | x | | | | | | | | x | | | Weights of decision-makers per context and over time | |
| Conforti et al. (2015) | Real-world | 1 | Claims handling process of insurance company | Person's X^2, Kruskal–Wallis test, Jonkheere's test, Kolmogorov–Smirnov Z two-samples test, Mann–Whitney test | | x | | | | | | x | Prescriptive business process monitoring | Event log | time, re-source, data | observation instance, features from decision point are mapped to risk | x | | x | | | | | | | | | | | | | | | x | | x | | | | | | | | | Percentage of faulty instances (mean, median) | |
| Cuzzocrea et al. (2016a) | Real-world | 1 | BPIC2011 | | | | | | x | | | | Pattern detection | Event log | data | pattern-based encoding | | | x | | | | | | | | | | x | | x | | x | | x | x | x | x | | x | x | | | | AUC, G-mean, F-measure | cross |
| Cuzzocrea et al. (2015) | Real-world | 1 | BPIC2011 | | | | | | x | | | | Pattern detection | Event log | | | | | x | | | | | | | | | | x | | x | | x | | x | x | x | x | | | x | | | | AUC, G-mean, Recall, Precision | cross |
| Cuzzocrea et al. (2016b) | Real-world | 1 | BPIC2011 | | | | | | x | | | | Pattern detection | Event log | data | index-based encoding, index-based encoding with HMM-based | | | x | x | | | | | | x | | | x | | x | | x | | | | | | | x | | | | | Accuracy, AUC, TP, FP, Precision, Recall | cross |
| Cuzzocrea et al. (2019) | Real-world | 2 | BPIC2013, Harbor | | | | | | | | | x | Predictive business process monitoring | Event log | re-source, data | event and trace abstraction functions, single trace vs. window across traces encoding | | | x | | | | x | | | | x | | | | | | | | x | x | x | | x | | x | | | | Precision, Recall, F-measure | split |

(*continued on next page*)

**Table C.7** (*continued*).

| Paper | Evaluation | | | | | BPM lifecycle phase | | | | | | | BPM task category | Data input | | | Feature engineering | | Model building | | | | | | | | | | | | | | | | | | | | | | | | | | Model assessment | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | | | | ML paradigm | | | | | | | | ML concept | | | | | | | | | | ML algorithm group | | | | | | | | | | | |
| | Data type | Number of datasets | Name of datasets | Statistical method | Further | Implementation available | Process identification | Process discovery | Process analysis | Process redesign | Process implementation | Process monitoring | BPM task category | Dataset type | Rel. context information | Feature extraction | Feature creation | Feature selection | Supervised learning | Semi-supervised learning | Reinforcement learning | Self-supervised learning | Unsupervised learning | Other | Multi-task learning | Active learning | Online/increm. learning | Transfer learning | Ensemble learning | Deep learning | Meta learning | Federated learning | Multi-view learning | Other | Instance-based | Decision tree-based | Clustering-based | Artificial-NN-based | Regression-based | Bayesian-based | Rule-based | Reinf.-learn.-based | Genetic-based | Other | Metrics | Validation strategy |
| De Koninck and De Weerdt (2019) | Real-world | 4 | BPIC2012, BPIC2015, BPIC2017, BPIC2018 | | | x | | | x | | | | Trace clustering | Event log | | activity profiles, pairs, 3-grams, PCA | | | | | | | x | | | | | | | | | | | x | | | x | | | | | | | | Internal consistency, Process model quality, Computational performance | |
| De Koninck, De Weerdt, et al. (2017) | Real-world, Synthetic | 10 | Telecom, Volvo, insurance, admin, purchase, tender, incident, environment, reviewing, incman | | | x | | | x | | | | Trace clustering | Event log | | binary | | | x | | | | x | | | | | | | | | | | x | x | | x | | | | | | | | Accuracy, Explanation length, Run time, Normalised mutual information | cross |
| De Koninck, Nelissen, et al. (2017) | Real-world | 1 | Newspaper reading data | | | x | | | x | | | | Trace clustering | Text | | | | | | x | | | x | | | | | | | | | | | x | | | x | | | | | | | | F1-Score, Normalised mutual information, Relative improvement | |
| De Koninck et al. (2021) | Real-world | 2 | BPIC2015, TABREAD | | | | | | x | | | | Trace clustering | Event log | | | | | | | | | x | | | | | | | | | | | x | | | x | | | | | | | | F1 score, Relative improvement score, Jaccard index, Percentage of violated must-link constraints | |
| De Koninck et al. (2018) | Real-world | 1 | BPIC2015 | | | x | x | | | | | | Process representation creation and learning | Event log | | word2vec, doc2vec | | | | | | x | x | | | | | | | | | | | x | | | x | x | | | | | | | Rand index, Normalised mutual information, Cosine distance, Behaviour profiles | |
| de Leoni et al. (2020) | Real-world | 1 | Dutch reintegration company | | | x | | | | | | x | Prescriptive business process monitoring | Event log | | frequency-based | | | x | | | | | | | | | | x | | | | | | x | x | | | | | | | | | KPI, Accuracy | split |
| de Leoni et al. (2013) | Synthetic | 21 | Synthetic logs | | | | | x | | | | | Decision mining | Event log | data | observation instances including values of variables in case prior to execution of an specific activity | | | x | | | | | | | | | | | | | | | x | | x | | | | | | | | | Relative information gain, Execution time | |
| de Leoni et al. (2016) | Real-world | 1 | Illness management | | | x | | | x | | | | Performance analysis | Event log | data, re-source, time, confor-mance | trace manipulations and event filter | x | | x | | | | | | | | | | | | | | | x | | x | | | | | | | | | F-measure, Correlation rules, Number of reclamations, Fitness, Mean Absolute Percentage Error (MAPE) | split |
| de Leoni et al. (2014) | Real-world | 1 | Event log from employee insurance agency | | | x | | | x | | | | Performance analysis | Event log | data, re-source, time, confor-mance | independent variables mapped to dependent variable | x | | x | | | | | | | | | | | | | | | x | | x | | | | | | | | | Accuracy | |
| De Maio et al. (2016) | Synthetic | 1 | Own | | | | | | | | | x | Group decision-making | Event log | data | preference vector | | | | | x | | | | | | x | | | | | | | | | | | | | | | x | | | Weights of decision makers over time | |
| De Weerdt et al. (2013) | Real-world, Synthetic | 104 | Helpdesk process, CRM process, second-line CRM process, incoming document handling | | | x | | | x | | | | Trace clustering | Event log | | | | | | | | | x | | | x | | | | | | | | | | | x | | | | | | | | Cluster set entropy, Weighted average f1-score, Place/transition connection degree | |
| De Weerdt et al. (2012) | Real-world | 1 | Incident management process | | | x | | | x | | | | Trace clustering | Event log | data, text | frequent keyword features, type information feature | | | x | x | | | | | | | | | | | | | | x | | x | x | | | | | | | | Accuracy | cross |
| Delcoucq et al. (2022) | Real-world, Synthetic | 3 | Hospital, Review, Repair | | | | | | | | x | | Resource allocation planning | Event log | re-source | | | | | | | | x | | | | | | | | | | | x | | | x | | | | | | | | F1-Score, R-Precision, R-Recall | |

(*continued on next page*)

**Table C.7** (*continued*).

| Paper | Evaluation | | | | | | BPM lifecycle phase | | | | | | BPM task category | Data input | | Feature engineering |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Data type | Number of datasets | Name of datasets | Statistical method | Further | Implementation available | Process identification | Process discovery | Process analysis | Process redesign | Process implementation | Process monitoring | BPM task category | Dataset type | Rel. context information | Feature extraction |
| Delias et al. (2019) | Real-world, Synthetic | 1 | BPIC2011 | | | | | | x | | | | Trace clustering | Event log | data | features grouped into 5 criteria |
| Di Francescomarino et al. (2016) | Real-world | 1 | BPIC2011 | Two-tailed non-parametric Wilcoxon test | | | | | | | | x | Predictive business process monitoring | Event log | | frequency-based, sequence-based |
| Di Francescomarino et al. (2017) | Real-world | 6 | BPIC2011, WABO, CoSeLoG, Helpdesk, BPIC2012, BPIC2013 | | | x | | | | | | x | Predictive business process monitoring | Event log | | one-hot |
| Di Mauro et al. (2019) | Real-world | 3 | Receipt phase, BPIC2012, Helpdesk | | | x | | | | | | x | Predictive business process monitoring | Event log | | embedding, time diff between current and last event |
| Diamantini et al. (2016) | Real-world, Synthetic | 2 | BPIC2011, synthetic log | | | | | x | | | | | Discovery of procedural process model | Event log | | |
| Effendi and Sarno (2017) | Real-world | 1 | Online book store management process | | | | | x | | | | | Decision mining | Event log | data | |
| Es-Soufi et al. (2016) | NA | NA | Past executions of design processes | | | | | | x | | | | Pattern detection | Event log | | |
| Evermann et al. (2017) | Real-world | 2 | BPIC2012, BPIC2013 | Kolmogorov–Smirnov test | | x | | | | | | x | Predictive business process monitoring | Event log | re-source | embedding |
| Ferreira et al. (2007) | Real-world, Synthetic | 2 | Own | | | | | | x | | | | Trace clustering | Event log | | |
| Ferreira and Gillblad (2009) | Synthetic | 1 | Synthetic event stream | | | x | | x | | | | | Discovery of procedural process model | Event stream | | symbol sequence |
| Ferreira et al. (2013) | Synthetic | 1 | Purchase process | | | | | x | | | | | Discovery of procedural process model | Event log | data | |
| Ferreira and Vasilyev (2015) | Real-world | 2 | BPIC2012, BPIC2013 | | | | | | x | | | | Performance analysis | Event log | time, re-source, data | predicates (e.g. performs, flow or user) |
| Firouzian et al. (2019a) | Real-world | 1 | BPIC2012 | | | | | | | | | x | Dynamic resource allocation | Event log | time, re-source | |
| Firouzian et al. (2019b) | Real-world, Synthetic | 2 | BPIC2012, Own | | | | | | | | | x | Predictive business process monitoring | Event log | data | |
| Folino et al. (2022) | Real-world | 5 | BPIC2012, hospital billing, road traffic fines | Friedman test, Nemenyi test, Critical Distance | | x | | | | | | x | Predictive business process monitoring | Event log | time, re-source, data | embedding, one-hot |

| Paper | Feature engineering | | Model building | | | | | | | | | | | | | | | | | | | | | | | | | | Model assessment | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | ML paradigm | | | | | | ML concept | | | | | | | | | | ML algorithm group | | | | | | | | | | | |
| | Feature creation | Feature selection | Supervised learning | Semi-supervised learning | Reinforcement learning | Self-supervised learning | Unsupervised learning | Other | Multi-task learning | Active learning | Online/increm. learning | Transfer learning | Ensemble learning | Deep learning | Meta learning | Federated learning | Multi-view learning | Other | Instance-based | Decision tree-based | Clustering-based | Artificial-NN-based | Regression-based | Bayesian-based | Rule-based | Reinf.-learn.-based | Genetic-based | Other | Metrics | Validation strategy |
| Delias et al. (2019) | x | | | | | | x | | | | | | | | | | | x | | | x | | | | | | | | Accuracy, cyclomatic number (CN), coefficient of connectivity (CNC), Coefficient of network complexity (CNC_k), Density, Performance (Case duration) | |
| Di Francescomarino et al. (2016) | | | x | | | | x | | | | | | x | | | | | | | x | x | | | | | | | | F1, Failure rate, Earliness | split |
| Di Francescomarino et al. (2017) | | | x | | | | | | | | | | | x | | | | | | | | x | | | | | | | Average Damerau–Levenshtein similarity | split |
| Di Mauro et al. (2019) | | | x | | | | | | | | | | | x | | | | | | | | x | | | | | | | Brier score, Accuracy | cross |
| Diamantini et al. (2016) | | | | | | | x | | | | | | | | | | | x | | | x | | | | | | | | Diversity, Complete-ness/Coverage, Representativeness, Frequency of subpatterns | |
| Effendi and Sarno (2017) | | | x | | | | | | | | | | | | | | | x | | x | | | | | | | | | | |
| Es-Soufi et al. (2016) | | | x | | | | | | | | | | | | | | | x | | x | | | | | | | | | | split |
| Evermann et al. (2017) | | | x | | | | | | | | | | | x | | | | | | | | x | | | | | | | Mean and standard deviation of training precision and validation precision | cross |
| Ferreira et al. (2007) | | | | | | | x | | | | | | | | | | | x | | | x | | | | | | | | | |
| Ferreira and Gillblad (2009) | | | | | | | x | | | | | | | | | | | x | | | x | | | x | | | | | G-Score, Execution time | |
| Ferreira et al. (2013) | | | | | | | x | | | | | | | | | | | x | | | x | | | x | | | | | | |
| Ferreira and Vasilyev (2015) | x | | x | | | | | | | | | | | | | | | x | | | | | | | x | | | | | |
| Firouzian et al. (2019a) | | | | | x | | | | | | | | | | | | | x | | | | | | | | x | | | Averaged cycle time, Entropy (sum of entropy of all work lists at a time t), Variance (of workloads of all resources at time t ->balancing of workload) | |
| Firouzian et al. (2019b) | | | x | | | | | | | | x | | x | | | | | | x | | | | | | | | | | Accuracy, F1-Score | |
| Folino et al. (2022) | x | | | x | | | | | | | | x | | x | | | | | | | | x | | | | | | | Accuracy, AUC | split |

(*continued on next page*)

**Table C.7** (*continued*).

| Paper | Evaluation | | | | | | BPM lifecycle phase | | | | | | BPM task category | Data input | | Feature engineering | | | Model building | | | | | | | | | | | | | | | | | | | | | | | | | | Model assessment | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | | | | ML paradigm | | | | | | ML concept | | | | | | | | | | ML algorithm group | | | | | | | | | | | |
| | Data type | Number of datasets | Name of datasets | Statistical method | Further | Implementation available | Process identification | Process discovery | Process analysis | Process redesign | Process implementation | Process monitoring | BPM task category | Dataset type | Rel. context information | Feature extraction | Feature creation | Feature selection | Supervised learning | Semi-supervised learning | Reinforcement learning | Self-supervised learning | Unsupervised learning | Other | Multi-task learning | Active learning | Online/increm. learning | Transfer learning | Ensemble learning | Deep learning | Meta learning | Federated learning | Multi-view learning | Other | Instance-based | Decision tree-based | Clustering-based | Artificial-NN-based | Regression-based | Bayesian-based | Rule-based | Reinf.-learn.-based | Genetic-based | Other | Metrics | Validation strategy |
| Folino et al. (2011) | Real-world, Synthetic | 4 | Claim handling process, 2 logistic processes, collaboration process | | | | | | | | | x | Predictive business process monitoring | Event log | data | | | | x | | | | | | | | | | | | | | | x | | x | | | | | | | | | Accuracy, Confidence | cross |
| Folino et al. (2012) | Real-world | 1 | Logistic process | | | x | | | | | | x | Predictive business process monitoring | Event log | time, data | list, set, bag | x | x | x | | | | | | | | | | | | | | | x | | x | x | | | | x | | | | RMSE, MAE, MAPE, Precision, Recall, Computation time | cross |
| Folino et al. (2015) | Real-world | 1 | BPIC2013 | | | | | | x | | | | Trace clustering | Event log | time, resource, data | t-view (context data and abstract activities) | x | | x | | | | | | | | | | | | | | | x | | x | x | | | | x | | | | Fitness, Behavioural precision | cross |
| Galanti et al. (2020) | Real-world | 1 | Italian bank account closure process | | | x | | | | | | x | Predictive business process monitoring | Event log | resource, data | one-hot | | | x | | | | | | | | | | | x | | | | | | | | x | | | | | | | MAE, F1-Score, AUC ROC, AUC PR | split |
| García-Bañuelos et al. (2014) | Real-world | 3 | BPIC2012a, insurance claims handling process | | | x | | x | | | | | Discovery of procedural process model | Event log | | depending on the trace clustering approach used | | | | | | | x | | | | | | | | | | | x | | | x | | | | | | | | Size, Structural complexity (CFC, ACD, CNC and density), Time performance | |
| Gerlach et al. (2022) | Real-world, Synthetic | 11 | Small, wide, medium, huge, p2p, paper, BPIC2012, BPIC2013, BPIC2015, BPIC2017, BPIC2020 | | | x | | | | | | x | Predictive business process monitoring | Event log | | | | | x | | | | | | | | | | | x | | | | | | | | x | | | | | | | | |
| Greco et al. (2006) | Synthetic | NA | Order management process, review paper process | | | | | x | | | | | Discovery of procedural process model | Event log | | | | | | | | | x | | | | | | | | | | | x | | | x | | | | | | | | Soundness | |
| Greco et al. (2004) | Synthetic | 1 | Own | | | | | x | | | | | Discovery of procedural process model | Event log | | | | | | | | | x | | | | | | | | | | | x | | | x | | | | | | | | Soundness | |
| Guzzo et al. (2021) | Real-world | 2 | BPIC2015, BPIC2019 | | | x | x | | | | | | Process representation creation and learning | Event log | time, resource, data | embedding | x | x | | | | x | | | | | | | | x | | | | | | | | x | | | | | | | Precision, Recall, F score, Adjusted rank index, Adjusted mutual information, Homogeneity score, Completeness score | cross |
| Ha et al. (2016) | Real-world | 1 | prBm6 | | | | | | x | | | | Trace clustering | Event log | | | | | | | | | x | | | | | | | | | | | x | | | x | | | | | | | | Precision, Fitness (compare vector with graph representation) | |
| Hanga et al. (2020) | Real-world | 4 | Helpdesk, BPIC2012, BPIC2013, RTFM | | | | | | | | | x | Predictive business process monitoring | Event log | | one-hot | | | x | | | | | | | | | | | x | | | | | | | | x | | | | | | | Accuracy, Similarity score | split |
| Harl et al. (2020) | Real-world | 1 | BPIC2012 | | | x | | | | | | x | Predictive business process monitoring | Event Log | | | | | x | | | | | | | | | | | x | | | | | | | | x | | | | | | | Accuracy | split |
| Heinrich et al. (2021) | Real-world | 11 | BPIC2011, BPIC2012, BPIC2013, Helpesk, EnvLog | | | | | | | | | x | Predictive business process monitoring | Event log | data | embedding | | | x | | | | | | | | | | | x | | | | | | | | x | | | | | | | Accuracy, Precision, Recall, F1 score | cross |
| Herbst (2000) | Real-world, Synthetic | NA | Own | | | | | x | | | | | Discovery of procedural process model | Event log | | dependency graph | | | | | | | x | | | | | | | | | | | x | | | | | x | | | | | | Nodes/edges of result model, No. splits, Time, Correct, Log-likelihood | |
| Herbst and Karagiannis (2000) | Synthetic | 1 | Simplified release process in car manufacturing | | | | | x | | | | | Discovery of procedural process model | Event log | | | | | | | | | x | | | | x | | | | | | | | | | | | x | | | | | | Nodes/edges of result model, Search steps, Time spent, Log-likelihood | |

(*continued on next page*)

**Table C.7** (*continued*).

| Paper | Evaluation | | | | | BPM lifecycle phase | | | | | | | BPM task category | Data input | | Feature engineering |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Data type | Number of datasets | Name of datasets | Statistical method | Further | Implementation available | Process identification | Process discovery | Process analysis | Process redesign | Process implementation | Process monitoring | BPM task category | Dataset type | Rel. context information | Feature extraction |
| Herbst and Karagiannis (2004) | Real-world, Synthetic | NA | Different Simulated/Two Industrial logs | | Workflow model discovery with different experts and with different models | | | x | | | | | Discovery of procedural process model | Event log | | |
| Hsieh et al. (2021) | Real-world | 1 | BPIC2012 | | | | | | | | | x | Predictive business process monitoring | Event log | re-source, data | one-hot |
| Huang et al. (2011) | Real-world | 1 | Radiology CT-scan examination process | | | x | | | | | | x | Dynamic resource allocation | Event log | time, re-source, data | |
| Huang et al. (2010) | Real-world | 1 | Radiology CT-scan examination process | | | x | | | | | | x | System adaption | Other | | process data objects |
| Huo et al. (2021) | Real-world, Synthetic | 6 | BPIC2012, BPIC2013, BPIC2017, Loan, Large, Huge | | | | | | | | | x | Descriptive business process monitoring | Event Log | | one-hot |
| Jalayer et al. (2020) | Real-world | 4 | Helpdesk, BPIC2012, BPIC2012 W, BPIC2012O | | | | | | | | | x | Predictive business process monitoring | Event log | | |
| Jlailaty et al. (2017) | Real-world | 1 | 250 E-Mails taken from a Ph.D student | | | | x | | | | | | Event log creation | Text | | word2vec, tf-idf |
| Jung et al. (2009) | Real-world | 1 | Insurance process | | | | | | x | | | | Trace clustering | Set of process models | | process vector |
| Jung et al. (2008) | Synthetic | 1 | Insurance Processes | | | | | | x | | | | Trace clustering | Set of business process models | | process vector |
| Junior et al. (2020) | Real-world, Synthetic | 11 | BPIC2012, BPIC2013, BPIC2015, BPIC2017 | | | x | | | x | | | | Pattern detection | Event log | | word2vec |
| Kang et al. (2012) | Real-world | 1 | Compressive strength process | | | | | | | | | x | Prescriptive business process monitoring | Tabular data | | |
| Kazakov et al. (2018) | NA | 2 | NA | | | | | | | | x | | Resource allocation planning | NA | | BP function features |
| Khan et al. (2021) | Real-world | 1 | MIMIC | | | | | | | | | x | Prescriptive business process monitoring | Event Log | data | |
| Khodyrev and Popova (2014) | Real-world | 2 | CarRep, CFactRep | | | | | | x | | | | Process simulation | Event log | time, data | petri-net-element-based |
| Kim et al. (2002) | Real-world | 1 | Order processing workflow | | | | | x | | | | | Discovery of procedural process model | Case base (set of cases in the form of is-a hierarchy), vocabulary base (task names and their nouns/verbs) | | |
| Klijn and Fahland (2020) | Real-world | 2 | BPIC2020, RTP | | | x | | | | | | x | Predictive business process monitoring | Event log | inter-case | last state/aggregate/index-based encoding, prefix-length/cluster/single bucketing |
| Krajsic and Franczyk (2020) | Real-world | 1 | BPIC2019 | | | | | | x | | | | Pattern detection | Event log | | one-hot |

| Paper | Feature engineering | | Model building: ML paradigm | | | | | | ML concept | | | | | | | | | | ML algorithm group | | | | | | | | | | Model assessment | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Feature creation | Feature selection | Supervised learning | Semi-supervised learning | Reinforcement learning | Self-supervised learning | Unsupervised learning | Other | Multi-task learning | Active learning | Online/increm. learning | Transfer learning | Ensemble learning | Deep learning | Meta learning | Federated learning | Multi-view learning | Other | Instance-based | Decision tree-based | Clustering-based | Artificial-NN-based | Regression-based | Bayesian-based | Rule-based | Reinf.-learn.-based | Genetic-based | Other | Metrics | Validation strategy |
| Herbst and Karagiannis (2004) | | | | | | | x | | | | | | | | | | | x | | | | | | x | | | | | Semantic equality between process models | |
| Hsieh et al. (2021) | | | x | | | | | | | | | | | x | | | | | | | | x | | | | | | | Sparsity as average Levenshtein distance, Proximity as euclidean distance | split |
| Huang et al. (2011) | | | | | x | | | | | | | | | | | | | x | | | | | | | | x | | | Flow time, MSE (with 5% confidence interval), STD | |
| Huang et al. (2010) | | | | | x | | | | | | | | | | | | | x | | | | | | | | x | | | Reward value, Flow time, Confidence level (90%) | |
| Huo et al. (2021) | | | | | | x | | | | | | | | x | | | | | | | | x | | | | | | | F1-score | split |
| Jalayer et al. (2020) | | | x | | | | | | | | | | | x | | | | | | | | x | | | | | | | Accuracy, Demerau-Levenshtein | split |
| Jlailaty et al. (2017) | | x | | | | x | x | | | | | | | | | | | x | | | x | x | | | | | | | F1-Score, Rand-index, Precision, Recall, Purity | cross |
| Jung et al. (2009) | | | | | | | x | | | | | | | | | | | x | | | x | | | | | | | | Cosine similarity | |
| Jung et al. (2008) | | | | | | | x | | | | | | | | | | | x | | | x | | | | | | | | Similarity | |
| Junior et al. (2020) | | | x | | | x | | | | | | | | | | | | x | x | | | x | | | | | | | F score | |
| Kang et al. (2012) | | | x | | | | | | | | | | | | | | | x | x | | | | | | | | | | False warning, Missing warning, True warning, True no-warning, Error rate, Probability of target value, FP, FN, TP, TN | cross |
| Kazakov et al. (2018) | | | x | | | | | | | | | | | | | | | x | | | | | | x | | | | | | |
| Khan et al. (2021) | | | | | x | | | | | | | | | x | | | | | | | | x | | | | x | | | Estimated policy value | split |
| Khodyrev and Popova (2014) | x | x | x | | | | | | | | | | | | | | | x | | x | | | | | | | | | Confidence intervals | split |
| Kim et al. (2002) | | | x | | | | | | | | | | | | | | | x | x | | | | | | | | | | Similarity | |
| Klijn and Fahland (2020) | x | | x | | | | | | | | | | x | | | | | | | x | | | | | | | | | MAE, Computation time | split |
| Krajsic and Franczyk (2020) | | | | | | x | | | | | | | | | | | | x | | | | x | | | | | | | Accuracy, Misclassification, Precision, Recall, F1-Score, Specificity | split |

(*continued on next page*)

**Table C.7** (*continued*).

| Paper | Evaluation | | | | | BPM lifecycle phase | | | | | | | BPM task category | Data input | | | Feature engineering | | Model building | | | | | | | | | | | | | | | | | | | | | | | | | | Model assessment | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | | | | ML paradigm | | | | | | ML concept | | | | | | | | | | ML algorithm group | | | | | | | | | | | |
| | Data type | Number of datasets | Name of datasets | Statistical method | Further | Implementation available | Process identification | Process discovery | Process analysis | Process redesign | Process implementation | Process monitoring | BPM task category | Dataset type | Rel. context information | Feature extraction | Feature creation | Feature selection | Supervised learning | Semi-supervised learning | Reinforcement learning | Self-supervised learning | Unsupervised learning | Other | Multi-task learning | Active learning | Online/increm. learning | Transfer learning | Ensemble learning | Deep learning | Meta learning | Federated learning | Multi-view learning | Other | Instance-based | Decision tree-based | Clustering-based | Artificial-NN-based | Regression-based | Bayesian-based | Rule-based | Reinf.-learn.-based | Genetic-based | Other | Metrics | Validation strategy |
| Krajsic and Franczyk (2021) | Synthetic | 4 | Loan Application Process Variants | | | | | | | | | x | Descriptive business process monitoring | Event log | | | | | | | | x | x | | | | | | | x | | | | | x | x | | x | | | | | | | Excess-Mass Score, Mass-Volume-Score | split |
| Kratsch et al. (2022) | Real-world | 1 | Own | Friedman test | | x | x | | | | | | Event log creation | Video data | | | | | | | | | | x | | | | | | | | | | x | | | | | | | | | | x | | |
| Lakshmanan et al. (2015) | Synthetic | 1 | Automobile Insurance Claims Handling Process | | | | | | | | | x | Predictive business process monitoring | Event log | data | | | | x | | | | | | | | | | | | | | | x | | x | | | | | | | | | Normalised Log Likelihood | split |
| Lamma, Mello, Montali, et al. (2007) | Real-world | 2 | Cervical cancer screening careflow, NetBill | | | | | x | | | | | Discovery of declarative process model | Event log | | | | | x | | | | | | | | | | | | | | | x | | | | | | | x | | | | Accuracy | cross |
| Lamma, Mello, Riguzzi, et al. (2007) | Real-world | 2 | Electronic auction, NetBill | | | | | x | | | | | Discovery of declarative process model | Interaction log | | | | | x | | | | | | | | | | | | | | | x | | | | | | x | x | | | | | |
| Lee et al. (2022) | Synthetic | 6 | Loan application process | | | x | | | | | | x | Predictive business process monitoring | Event log | time | one-hot | x | | x | | | | | | | | | | x | x | | | | | | x | | x | | | | | | | F1-score | split |
| Lee et al. (2021) | Real-world | 1 | Hospital Billing | | | x | | | | | | x | Descriptive business process monitoring | Event log | | caseid-activity-pairs | | | | | | | x | | | | | | | | | | | x | | | x | | | x | | | | | | cross |
| Lee et al. (2018) | Real-world | 2 | BPIC2012, BPIC2013 | | | | | | | | | x | Predictive business process monitoring | Event log | | frequency-based, one-hot | | | x | | | | | | | | | | | | | | | x | | | | | | | | | | x | Precision, RMSE | split |
| Leno et al. (2018) | Real-world, Synthetic | 12 | BPIC2015, BPIC2011, synthetic log | | | x | | x | | | | | Discovery of declarative process model | Event log | data | violation and fulfilment vectors | x | | x | | | | x | | | | | | | | | | | x | | x | x | | | | x | | | | Support, Confidence, Execution time | |
| Leno et al. (2020) | Real-world, Synthetic | 5 | BPIC2011, BPIC2013, BPIC2017, Sepsis, RTFM | | | x | | x | | | | | Discovery of declarative process model | Event log | data | violation and fulfilment vectors | x | | x | | | | | | | | | | x | | | | | | | x | | | | | x | | | | Support, Confidence, Recall, Precision, F-Score | split |
| Li et al. (2010) | Real-world, Synthetic | 2 | Synthetic process models | | Case study | | | x | | | | | Discovery of procedural process model | Set of business process models | | Order matrix | | | | | | | x | | | | | | | | | | | x | | | x | | | | | | | | Computation time, Average weighted distance | |
| Liu et al. (2012) | Real-world | 10 | Manufacturing processes | | | | | | | | x | | Resource allocation planning | Event log | time, re-source | frequent 3-itemset | | | | | | | x | | | | | | | | | | | x | | | | | | | x | | | | Precision, Time elapsed, Correct prediction count, Strong rules count | cross |
| Liu et al. (2008) | Real-world | 3 | Own | | | | | | | | | x | Dynamic resource allocation | Event log | re-source | | | | x | | | | | | | | | | | | | | | x | x | x | | | | x | | | | | Accuracy | |
| López et al. (2021) | Real-world | 1 | NA | | | x | | x | | | | | Process model extraction | Text | | bert model | | | x | | | x | | | | | | x | | x | | | | | | | | x | | | | | | | Precision, Recall, F1-Score | split, cross |
| Lu et al. (2016) | Real-world | 1 | CORDYS BPM | | | | | x | | | | | Discovery of procedural process model | Event log | data | | | | | | | | x | | | | | | | | | | | x | | | x | | | | | | | | Flow condition, Descriptive comparison to related work: Ability to model decomposable cyclic dependencies, to deal with noise | |
| Maggi et al. (2018) | Real-world, Synthetic | 84 | BPIC2012, BPIC2013, BPIC2014, RTFM, Sepsis, BPIC2017, synthetic logs | | | | | x | | | | | Discovery of declarative process model | Event log | | | | | | | | | x | | | | | | | | | | | x | | | | | | | x | | | | Computation time, Memory consumption | |
| Maggi et al. (2012) | Real-world, Synthetic | NA | Synthetic logs, process from dutch municipality | | | | | x | | | | | Discovery of declarative process model | Event log | | frequent activity sets | | | | | | | x | | | | | | | | | | | x | | | | | | | x | | | | Computation time, Conditional-probability increment ratio | |
| Mannhardt et al. (2016) | Real-world | 2 | RTFM, Sepsis | | | x | | x | | | | | Decision mining | Event log | | | | | x | | | | | | | | | | | | | | | x | | x | | | | | | | | | Fitness, Precision (compare to different methods) | |
| Márquez-Chamorro, Resinas, Ruiz-Cortés, and Toro (2017) | Real-world | 2 | BPIC2013, IT Department of Health Service | | | x | | | | | | x | Predictive business process monitoring | Event log | time, re-source, data | Window-based | x | | x | | | | | | | | | | | | | | | x | | | | | | | | | x | | Precision, Recall, Specificity, Mean, StDev, F-Measure, AUC-ROC | cross |

(*continued on next page*)

**Table C.7** (*continued*).

| Paper | Evaluation | | | | | BPM lifecycle phase | | | | | | | BPM task category | Data input | | | Feature engineering | | | Model building: ML paradigm | | | | | | Model building: ML concept | | | | | | | | | | Model building: ML algorithm group | | | | | | | | | | Model assessment | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Data type | Number of datasets | Name of datasets | Statistical method | Further | Implementation available | Process identification | Process discovery | Process analysis | Process redesign | Process implementation | Process monitoring | BPM task category | Dataset type | Rel. context information | Feature extraction | Feature creation | Feature selection | Supervised learning | Semi-supervised learning | Reinforcement learning | Self-supervised learning | Unsupervised learning | Other | Multi-task learning | Active learning | Online/increm. learning | Transfer learning | Ensemble learning | Deep learning | Meta learning | Federated learning | Multi-view learning | Other | Instance-based | Decision tree-based | Clustering-based | Artificial-NN-based | Regression-based | Bayesian-based | Rule-based | Reinf.-learn.-based | Genetic-based | Other | Metrics | Validation strategy |
| Martens et al. (2014) | Real-world | 1 | CoSeLoG | | | | | x | | | | | Discovery of procedural process model | Set of business process models | | graph-based | | | x | | | | | | | | | | | | | | | x | | | | | | | | | x | | Graph-Edit Distance, Precision, Reall, F-Measure | |
| Măruşter et al. (2006) | Real-world, Synthetic | 2 | Simulated Log, Dutch governmental fine collection | | | | | x | | | | | Discovery of procedural process model | Event log | | | | | x | | | | | | | | | | | | | | | x | | | | | | | x | | | | F-measure | cross |
| Măruşter et al. (2002) | Synthetic | 5 | Own | | | | | x | | | | | Discovery of procedural process model | Event log | | | | | x | | | | | | | | | | | | | | | x | | | | | x | | | | | | Accuracy | cross |
| Mehdiyev et al. (2020) | Real-world | 3 | BPIC2012, BPIC2013, helpdesk | | | | | | | | | x | Predictive business process monitoring | Event log | re-source, data | n-gram, hashing | | | x | | | x | | | | | | | | x | | | | | | | | x | | | | | | | Accuracy, Precision, Recall, F-measure, MCC, AUC | cross |
| Metzger et al. (2020) | Real-world | 3 | BPIC2012, BPIC2017, Traffic | | | x | | | | | | x | System adaption | Event log | | | | | x | | x | | | | | | x | | x | x | | | | | | | | x | | | | x | | | Execution cost | |
| Metzger et al. (2019) | Real-world | 4 | Cargo2000, Traffic, BPIC2012, BPIC2017 | | | x | | | | | | x | System adaption | Event log | | one-hot | | | x | | | | | | x | | | | x | x | | | | | | | | x | | | | | | | Cost savings | |
| Montani and Leonardi (2012) | Real-world | 1 | Stoke management process | | | x | | | | | | x | Descriptive business process monitoring | Event log | time | | x | | x | | | | | | | | | | | | | | | x | x | | x | | | | | | | | Cluster homogeneity | |
| Mustansir et al. (2022) | Real-world | 3 | Own | | | | | | | x | | | Redesign suggestion identification | Text | | | | | x | | | | | | | | | | | x | | | | | | | | x | | | | | | | F1-Score | cross |
| Nguyen et al. (2019) | Real-world, Synthetic | 2 | BPIC2012, BPIC2013 | | | x | x | | | | | | Data improvement | Event log | time | one-hot encoding, standardising | x | | | | | x | | | | | | | | x | | | | x | | | | x | | | | | | | Precision, Recall, F1-score | |
| Nguyen et al. (2016) | Real-world, Synthetic | 3 | BPIC2013, Receipt, bank | | | | | | x | | | | Trace clustering | Event log | | | | | | | | | x | | | | | | | | | | | x | | | x | | | | | | | | Weighted average conformance fitness, Weighted average structure complexity | |
| Nolle, Luettgen, et al. (2018) | Real-world, Synthetic | 700 | BPIC2012, BPIC2017, P2P, Small, Medium, Large, Huge, Wide | | | x | | | x | | | | Pattern detection | Event log | re-source | one-hot | | | | | | x | | | | | | | | | | | | x | | | | x | | | | | | | F1-Score, DAE error heatmap | |
| Nolle et al. (2019) | Real-world, Synthetic | 13 | BPIC2012, BPIC2013, BPIC2015, BPIC2017, Paper, P2P, Small, Medium, Large, Huge, Gigantic, Wide, Anonymous | Nemenyi post hoc test, Friedman test | | x | | | | | | x | Descriptive business process monitoring | Event log | time, re-source, data | integer, embedding | x | | x | | | | | | x | | | | | x | | | | | | | | x | | | | | | | Precision, Recall, F1-Score, Variance, Confusion matrix, Critical difference, Anomaly scores | |
| Nolle, Seeliger, et al. (2018) | Real-world, Synthetic | 81 | BPIC2012, BPIC2013, BPIC2015, BPIC2017, COMP, P2P, Small, Medium, Large, Huge, Wide | | | x | | | | | | x | Descriptive business process monitoring | Event log | time, re-source, data | integer, embedding | x | | x | | | | | | x | | | | | x | | | | | | | | x | | | | | | | F1-Score, Anomaly scores | |
| Nolle et al. (2020) | Synthetic | 252 | Own | | | x | x | | | | | | Data improvement | Event log | time, re-source, data | | | | x | | | | | | x | | | | | x | | | | | | | | x | | | | | | | F1-Scores, Correction Accuracy, Average Error, Alignment Optimality | |
| Park and Song (2020) | Real-world | 3 | Helpdesk, BPIC2012, healthcare service process of South Korean hospital | | | x | | | | | | x | Predictive business process monitoring | Event log | time | transition-system-based | x | | x | | | | | | | | | | | x | | | | | | | | x | | | | | | | MAE, Mean absolute percentage error | cross |
| Park and Song (2019) | Real-world, Synthetic | 2 | BPIC2012, Own | | | | | | | | | x | Dynamic resource allocation | Event log | time, re-source | one-hot (activity, resource) | x | | x | | | | | | x | | | | | x | | | | | | | | x | | | | | | | Accuracy | |
| Pasquadibis-ceglie et al. (2019) | Real-world | 2 | Helpdesk, BPIC2012 | | | | | | | | | x | Predictive business process monitoring | Event log | | frequency-based, temporal features | | | x | | | | | | | | | | | x | | | | | | | | x | | | | | | | Accuracy | split |

(*continued on next page*)

**Table C.7** (*continued*).

| Paper | Evaluation | | | | | BPM lifecycle phase | | | | | | | BPM task category | Data input | | Feature engineering | | | Model building ML paradigm | | | | | | | ML concept | | | | | | | | | | ML algorithm group | | | | | | | | | | Model assessment | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Data type | Number of datasets | Name of datasets | Statistical method | Further | Implementation available | Process identification | Process discovery | Process analysis | Process redesign | Process implementation | Process monitoring | BPM task category | Dataset type | Rel. context information | Feature extraction | Feature creation | Feature selection | Supervised learning | Semi-supervised learning | Reinforcement learning | Self-supervised learning | Unsupervised learning | Other | Multi-task learning | Active learning | Online/increm. learning | Transfer learning | Ensemble learning | Deep learning | Meta learning | Federated learning | Multi-view learning | Other | Instance-based | Decision tree-based | Clustering-based | Artificial-NN-based | Regression-based | Bayesian-based | Rule-based | Reinf.-learn.-based | Genetic-based | Other | Metrics | Validation strategy |
| Pasquadibisceglie, Appice, Castellano, and Malerba (2020) | Real-world | 3 | BPIC2012, Receipt Phase, BPIC2013 | | | x | | | | | | x | Predictive business process monitoring | Event log | time, re-source | rbg encoding | x | | x | | | | | | | | | | | x | | | | | | | | x | | | | | | | Accuracy, Precision, Recall, F1-Score, Standard deviation | cross |
| Pasquadibisceglie, Appice, et al. (2021) | Real-world | 5 | BPIC2012, BPIC2013, Receipt, BPIC2017, BPIC2020 | Friedman test | | x | | | | | | x | Predictive business process monitoring | Event log | | one-hot | | | x | | | | | | x | | | | | x | | | x | | | | | x | | | | | | | Accuracy, Macro-precision, Macro-recall, Macro-Fscore, Macro-AUCROC, Macro AUCPR | split |
| Pasquadibisceglie, Appice, Castellano, Malerba, and Modugno (2020) | Real-world | 4 | Sepsis, BPIC2011, BPIC2012, Production | Nemenyi test | | x | | | | | | x | Predictive business process monitoring | Event log | | min–max normalisation | | | x | | | | | | | | | | | x | | | | | | | | x | | | | | | | AUC, F-Score | split |
| Pasquadibisceglie, Castellano, et al. (2021) | Real-world | 4 | Sepsis, BPIC2011, BPIC2012, Production | | | x | | | | | | x | Predictive business process monitoring | Event log | | Aggregation encoding | | | x | | | | | | | | | | | x | | | | | | | | x | | | | | | | ROC AUC | |
| Pauwels and Calders (2020) | Real-world | 4 | BPIC2012, BPIC2015, Helpdesk, BPIC2018 | | | x | | | | | | x | Predictive business process monitoring | Event log | | | | | | | | | x | | | | | | | | | | | x | | | | | | x | | | | | Accuracy of suffix prediction, Runtime | split |
| Pauwels and Calders (2021) | Real-world | 4 | Helpdesk, BPIC2011, BPIC2012, BPIC2015 | | | x | | | | | | x | Predictive business process monitoring | Event Log | | one-hot | | | x | | | | x | | | | x | | | x | | | | | | | | x | | x | | | | | Accuracy, Runtime | |
| Pegoraro et al. (2021) | Real-world | 2 | BPIC2016, MIMIC | | | | | | | | | x | Predictive business process monitoring | Event log | text | one-hot encoding, min–max normalisation, exchangeable text encoding model (BoNG, Doc2Vec, LDA, BoW) | | | x | | | | | | | | | | | x | | | | | | | | x | | | | | | | F1-Score, MAE | |
| Pfeiffer et al. (2021) | Real-world, Synthetic | 7 | Helpdesk, BPIC2012, WE, BPIC2013, CP, BPIC2017, RFP, MobIS | | | x | | | | | | x | Predictive business process monitoring | Event log | re-source, time, cost, data | gramian angular fields | | | x | | | x | | | | | | x | | x | | | | | x | | | x | | | | | | | Accuracy, MAE | split |
| Pflug and Rinderle-Ma (2016) | Real-world | 1 | BPIC2011 | | | | | | | | | x | Dynamic resource allocation | Event log | data | | | | | | | | x | | | | | | | | | | | x | | | x | | | | | | | | Cumulated throughput time, Average throughput time at activity fetching | |
| Pham et al. (2021) | Real-world | 5 | BPIC2012, BPIC2015, BPIC2017, Helpdesk | | | | | | | | | x | Predictive business process monitoring | Event log | re-source | one-hot | | | x | | | | | | | | | | | x | | | | | | | | x | | | | | | | Accuracy | split |
| Philipp et al. (2020) | Real-world | 3 | BPIC2012, BPIC2013, German software company dataset | | | | | | | | | x | Predictive business process monitoring | Event log | | positional encoding | | | x | | | | | | | | | | | x | | | | | | | | x | | | | | | | Accuracy | split |
| Polančič et al. (2020) | Real-world | 1 | Questionnaire | T-test | | | | x | | | | | Process model extraction | Image | | | | | x | | | | | | | | | x | | x | | | | | | | | x | | | | | | | Accuracy | split, cross |
| Polato et al. (2018) | Real-world | 3 | RTFM, Helpdesk, BPIC2012 | | | | | | | | | x | Predictive business process monitoring | Event log | data | one-hot, set of activity (set, bag, list), transition-system-based features | x | | x | | | | | | | | | | | | | | | x | | | | | x | x | | | | | Accuracy, MAPE, RMSPE, MAE | split, cross |
| Polato et al. (2014) | Real-world | 1 | RTFM | | | x | | | | | | x | Predictive business process monitoring | Event log | data | last, one-hot | | | x | | | | | | | | | | | | | | | x | x | | | | | x | | | | | MAPE, RMSPE | cross |

(*continued on next page*)

**Table C.7** (*continued*).

| Paper | Evaluation | | | | | BPM lifecycle phase | | | | | | | BPM task category | Data input | | Feature engineering | | | Model building | | | | | | | | | | | | | | | | | | | | | | | | | | Model assessment | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | | | | ML paradigm | | | | | | ML concept | | | | | | | | | | ML algorithm group | | | | | | | | | | | |
| | Data type | Number of datasets | Name of datasets | Statistical method | Further | Implementation available | Process identification | Process discovery | Process analysis | Process redesign | Process implementation | Process monitoring | BPM task category | Dataset type | Rel. context information | Feature extraction | Feature creation | Feature selection | Supervised learning | Semi-supervised learning | Reinforcement learning | Self-supervised learning | Unsupervised learning | Other | Multi-task learning | Active learning | Online/increm. learning | Transfer learning | Ensemble learning | Deep learning | Meta learning | Federated learning | Multi-view learning | Other | Instance-based | Decision tree-based | Clustering-based | Artificial-NN-based | Regression-based | Bayesian-based | Rule-based | Reinf.-learn.-based | Genetic-based | Other | Metrics | Validation strategy |
| Qafari and van der Aalst (2019) | modified Real-world | 1 | Hospital Billing | | | x | | | x | | | | Performance analysis | Event log | data | situation features | | | x | | | | | | | | | | | | | | | x | | x | | | | | | | | | Accuracy, Discrimination | split |
| Qian et al. (2020) | Real-world | 2 | Cooking Recipes, Maintenance Manuals | Two-tailed t-test | | x | | x | | | | | Process model extraction | Text | | word2vec, bert | | | x | | | x | | | x | | | | | x | | | | | | | | x | | | | | | | Accuracy, Behaviour similarity | cross |
| Qiao et al. (2011) | Real-world, Synthetic | 5 | SAP (industry), SAP (cross industry), SAP (reference), APQC, Real practice | | | | | x | | | | | Discovery of procedural process model | Set of business process models, process description documents and keywords | | vector-space model, graph-based | | | | | | | x | | | | | | | | | | | x | | | x | | | x | | | | | Precision, Running time | split |
| Rizzi et al. (2020) | Real-world, Synthetic | 8 | BPIC2011, Claim Management | | | x | | | | | | x | Predictive business process monitoring | Event log | | frequency, simple, complex | | x | x | | | | | | | | | | x | | | | | | | x | | | | | | | | | Accuracy, AUC | split |
| Rogge-Solti and Kasneci (2014) | Real-world | 1 | Dutch surgery log | | | x | | | x | | | | Pattern detection | Event log | time | | | | | | | | x | | | | | | | | | | | x | | | | | | x | | | | | ROC, Accuracy | cross |
| Rozinat and van der Aalst (2006) | NA | NA | Variety of domains (e.g. healthcare) | | | x | | x | | | | | Decision mining | Event log | | | | | x | | | | | | | | | | | | | | | x | | x | | | | | | | | | | |
| Samiri et al. (2017) | Synthetic | 1 | Own | Auto-correlation, partial auto-correlation, mean percentage of residuals, augmented Dickey–Fuller for unit-root test | | | | | | | | x | System adaption | Event log | | | | | | | x | | | | | | | | | | | | | x | | | | | | | | x | | | Mean error, MAE, root-mean-square error, mean percentage error, mean absolute percentage error | |
| Saraeian et al. (2019) | Synthetic | 1 | Supply chain dataset | | | | | | | | | x | System adaption | Tabular data | | | | | x | | | | | | | | | | | | | | | x | | | | x | | | | | | | MAE, relative root-mean square error (RRMSE), root RMSE | |
| Sarno, Ginardi, et al. (2013) | Synthetic | 1 | Synthetic process models | | | | | | x | | | | Trace clustering | Set of business process models | | process matrix | | | | | | | x | | | | | | | | | | | x | | | x | | | | | | | | Silhouette index, Similarity, Dissimilarity | |
| Sarno, Sari, et al. (2013) | Real-world | 1 | Special order case | | | x | | x | | | | | Decision mining | Event log | re-source, data | | | | x | | | | | | | | | | | | | | | x | | x | | | | | | | | | | |
| Sarno and Sungkono (2016a) | Synthetic | NA | Own | | | | | x | | | | | Discovery of procedural process model | Event log | | | | | | | | | x | | | | | | | | | | | x | | | x | | | x | | | | | Fitness, Validity | |
| Sarno and Sungkono (2016b) | Synthetic | NA | Own | | | | | x | | | | | Discovery of procedural process model | Event log | | | | | | | | | x | | | | | | | | | | | x | | | x | | | x | | | | | Fitness | |
| Savickas and Vasilecas (2018) | Real-world, Synthetic | 3 | MP, DFI, EIMSD | | | | | | x | | | | Performance analysis | Event log | time, re-source, data | sequence matrix | x | | | | | | x | | | | | | | | | | | x | | | | | | x | | | | | Rate of correct, partial, wrong, and missing activities | split |
| Seeliger et al. (2021) | Real-world, Synthetic | 4 | TC-DS18, TC-DS20, BPIC2015, BPIC2019 | | | x | x | | | | | | Process representation creation and learning | Event log | re-source, data | embedding, one-hot | | | x | | | | | | | | | | | x | | | | | | | | x | | | | | | | F1-Bcubed, Fitness, Precision, Simplicity | |
| Seeliger et al. (2018) | Real-world, Synthetic | 293 | Own | | | x | | | x | | | | Trace clustering | Event log | re-source, data | frequent itemsets, integer | | | | | | | x | | | | | | | | | | | x | | | x | | | | | | | | Weighted fitness, Weighted precision, Weighted generalisation, Purity, Adjusted rand index, Cluster set entropy, Graph density, Cyclomatic number, Coefficient of connectivity, Coefficient of network complexity | |

(*continued on next page*)

**Table C.7** (*continued*).

| Paper | Evaluation | | | | | BPM lifecycle phase | | | | | | | BPM task category | Data input | | Feature engineering | Model building | | | | | | | | | | | | | | | | | | | | | | | | | | | | Model assessment | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | | | | ML paradigm | | | | | | ML concept | | | | | | | | | | ML algorithm group | | | | | | | | | | | |
| | Data type | Number of datasets | Name of datasets | Statistical method | Further | Implementation available | Process identification | Process discovery | Process analysis | Process redesign | Process implementation | Process monitoring | BPM task category | Dataset type | Rel. context information | Feature extraction | Feature creation | Feature selection | Supervised learning | Semi-supervised learning | Reinforcement learning | Self-supervised learning | Unsupervised learning | Other | Multi-task learning | Active learning | Online/increm. learning | Transfer learning | Ensemble learning | Deep learning | Meta learning | Federated learning | Multi-view learning | Other | Instance-based | Decision tree-based | Clustering-based | Artificial-NN-based | Regression-based | Bayesian-based | Rule-based | Reinf.-learn.-based | Genetic-based | Other | Metrics | Validation strategy |
| Senderovich et al. (2017) | Real-world | 2 | Israeli emergency department process, manufacturing process | | | x | | | | | | x | Predictive business process monitoring | Event log | time, re-source, data, inter-case | Window-based | x | | x | | | | | | | | | | x | | | | | | | x | | | x | | | | | | MAE, RMSE | split, cross |
| Senderovich et al. (2015) | Real-world, Synthetic | 3 | Israeli bank's call centre, israeli telecommunication company call centre, synthetic log | | | | | | | | | x | Predictive business process monitoring | Event log | time | transition-system-based | | x | x | | | | | | | | | | | | | | | x | | x | | | x | | | | | | RASE (prediction error, seconds) | split |
| Senderovich et al. (2014) | Real-world | 1 | Call-centre log | | | | | | | | x | | Resource allocation planning | Service log | | | | | x | | | | | | | | | | x | | | | | | | x | | | x | | | | | | Misclassification rate | split |
| Shraga et al. (2020) | Real-world, Synthetic | 5 | Own, hospital, sepsis | | | x | | x | | | | | Discovery of procedural process model | Event log | time, re-source, data | | x | | | | | | x | | | | | | | | | | | x | | | x | | | | | | | | Fitness, Precision, Generalisation | split |
| Silva et al. (2005) | Synthetic | 1 | Simulated report process | | | | | x | | | | | Discovery of procedural process model | Event log | | | | | | | | | x | | | | | | | | | | | x | | | | | | x | | | | | X^2-test | |
| Sim et al. (2022) | Real-world | 9 | BPIC2012-BPIC2018, Steel processing, port logistics | | Expert review | | x | | | | | | Event log creation | Event data | | one-hot | | | x | | | | | | | | | | | x | | | | | | | | x | | | | | | | Accuracy | split |
| Sindhgatta et al. (2020) | Real-world | 4 | BPIC2013, BPIC2012, BPIC2015, helpdesk | | | x | | | | | | x | Predictive business process monitoring | Event log | | one-hot | | | x | | | | | | x | | | | | x | | | | | | | | x | | | | | | | Accuracy, Attention | split |
| Sommers et al. (2021) | Real-world, Synthetic | 11 | PTAndLogGenera-tor, BPIC | | | x | | x | | | | | Discovery of procedural process model | Event log | | one-hot, trace graph | | | x | | | | | | | | | | | x | | | | | | | | x | | | | | | | F-Score, Simplicity score | |
| Stierle, Weinzierl, et al. (2021) | Real-world | 4 | BPIC2017, BPIC2018, BPIC2020, sp2020 | | Case study | x | | | x | | | | Performance analysis | Event log | | one-hot | | | x | | | | | | | | | | | x | | | | | | | | x | | | | | | | AUC ROC, Sensitivity, Specificity | cross |
| Tavares et al. (2019) | Real-world, Synthetic | 2 | Frozen, Healthcare | | | x | | | | | | x | Descriptive business process monitoring | Event log | time | graph-based | | | | | | | x | | | | x | | | | | | | | | | x | | | | | | | | Graph Distance, Variation | |
| Tax et al. (2016) | Real-world, Synthetic | 2 | Artificial Digital Photocopier, smart home environment | | | x | x | | | | | | Process representation creation and learning | Event log | time, re-source | n-gram | x | | x | | | | | | | | | | | | | | | x | | | | | | x | | | | | Levenshtein similarity | cross |
| Tax et al. (2017) | Real-world | 3 | Helpdesk, BPIC2012, Environmental permit | | | x | | | | | | x | Predictive business process monitoring | Event Log | time | one-hot, index-based | x | | x | | | | | | | | | | | x | | | | | | | | x | | | | | | | MAE, Accuracy | split |
| Taymouri et al. (2020) | Real-world | 3 | Helpdesk, BPIC2012, BPIC2017 | | | x | | | | | | x | Predictive business process monitoring | Event log | | one-hot | | | | x | | | | | x | | | | | x | | | | | | | | x | | | | | | | Weighted average accuracy, weighted average MAE | split |
| Taymouri, La Rosa, and Erfani (2021) | Real-world | 3 | Helpdesk, BPIC2012, BPIC2017 | Paired t-test | | x | | | | | | x | Predictive business process monitoring | Event log | time | one-hot | x | | | x | | | | | | | | | | x | | | | | | | | x | | | | | | | MAE, Average SDL | split |
| Teinemaa et al. (2016) | Real-world | 2 | debt recovery process, the lead-to-contract | | | x | | | | | | x | Predictive business process monitoring | Event log | re-source, data, text | Index-based encoding + text encoding (Bag-of-n-grams model with and without Naive Bayes count ratios, Latent dichlet allocation topic modelling, doc2vec) | | x | x | | | x | | | | | | | x | | | | | | | x | | x | x | x | | | | | F-Score, Precision, Recall, Earliness, Efficiency | cross |

(*continued on next page*)

**Table C.7** (*continued*).

| Paper | Evaluation | | | | | | BPM lifecycle phase | | | | | | BPM task category | Data input | | Feature engineering | | | Model building: ML paradigm | | | | | | ML concept | | | | | | | | | | ML algorithm group | | | | | | | | | | Model assessment | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Data type | Number of datasets | Name of datasets | Statistical method | Further | Implementation available | Process identification | Process discovery | Process analysis | Process redesign | Process implementation | Process monitoring | BPM task category | Dataset type | Rel. context information | Feature extraction | Feature creation | Feature selection | Supervised learning | Semi-supervised learning | Reinforcement learning | Self-supervised learning | Unsupervised learning | Other | Multi-task learning | Active learning | Online/increm. learning | Transfer learning | Ensemble learning | Deep learning | Meta learning | Federated learning | Multi-view learning | Other | Instance-based | Decision tree-based | Clustering-based | Artificial-NN-based | Regression-based | Bayesian-based | Rule-based | Reinf.-learn.-based | Genetic-based | Other | Metrics | Validation strategy |
| Teinemaa et al. (2018) | Real-world | 4 | BPIC2017, RTFM, Unemployment | | | x | | | | | | x | Prescriptive business process monitoring | Event log | | aggregation encoding | | | x | | | | | | | | | | x | | | | | | | x | | | | | | | | | Cost | cross |
| Tello et al. (2019) | Real-world | 1 | SAP | | | | x | | | | | | Event log creation | Event log | time, re-source | event sequences | | | x | | | | x | | | | | | x | | | | | | x | | x | x | | x | | | | | Accuracy, ROC, Testing time | split, cross |
| Theis and Darabi (2019) | Real-world | 3 | BPIC2012, BPIC2013, Helpdesk | | | x | | | | | | x | Predictive business process monitoring | Event log | | discretisation, normalisation | | | x | | | | | | | | | | | x | | | | | | | | x | | | | | | | Accuracy, Precision, Recall, F-Score, Multiclass AUC | split |
| Theis and Darabi (2020) | Synthetic | 1 | Own | T-test, Wilcoxon signed-rank test | | x | | | x | | | | Performance analysis | Other | | | | | | x | | | | | | | | | | x | | | | | | | | x | | | | | | | Generalisation score, Number of generated variants, TP rates, Fitness, Precision | split |
| Turner et al. (2008) | Synthetic | 3 | NA | | | | | x | | | | | Discovery of procedural process model | Event log | | graph-based | | | x | | | | | | | | | | | | | | | x | | | | | | | | | x | | | |
| Valdés et al. (2022a) | Synthetic | 1 | Own | Gamma test | | | | | x | | | | Pattern detection | Time series data | | Trace-fitness-features | | | x | | | | | | | | | | | | | | | x | x | x | | x | | x | x | | | | Precision, Recall, ROC | cross |
| Valdés et al. (2022b) | Synthetic | 1 | Own | | | | | | x | | | | Pattern detection | Time series data | | DFG-based features | | | x | | | | | | | | | | | | | | | x | x | | | | | | | | | | Precision, Recall | cross |
| van der Aalst et al. (2005) | Real-world, Synthetic | 10 | Own | | | | | x | | | | | Discovery of procedural process model | Event log | | | | | x | | | | | | | | | | | | | | | x | | | | | | | | | x | | Ratio how often correct process model was found | |
| Varela-Vaca et al. (2019) | Real-world | 1 | ERP feature model | | | x | | | x | | | | Trace clustering | Event log | data | | | | | | | | x | | | | | | | | | | | x | | | x | | | | | | | | Density, Cyclomatic number, Coefficient of connectivity, Control Flow complexity | |
| Vázquez-Barreiros et al. (2015) | Real-world, Synthetic | 111 | Own | Friedman test, Holm test | | x | | x | | | | | Discovery of procedural process model | Event log | | | | | x | | | | | | | | | | | | | | | x | | | | | | | | | x | | Behavioural precision, Behavioural recall, Structural precision, Structural recall, Completeness, Precision and simplicity | |
| Vázquez-Barreiros et al. (2014) | NA | 21 | NA | | | x | | x | | | | | Discovery of procedural process model | Event log | | | | | x | | | | | | | | | | | | | | | x | | | | | | | | | x | | Behavioural precision, Behavioural recall, Structural precision, Structural recall, Proper completion, Alignment precision, Simplicity | |
| Verenich et al. (2016) | Real-world | 2 | Bondora, Environmental permit | | | x | | | | | | x | Predictive business process monitoring | Event log | time, re-source, data | index-based encoding | | | x | | | | | | | | | | | | | | | x | x | | | | x | | | | | | Accuracy, TP, FP, AUC, ROC | split, cross |
| Vergidis et al. (2007) | Real-world | 5 | Travel agent process | | | | | | | x | | | (Semi-)automatic process redesign | Business process designs (activities, connections, routing decisions) | | | | | x | | | | | | | | | | | | | | | x | | | | | | | | | x | | Cost, Duration, Success ratio | |
| Wahid et al. (2019) | Real-world, Synthetic | 2 | Port logistics process, synthetic log | | | | | | | | | x | Predictive business process monitoring | Event log | Time | entity embedding, continuous variables | x | | x | | | | | | | | | | | x | | | | | | | | x | | | | | | | RMSE, MAE, Accuracy | split |
| Wang et al. (2012) | Real-world, Synthetic | 3 | Artificial, boiler, trans | | | x | | x | | | | | Discovery of procedural process model | Event log, set of process models | | principle components as features | x | x | x | | | | | | | | | | | | | | | x | x | | | | x | | | | | | Training time, Accuracy | split |
| Wang et al. (2019) | Real-world | 6 | BPIC2012, BPIC2017, Sepsis, Production, RTFM, Hospital Billing | | | | | | | | | x | Predictive business process monitoring | Event log | | one-hot, normalisation | | | x | | | | | | | | | | | x | | | | | | | | x | | | | | | | Accuracy, Execution time, Earliness, AUC | split |
| Weinzierl, Dunzer, et al. (2020) | Real-world | 2 | Helpdesk, BPIC2019 | | | x | | | | | | x | Prescriptive business process monitoring | Event log | time | one-hot | x | | x | | | | | | x | | | | | x | | | | | x | | | x | | | | | | | Damerau–Levenshtein Distance, % in Time | split |

(*continued on next page*)

**Table C.7** (*continued*).

| Paper | Evaluation | | | | | BPM lifecycle phase | | | | | | | BPM task category | Data input | | Feature engineering | | | Model building | | | | | | | | | | | | | | | | | | | | | | | | | Model assessment | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | | | | ML paradigm | | | | | | ML concept | | | | | | | | | | ML algorithm group | | | | | | | | | | | |
| | Data type | Number of datasets | Name of datasets | Statistical method | Further | Implementation available | Process identification | Process discovery | Process analysis | Process redesign | Process implementation | Process monitoring | BPM task category | Dataset type | Rel. context information | Feature extraction | Feature creation | Feature selection | Supervised learning | Semi-supervised learning | Reinforcement learning | Self-supervised learning | Unsupervised learning | Other | Multi-task learning | Active learning | Online/increm. learning | Transfer learning | Ensemble learning | Deep learning | Meta learning | Federated learning | Multi-view learning | Other | Instance-based | Decision tree-based | Clustering-based | Artificial-NN-based | Regression-based | Bayesian-based | Rule-based | Reinf.-learn.-based | Genetic-based | Other | Metrics | Validation strategy |
| Weinzierl et al. (2022) | modified Real-world | 4 | BPIC2012, BPIC2013, BPIC2019, BPIC2020 | | Expert interviews | x | | | x | | | | Pattern detection | Event log | re-source, data | binary, index-based | | | x | | | | | | | | | | | x | | | | | | | | x | | | | | | | Accuracy, Precision, Recall, F1-Score | split |
| Wetzstein et al. (2009) | NA | NA | Own | | Simulation | | | | x | | | | Performance analysis | NA | | | | | x | | | | | | | | | | | | | | | x | | | | | | | x | | | | Accuracy | |
| Weytjens and De Weerdt (2021) | Real-world | 3 | BPIC2017, BPIC2019, BPIC2020 | | | x | | | | | | x | Predictive business process monitoring | Event log | time, data | categorical (integer, embedding), numerical (standardised) | x | | x | | | | | | | | | | | x | | | | | | | | x | | x | | | | | MAE, Normalised MAE, Dropout inclusion, Byes inclusion | split |
| Wibisono et al. (2015) | Real-world | 1 | Driver license application | | | | | | | | | x | Dynamic resource allocation | NA | | | | | x | | | | | | | | | | | | | | | x | | | | | | x | | | | | Average completion time (mean, sd, single values) | |
| Wickramanayake et al. (2022) | Real-world | 2 | BPIC2012, BPIC2017 | | Expert review to check in-terpretability of the model(s) | x | | | | | | x | Predictive business process monitoring | Event log | | one-hot | | | x | | | | | | | | | | | x | | | | | | | | x | | | | | | | Accuracy, Precision, Recall, F1-Score | split |
| Xu et al. (2016) | Real-world | 1 | Process management system | | | | | | | | x | | Resource allocation planning | Event log | time, re-source | | | | x | | | | | | | | | | | | | | | x | | | | | | | | | x | | No. of instances scheduled, Ratio of scheduled instances by adjustment, Processing time | |
| Xu and Liu (2019) | Real-world | 3 | BPIC2011, BPIC2018, Hospital Billing | | Case study | | x | | | | | | Data improvement | Event log | | frequency-based | | | | | | | x | | | | | | | | | | | x | x | | x | x | | | | | | | Success rate | |
| Yang et al. (2018) | Real-world, Synthetic | 2 | WABO, BPIC2013 | | | | | x | | | | | Organisational model discovery | Event log | re-source | performer by activity matrix | | | | | | | x | | | | | | | | | | | x | | | x | | | x | | | | | Bcubed precision, Bcubed recall, Bcubed f-measure, Average group size, Average number of membership | |
| Yang et al. (2017) | Real-world | 3 | Tracheal Intubation Data, Trauma Resuscitation Data, Emergency Department Data | Wald test | | | | | x | | | | Trace clustering | Event log | data | activity occurrence per time unit | | x | x | | | | x | | | | | | | | | | | x | | | x | | x | | | | | | (Novel trace) similarity metric, F measure, G-mean | cross |
| Yeshchenko et al. (2019) | Real-world, Synthetic | 3 | BPIC2011, Helpdesk, synthetic log | | | x | | | x | | | | Pattern detection | Event log | | | | | | | | | x | | | | x | | | | | | | | | | x | | | | | | | | F-Score, Erratic measure | |
| Yeshchenko et al. (2018) | Real-world | 4 | BPIC2012, BPIC2017, BPIC2013, Road traffic | | | x | | | | | | x | Predictive business process monitoring | Event log, news | time, re-source, senti-ment | index-based, last-state, day/before/window for sentiments | x | | x | | | | | | | | | | x | | | | | | | x | | | | | | | | | MAE | split |
| Zhao et al. (2016) | Real-world | 1 | E-healthcare process in a Chinese hospital | | | | | | | | | x | Dynamic resource allocation | Event log | re-source | | | | | | | | x | | | | | | x | | | | | | | | x | | | | | | | | MAE, Recommendation Accuracy, Clustering time, Prediction time (compare these metrics with kmeans) | split, cross |
| Zhao et al. (2020) | Real-world | 1 | Resource management system | | | | | | | | | x | Dynamic resource allocation | Event log, resource data | time, re-source, data | key characteristic and team faultiness features | x | x | x | | | | x | | | | | | x | | | | | | | | x | x | | | | | | | MSE, Information value | split |

## References


Afflerbach, P., Hohendorf, M., & Manderscheid, J. (2017). Design it like Darwin – A value-based application of evolutionary algorithms for proper and unambiguous business process redesign. *Information Systems Frontiers*, *19*(5), 1101–1121.

Alpaydin, E. (2014). *Introduction to machine learning* (3rd ed.). MIT Press.

Alves, R. M., Barbieri, L., Stroeh, K., Peres, S. M., & Madeira, E. R. M. (2022). Context-aware completion time prediction for business process monitoring. In *Proceedings of the 2022 World Conference on Information Systems and Technologies* (pp. 355–365). Springer.

Appice, A., Di Mauro, N., & Malerba, D. (2019). Leveraging shallow machine learning to predict business process behavior. In *Proceedings of the 2019 IEEE International Conference on Services Computing* (pp. 184–188). IEEE.

Appice, A., & Malerba, D. (2015). A co-training strategy for multiple view clustering in process mining. *IEEE Transactions on Services Computing*, *9*(6), 832–845.

Bae, H., Lee, S., & Moon, I. (2014). Planning of business process execution in business process management environments. *Information Sciences*, *268*, 357–369.

Baier, L., Reimold, J., & Kühl, N. (2020). Handling concept drift for predictions in business process mining. In *Proceedings of the 22nd IEEE Conference on Business Informatics* (pp. 76–83). IEEE.

Bauer, K., Hinz, O., van der Aalst, W., & Weinhardt, C. (2021). Expl(AI)n it to me – Explainable AI and information systems research. *Business & Information Systems Engineering*, *63*(2), 79–82.

Bazhenova, E., Buelow, S., & Weske, M. (2016). Discovering decision models from event logs. In *Proceedings of the 19th International Conference on Business information systems* (pp. 237–251). Springer.

Beese, J., Haki, M. K., Aier, S., & Winter, R. (2019). Simulation-based research in information systems. *Business & Information Systems Engineering*, *61*(4), 503–521.

Berkenstadt, G., Gal, A., Senderovich, A., Shraga, R., & Weidlich, M. (2020). Queueing inference for process performance analysis with missing life-cycle data. In *Proceedings of the 2nd International Conference on Process Mining* (pp. 57–64). IEEE, http://dx.doi.org/10.1109/ICPM49681.2020.00019.

Bernard, G., & Andritsos, P. (2019). Accurate and transparent path prediction using process mining. In *Proceedings of the 23rd European Conference on Advances in Databases and Information Systems* (pp. 235–250). Springer.

Bevacqua, A., Carnuccio, M., Folino, F., Guarascio, M., & Pontieri, L. (2013). A data-driven prediction framework for analyzing and monitoring business process performances. In *Proceedings of the 15th International Conference on Enterprise Information Systems* (pp. 100–117). Springer.

Beverungen, D., Buijs, J., Becker, J., Di Ciccio, C., van der Aalst, W. M. P., Bartelheimer, C., et al. (2021). Seven paradoxes of business process management in a hyper-connected world. *Business & Information Systems Engineering*, *63*(2), 145–156. http://dx.doi.org/10.1007/s12599-020-00646-z.

Bezerra, F., Wainer, J., & van der Aalst, W. M. P. (2009). Anomaly detection using process mining. In *Proceedings of the 14th International Conference on Enterprise, Business Process and Information Systems Modeling* (pp. 149–161). Springer.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Blum, A. (1998). On-line algorithms in machine learning. In *Online algorithms – State of the art* (pp. 306–325). Springer.

Boltenhagen, M., Chatain, T., & Carmona, J. (2019). Generalized alignment-based trace clustering of process behavior. In *Proceedings of the 40th International Conference on Application and Theory of Petri Nets and Concurrency* (pp. 237–257). Springer.

Borkowski, M., Fdhila, W., Nardelli, M., Rinderle-Ma, S., & Schulte, S. (2019). Event-based failure prediction in distributed business processes. *Information Systems*, *81*, 220–235.

Bose, R. J. C., Mans, R. S., & van der Aalst, W. M. P. (2013). Wanna improve process mining results? In *Proceedings of the 2013 IEEE Symposium on Computational Intelligence and Data Mining* (pp. 127–134). IEEE.

Bose, R. J. C., & van der Aalst, W. M. P. (2009). Context aware trace clustering: Towards improving process mining results. In *Proceedings of the 2009 SIAM International Conference on Data Mining* (pp. 401–412). SIAM.

Bozorgi, Z. D., Teinemaa, I., Dumas, M., La Rosa, M., & Polyvyanyy, A. (2020). Process mining meets causal machine learning: Discovering causal rules from event logs. In *Proceedings of the 2nd International Conference on Process Mining* (pp. 129–136). IEEE.

Bozorgi, Z. D., Teinemaa, I., Dumas, M., La Rosa, M., & Polyvyanyy, A. (2021). Prescriptive process monitoring for cost-aware cycle time reduction. In *Proceedings of the 3rd International Conference on Process Mining* (pp. 96–103). IEEE.

Breiman, L., Friedman, J. H., Olshen, R., & Stone, C. (1984). *Classification and Regression Trees*. Wadsworth.

Breuker, D., Matzner, M., Delfmann, P., & Becker, J. (2016). Comprehensible predictive models for business processes. *MIS Quarterly*, *40*(4), 1009–1034. http://dx.doi.org/10.25300/MISQ/2016/40.4.10.

Brunk, J., Stierle, M., Papke, L., Revoredo, K., Matzner, M., & Becker, J. (2021). Cause vs. effect in context-sensitive prediction of business process instances. *Information Systems*, *95*, Article 101635.

Brynjolfsson, E., & McAfee, A. (2017). The business of artificial intelligence. URL https://hbr.org/2017/07/the-business-of-artificial-intelligence.

Buijs, J., van Dongen, B. F., & van der Aalst, W. M. P. (2012). A genetic algorithm for discovering process trees. In *Proceedings of the 2012 IEEE Congress on Evolutionary Computation* (pp. 1–8). IEEE.

Buijs, J., van Dongen, B. F., & van der Aalst, W. M. P. (2013). Mining configurable process models from collections of event logs. In *Proceedings of the 11th International Conference on Business Process Management* (pp. 33–48). Springer.

Camargo, M., Dumas, M., & González-Rojas, O. (2019). Learning accurate LSTM models of business processes. In *Proceedings of the 17th International Conference on Business Process management* (pp. 286–302). Springer.

Camargo, M., Dumas, M., & González-Rojas, O. (2022). Learning accurate business process simulation models from event logs via automated process discovery and deep learning. In *Proceedings of the 34th International Conference on Advanced Information Systems Engineering* (pp. 55–71). Springer.

Cao, F., Estert, M., Qian, W., & Zhou, A. (2006). Density-based clustering over an evolving data stream with noise. In *Proceedings of the 2006 SIAM International Conference on Data Mining* (pp. 328–339). SIAM.

Ceravolo, P., Damiani, E., Schepis, E. F., & Tavares, G. M. (2022). Real-time probing of control-flow and data-flow in event logs. *Procedia Computer Science*, *197*, 751–758.

Chesani, F., Lamma, E., Mello, P., Montali, M., Riguzzi, F., & Storari, S. (2009). Exploiting inductive logic programming techniques for declarative process mining. *Transactions on Petri Nets and Other Models of Concurrency II: Special Issue on Concurrency in Process-Aware Information Systems*, 278–295.

Ciasullo, M. V., Fenza, G., Loia, V., Orciuoli, F., Troisi, O., & Herrera-Viedma, E. (2018). Business process outsourcing enhanced by fuzzy linguistic consensus model. *Applied Soft Computing*, *64*, 436–444.

Cohen, W. W. (1995). Fast effective rule induction. In *Proceedings of the 12th International Conference on Machine Learning* (pp. 115–123). Elsevier, http://dx.doi.org/10.1016/B978-1-55860-377-6.50023-2.

Conforti, R., de Leoni, M., La Rosa, M., van der Aalst, W. M. P., & ter Hofstede, A. H. (2015). A recommendation system for predicting risks across multiple business process instances. *Decision Support Systems*, *69*, 1–19.

Cooper, H. M. (1988). Organizing knowledge syntheses: A taxonomy of literature reviews. *Knowledge in Society*, *1*(1), 104–126.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297. http://dx.doi.org/10.1007/BF00994018.

Cuzzocrea, A., Folino, F., Guarascio, M., & Pontieri, L. (2015). A multi-view learning approach to the discovery of deviant process instances. In *Proceedings of the 2015 OTM Confederated International Conferences "On The Move to Meaningful Internet Systems"* (pp. 146–165). Springer.

Cuzzocrea, A., Folino, F., Guarascio, M., & Pontieri, L. (2016a). A multi-view multi-dimensional ensemble learning approach to mining business process deviances. In *Proceedings of the 2016 International Joint Conference on Neural Networks* (pp. 3809–3816). IEEE.

Cuzzocrea, A., Folino, F., Guarascio, M., & Pontieri, L. (2016b). A robust and versatile multi-view learning framework for the detection of deviant business process instances. *International Journal of Cooperative Information Systems*, *25*(4), Article 1740003.

Cuzzocrea, A., Folino, F., Guarascio, M., & Pontieri, L. (2019). Predictive monitoring of temporally-aggregated performance indicators of business processes against low-level streaming events. *Information Systems*, *81*, 236–266.

De Jong, K. A., Spears, W. M., & Gordon, D. F. (1993). Using genetic algorithms for concept learning. *Machine Learning*, *13*(2), 161–188.

De Koninck, P., & De Weerdt, J. (2019). Scalable mixed-paradigm trace clustering using super-instances. In *Proceedings of the 1st International Conference on Process Mining* (pp. 17–24). IEEE.

De Koninck, P., De Weerdt, J., & vanden Broucke, S. (2017). Explaining clusterings of process instances. *Data Mining and Knowledge Discovery*, *31*(3), 774–808.

De Koninck, P., Nelissen, K., Baesens, B., Snoeck, M., & De Weerdt, J. (2017). An approach for incorporating expert knowledge in trace clustering. In *Proceedings of the 29th International Conference on Advanced Information Systems Engineering* (pp. 561–576). Springer.

De Koninck, P., Nelissen, K., Baesens, B., Snoeck, M., & De Weerdt, J. (2021). Expert-driven trace clustering with instance-level constraints. *Knowledge and Information Systems*, *63*(5), 1197–1220.

De Koninck, P., vanden Broucke, S., & De Weerdt, J. (2018). Act2vec, trace2vec, log2vec, and model2vec: Representation learning for business processes. In *Proceedings of the 16th International Conference on Business Process Management* (pp. 305–321). Springer.

de Leoni, M., Dees, M., & Reulink, L. (2020). Design and evaluation of a process-aware recommender system based on prescriptive analytics. In *Proceedings of the 2nd International Conference on Process Mining* (pp. 9–16). IEEE.

de Leoni, M., Dumas, M., & García-Bañuelos, L. (2013). Discovering branching conditions from business process execution logs. In *Proceedings of the 16th International Conference on Fundamental Approaches to Software Engineering* (pp. 114–129). Springer.

de Leoni, M., van der Aalst, W. M. P., & Dees, M. (2014). A general framework for correlating business process characteristics. In *Proceedings of the 12th International Conference on Business Process Management* (pp. 250–266). Springer.

de Leoni, M., van der Aalst, W. M. P., & Dees, M. (2016). A general process mining framework for correlating, predicting and clustering dynamic behavior based on event logs. *Information Systems*, *56*, 235–257.

De Maio, C., Fenza, G., Loia, V., Orciuoli, F., & Herrera-Viedma, E. (2016). A framework for context-aware heterogeneous group decision making in business processes. *Knowledge-Based Systems*, *102*, 39–50.

De Morais, R. M., & Kazan, S. (2014). An analysis of BPM lifecycles: From a literature review to a framework proposal. *Business Process Management Journal*, *20*, 1463–7154. http://dx.doi.org/10.1108/BPMJ-03-2013-0035.

De Weerdt, J., vanden Broucke, S., Vanthienen, J., & Baesens, B. (2012). Leveraging process discovery with trace clustering and text mining for intelligent analysis of incident management processes. In *Proceedings of the 2012 IEEE Congress on Evolutionary Computation* (pp. 1–8). IEEE.

De Weerdt, J., vanden Broucke, S., Vanthienen, J., & Baesens, B. (2013). Active trace clustering for improved process discovery. *IEEE Transactions on Knowledge and Data Engineering*, *25*(12), 2708–2720.

Delcoucq, L., Dupiereux-Fettweis, T., Lecron, F., & Fortemps, P. (2022). Resource and activity clustering based on a hierarchical cell formation algorithm. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, 1–10.

Delias, P., Doumpos, M., Grigoroudis, E., & Matsatsinis, N. (2019). A non-compensatory approach for trace clustering. *International Transactions in Operational Research*, *26*(5), 1828–1846.

Di Francescomarino, C., Dumas, M., Maggi, F. M., & Teinemaa, I. (2016). Clustering-based predictive process monitoring. *IEEE Transactions on Services Computing*, *12*(6), 896–909.

Di Francescomarino, C., Ghidini, C., Maggi, F. M., & Milani, F. (2018). Predictive process monitoring methods: Which one suits me best? In *Proceedings of 16th International Conference on Business Process Management* (pp. 462–479). Springer.

Di Francescomarino, C., Ghidini, C., Maggi, F. M., Petrucci, G., & Yeshchenko, A. (2017). An eye into the future: Leveraging a-priori knowledge in predictive business process monitoring. In *Proceedings of the 15th International Conference on Business Process Management* (pp. 252–268). Springer.

Di Mauro, N., Appice, A., & Basile, T. (2019). Activity prediction of business process instances with inception cnn models. In *Proceedings of the 18th International Conference of the Italian Association for Artificial Intelligence* (pp. 348–361). Springer.

Diamantini, C., Genga, L., & Potena, D. (2016). Behavioral process mining for unstructured processes. *Journal of Intelligent Information Systems*, *47*(1), 5–32.

Dietterich, T. G. (2002). Ensemble learning. In *The handbook of brain theory and neural networks* (2nd ed.). (pp. 110–125). MIT Press.

Dumas, M., La Rosa, M., Mendling, J., & Reijers, H. A. (2018). Fundamentals of business process management, (2nd ed.). Springer, http://dx.doi.org/10.1007/978-3-642-33143-5.

Effendi, Y. A., & Sarno, R. (2017). Discovering process model from event logs by considering overlapping rules. In *Proceedings of the 4th International Conference on Electrical Engineering, Computer Science and Informatics* (pp. 1–6). IEEE.

Es-Soufi, W., Yahia, E., & Roucoules, L. (2016). On the use of process mining and machine learning to support decision making in systems design. In *Proceedings of the 13th IFIP International Conference on Product Lifecycle Management* (pp. 56–66). Springer, http://dx.doi.org/10.1007/978-3-319-54660-5_6.

Evermann, J., Rehse, J.-R., & Fettke, P. (2017). Predicting process behaviour using deep learning. *Decision Support Systems*, *100*, 129–140. http://dx.doi.org/10.1016/j.dss.2017.04.003.

Ferreira, D. R., & Gillblad, D. (2009). Discovering process models from unlabelled event logs. In *Proceedings of the 7th International Conference on Business process Management* (pp. 143–158). Springer.

Ferreira, D. R., Szimanski, F., & Ralha, C. G. (2013). Mining the low-level behaviour of agents in high-level business processes. *International Journal of Business Process Integration and Management*, *6*(2), 146–166.

Ferreira, D. R., & Vasilyev, E. (2015). Using logical decision trees to discover the cause of process delays from event logs. *Computers in Industry*, *70*, 194–207.

Ferreira, D. R., Zacarias, M., Malheiros, M., & Ferreira, P. (2007). Approaching process mining with sequence clustering: Experiments and findings. In *Proceedings of the 5th International Conference on Business Process Management* (pp. 360–374). Springer.

Ferri, C., Hernández-Orallo, J., & Modroiu, R. (2009). An experimental comparison of performance measures for classification. *Pattern Recognition Letters*, *30*(1), 27–38.

Feuerriegel, S., Dolata, M., & Schwabe, G. (2020). Fair AI. *Business & Information Systems Engineering*, *62*(4), 379–384.

Feuerriegel, S., Hartmann, J., Janiesch, C., & Zschech, P. (2023). Generative AI. *Business & Information Systems Engineering*, in press.

Firouzian, I., Zahedi, M., & Hassanpour, H. (2019a). Cycle time optimization of processes using an entropy-based learning for task allocation. *International Journal of Engineering*, *32*(8), 1090–1100.

Firouzian, I., Zahedi, M., & Hassanpour, H. (2019b). Investigation of the effect of concept drift on data-aware remaining time prediction of business processes. *International Journal of Nonlinear Analysis and Applications*, *10*(2), 153–166.

Fleiss, J. L., & Cohen, J. (1973). The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and Psychological Measurement*, *33*(3), 613–619.

Folino, F., Folino, G., Guarascio, M., & Pontieri, L. (2022). Semi-supervised discovery of DNN-based outcome predictors from scarcely-labeled process logs. *Business & Information Systems Engineering*, 1–21.

Folino, F., Greco, G., Guzzo, A., & Pontieri, L. (2011). Mining usage scenarios in business processes: Outlier-aware discovery and run-time prediction. *Data & Knowledge Engineering*, *70*(12), 1005–1029.

Folino, F., Guarascio, M., & Pontieri, L. (2012). Discovering context-aware models for predicting business process performances. In *Proceedings of the 2012 OTM Confederated International Conferences "On The Move to Meaningful Internet Systems"* (pp. 287–304). Springer.

Folino, F., Guarascio, M., & Pontieri, L. (2015). Mining multi-variant process models from low-level logs. In *Proceedings of the 18th International Conference on Business Information Systems* (pp. 165–177). Springer.

Folino, F., & Pontieri, L. (2021). AI-empowered process mining for complex application scenarios: Survey and discussion. *Journal on Data Semantics*, *10*, 77–106. http://dx.doi.org/10.1007/s13740-021-00121-2.

Forgy, E. W. (1965). Cluster analysis of multivariate data: Efficiency versus interpretability of classifications. *Biometrics*, *21*(3), 768–769.

Fürnkranz, J., Gamberger, D., & Lavrač, N. (2012). *Foundations of rule learning*. Springer.

Galanti, R., Coma-Puig, B., de Leoni, M., Carmona, J., & Navarin, N. (2020). Explainable predictive process monitoring. In *Proceedings of the 2nd International Conference on Process Mining* (pp. 1–8). IEEE.

Garcez, A. d., Bader, S., Bowman, H., Lamb, L. C., de Penning, L., Illuminoo, B., et al. (2022). Neural-symbolic learning and reasoning: A survey and interpretation. In *Neuro-symbolic artificial intelligence: The state of the art* (pp. 1–51). IOS Press.

Garcez, A. d., Besold, T. R., De Raedt, L., Földiak, P., Hitzler, P., Icard, T., et al. (2015). Neural-symbolic learning and reasoning: Contributions and challenges. In *Proceedings of the 2015 AAAI Spring Symposium – Knowledge representation and reasoning: Integrating symbolic and neural approaches* (pp. 18–21). AI Access Foundation.

García-Bañuelos, L., Dumas, M., La Rosa, M., De Weerdt, J., & Ekanayake, C. C. (2014). Controlled automated discovery of collections of business process models. *Information Systems*, *46*, 85–101.

Gerlach, Y., Seeliger, A., Nolle, T., & Mühlhäuser, M. (2022). Inferring a multi-perspective likelihood graph from black-box next event predictors. In *Proceedings of the 34th International Conference on Advanced Information Systems Engineering* (pp. 19–35). Springer.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Greco, G., Guzzo, A., Pontieri, L., & Sacca, D. (2004). Mining expressive process models by clustering workflow traces. In *Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining* (pp. 52–62). Springer.

Greco, G., Guzzo, A., Pontieri, L., & Sacca, D. (2006). Discovering expressive process models by clustering log traces. *IEEE Transactions on Knowledge and Data Engineering*, *18*(8), 1010–1027.

Grigori, D., Casati, F., Castellanos, M., Dayal, U., Sayal, M., & Shan, M.-C. (2004). Business process intelligence. *Computers in Industry*, *53*(3), 321–343. http://dx.doi.org/10.1016/j.compind.2003.10.007.

Guo, B., Zhang, X., Wang, Z., Jiang, M., Nie, J., Ding, Y., et al. (2023). How close is chatgpt to human experts? comparison corpus, evaluation, and detection. arXiv preprint arXiv:2301.07597, arXiv:2301.07597.

Guzzo, A., Joaristi, M., Rullo, A., & Serra, E. (2021). A multi-perspective approach for the analysis of complex business processes behavior. *Expert Systems with Applications*, *177*, Article 114934.

Ha, Q.-T., Bui, H.-N., & Nguyen, T.-T. (2016). A trace clustering solution based on using the distance graph model. In *Proceedings of the 8th International Conference on Computational Collective Intelligence* (pp. 313–322). Springer.

Hanga, K. M., Kovalchuk, Y., & Gaber, M. M. (2020). A graph-based approach to interpreting recurrent neural networks in process mining. *IEEE Access*, *8*, 172923–172938.

Harl, M., Weinzierl, S., Stierle, M., & Matzner, M. (2020). Explainable predictive business process monitoring using gated graph neural networks. *Journal of Decision Systems*, *29*(sup1), 312–327.

Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). *The elements of statistical learning: Data mining, inference, and prediction*: *vol. 2*, Springer.

Hayes-Roth, F. (1985). Rule-based systems. *Communications of the ACM*, *28*(9), 921–932.

Heinrich, K., Zschech, P., Janiesch, C., & Bonin, M. (2021). Process data properties matter: Introducing gated convolutional neural networks (GCNN) and key-value-predict attention networks (KVP) for next event prediction with deep learning. *Decision Support Systems*, *143*, Article 113494. http://dx.doi.org/10.1016/j.dss.2021.113494.

Herbst, J. (2000). A machine learning approach to workflow management. In *Proceedings of the 11th European Conference on Machine Learning* (pp. 183–194). Springer.

Herbst, J., & Karagiannis, D. (2000). Integrating machine learning and workflow management to support acquisition and adaptation of workflow models. *Intelligent Systems in Accounting, Finance & Management*, *9*(2), 67–92.

Herbst, J., & Karagiannis, D. (2004). Workflow mining with InWoLvE. *Computers in Industry*, *53*(3), 245–264.

Herm, L.-V., Janiesch, C., Reijers, H. A., & Seubert, F. (2021). From symbolic RPA to intelligent RPA: Challenges for developing and operating intelligent software robots. In *Proceedings of the 19th International Conference on Business Process Management* (pp. 289–305). Springer.

Houy, C., Fettke, P., & Loos, P. (2010). Empirical research in business process management — Analysis of an emerging field of research. *Business Process Management Journal*, *16*, 619–661. http://dx.doi.org/10.1108/14637151011065946.

Hruschka, D. J., Schwartz, D., St. John, D. C., Picone-Decaro, E., Jenkins, R. A., & Carey, J. W. (2004). Reliability in coding open-ended data: Lessons learned from HIV behavioral research. *Field Methods*, *16*(3), 307–331.

Hsieh, C., Moreira, C., & Ouyang, C. (2021). DICE4EL: Interpreting process predictions using a milestone-aware counterfactual approach. In *Proceedings of the 3rd International Conference on Process Mining* (pp. 88–95). IEEE.

Huang, Z., van der Aalst, W. M. P., Lu, X., & Duan, H. (2010). An adaptive work distribution mechanism based on reinforcement learning. *Expert Systems with Applications*, *37*(12), 7533–7541.

Huang, Z., van der Aalst, W. M. P., Lu, X., & Duan, H. (2011). Reinforcement learning based resource allocation in business process management. *Data & Knowledge Engineering*, *70*(1), 127–145.

Huo, S., Völzer, H., Reddy, P., Agarwal, P., Isahagian, V., & Muthusamy, V. (2021). Graph autoencoders for business process anomaly detection. In *Proceedings of the 19th International Conference on Business Process Management* (pp. 417–433). Springer.

Jaiswal, A., Babu, A. R., Zadeh, M. Z., Banerjee, D., & Makedon, F. (2021). A survey on contrastive self-supervised learning. *Technologies*, *9*(2), 1–22. http://dx.doi.org/10.3390/technologies9010002.

Jalayer, A., Kahani, M., Beheshti, A., Pourmasoumi, A., & Motahari-Nezhad, H. R. (2020). Attention mechanism in predictive business process monitoring. In *Proceedings of the 24th IEEE international Enterprise Distributed Object Computing Conference* (pp. 181–186). IEEE.

Janiesch, C., Zschech, P., & Heinrich, K. (2021). Machine learning and deep learning. *Electronic Markets*, *31*, 685–695. http://dx.doi.org/10.1007/s12525-021-00475-2.

Jing, L., & Tian, Y. (2021). Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(11), 4037–4058. http://dx.doi.org/10.1109/TPAMI.2020.2992393.

Jlailaty, D., Grigori, D., & Belhajjame, K. (2017). Business process instances discovery from email logs. In *Proceeding of the 14th International Conference on Services Computing* (pp. 19–26). IEEE.

Jobin, A., Ienca, M., & Vayena, E. (2019). The global landscape of AI ethics guidelines. *Nature Machine Intelligence*, *1*(9), 389–399.

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, *349*(6245), 255–260.

Jung, J.-Y., Bae, J., & Liu, L. (2008). Hierarchical business process clustering. In *Proceedings of the 2008 IEEE International Conference on Services Computing* (pp. 613–616). IEEE.

Jung, J.-Y., Bae, J., & Liu, L. (2009). Hierarchical clustering of business process models. *International Journal of Innovative Computing, Information and Control*, *5*(12), 1349–4198.

Junior, S. B., Ceravolo, P., Damiani, E., Omori, N. J., & Tavares, G. M. (2020). Anomaly detection on event logs with a scarcity of labels. In *Proceedings of the 2nd International Conference on Process Mining* (pp. 161–168). IEEE.

Junior, S. B., Tavares, G. M., da Costa, V. G. T., Ceravolo, P., & Damiani, E. (2018). A framework for human-in-the-loop monitoring of concept-drift detection in event log stream. In *Proceedings of the Web Conference 2018* (pp. 319–326).

Kaddour, J., Lynch, A., Liu, Q., Kusner, M. J., & Silva, R. (2022). Causal machine learning: A survey and open problems. (pp. 1–188). arXiv preprint arXiv:2206.15475.

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, *4*(19), 237–285. http://dx.doi.org/10.1613/jair.301.

Kang, B., Kim, D., & Kang, S.-H. (2012). Periodic performance prediction for real-time business process monitoring. *Industrial Management & Data Systems*, *112*(1), 4–23.

Kazakov, O. D., Novikov, S. P., Kulagina, N. A., & Shlapakova, S. N. (2018). Development of the concept of management of economic systems processes through construction and calling of machine learning models. In *Proceedings of the 2018 IEEE International Conference "Quality Management, Transport and Information Security, Information Technologies"* (pp. 316–321). IEEE.

Kazim, E., & Koshiyama, A. S. (2021). A high-level overview of AI ethics. *Patterns*, *2*(9), Article 100314.

Khan, A., Ghose, A., & Dam, H. (2021). Decision support for knowledge intensive processes using RL based recommendations. In *Proceedings of the 19th International Conference of Business Process Management (Forum)* (pp. 246–262). Springer.

Khodyrev, I., & Popova, S. (2014). Discrete modeling and simulation of business processes using event logs. *Procedia Computer Science*, *29*, 322–331.

Kim, J., Suh, W., & Lee, H. (2002). Document-based workflow modeling: A case-based reasoning approach. *Expert Systems with Applications*, *23*(2), 77–93.

Klijn, E. L., & Fahland, D. (2020). Identifying and reducing errors in remaining time prediction due to inter-case dynamics. In *Proceedings of the 2nd International Conference on Process Mining* (pp. 25–32). IEEE.

Ko, J., & Comuzzi, M. (2023). A systematic review of anomaly detection for business process event logs. *Business & Information Systems Engineering*.

Kotsiantis, S. B., Zaharakis, I. D., & Pintelas, P. E. (2006). Machine learning: A review of classification and combining techniques. *Artificial Intelligence Review*, *26*(3), 159–190. http://dx.doi.org/10.1007/s10462-007-9052-3.

Krajsic, P., & Franczyk, B. (2020). Lambda architecture for anomaly detection in online process mining using autoencoders. In *Proceeding of the 12th International Conference on Computational Collective Intelligence* (pp. 579–589). Springer.

Krajsic, P., & Franczyk, B. (2021). Variational autoencoder for anomaly detection in event data in online process mining. In *Proceedings of the 23rd International Conferences on Enterprise Information Systems* (pp. 567–574). SciTePress.

Kratsch, W., König, F., & Röglinger, M. (2022). Shedding light on blind spots – Developing a reference architecture to leverage video data for process mining. *Decision Support Systems*, *158*, Article 113794.

Krippendorff, K. (2018). *Content analysis: An introduction to its methodology*. Sage Publications.

Lakshmanan, G. T., Shamsi, D., Doganata, Y. N., Unuvar, M., & Khalaf, R. (2015). A markov prediction model for data-driven semi-structured business processes. *Knowledge and Information Systems*, *42*(1), 97–126.

Lamma, E., Mello, P., Montali, M., Riguzzi, F., & Storari, S. (2007). Inducing declarative logic-based models from labeled traces. In *Proceedings of the 5th International Conference on Business Process Management* (pp. 344–359). Springer.

Lamma, E., Mello, P., Riguzzi, F., & Storari, S. (2007). Applying inductive logic programming to process mining. In *Proceedings of the 17th International Conference on Inductive Logic Programming* (pp. 132–146). Springer.

Landis, J. R., & Koch, G. G. (1977). An application of hierarchical kappa-type statistics in the assessment of majority agreement among multiple observers. *Biometrics*, 363–374.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. http://dx.doi.org/10.1038/nature14539.

Lee, W. L. J., Burattin, A., Munoz-Gama, J., & Sepulveda, M. (2021). Orientation and conformance: A HMM-based approach to online conformance checking. *Information Systems*, *102*, Article 101674. http://dx.doi.org/10.1016/j.is.2020.101674.

Lee, S., Lu, X., & Reijers, H. A. (2022). The analysis of online event streams: Predicting the next activity for anomaly detection. In *Proceeding of the 16th International Conference on Research Challenges in Information Science* (pp. 248–264). Springer.

Lee, W. L. J., Parra, D., Munoz-Gama, J., & Sepulveda, M. (2018). Predicting process behavior meets factorization machines. *Expert Systems with Applications*, *112*, 87–98.

Leno, V., Dumas, M., & Maggi, F. M. (2018). Correlating activation and target conditions in data-aware declarative process discovery. In *Proceedings of the 16th International Conference on Business Process Management* (pp. 176–193). Springer.

Leno, V., Dumas, M., Maggi, F. M., La Rosa, M., & Polyvyanyy, A. (2020). Automated discovery of declarative process models with correlated data conditions. *Information Systems*, *89*, Article 101482.

Leontjeva, A., Conforti, R., Di Francescomarino, C., Dumas, M., & Maggi, F. M. (2016). Complex symbolic sequence encodings for predictive monitoring of business processes. In *Proceedings of the 14th International Conference on Business Process Management* (pp. 297–313). Springer.

Lewis, D. D. (1998). Naive (Bayes) at forty: The independence assumption in information retrieval. In *Proceedings of the 10th European Conference on Machine Learning* (pp. 4–15). Springer.

Li, G., de Murillas, E. G. L., de Carvalho, R. M., & van der Aalst, W. M. P. (2018). Extracting object-centric event logs to support process mining on databases. In *Proceedings of the 30th International Conference on Advanced Information Systems Engineering (Forum)* (pp. 182–199). Springer.

Li, C., Reichert, M., & Wombacher, A. (2010). The MinAdept clustering approach for discovering reference process models out of process variants. *International Journal of Cooperative Information Systems*, *19*(03n04), 159–203.

Liu, T., Cheng, Y., & Ni, Z. (2012). Mining event logs to support workflow resource allocation. *Knowledge-Based Systems*, *35*, 320–331.

Liu, H., & Motoda, H. (1998). *Feature extraction, construction and selection: A Data mining perspective*: *vol. 453*, Springer.

Liu, Y., Wang, J., Yang, Y., & Sun, J. (2008). A semi-automatic approach for workflow staff assignment. *Computers in Industry*, *59*(5), 463–476.

Lombard, M., Snyder-Duch, J., & Bracken, C. C. (2006). Content analysis in mass communication: Assessment and reporting of intercoder reliability. *Human Communication Research*, *28*(4), 587–604. http://dx.doi.org/10.1111/j.1468-2958.2002.tb00826.x, arXiv:https://academic.oup.com/hcr/article-pdf/28/4/587/22337849/jhumcom0587.pdf.

López, H. A., Strømsted, R., Niyodusenga, J.-M., & Marquard, M. (2021). Declarative process discovery: Linking process and textual views. In *Proceedings of the 33rd International Conference on Advanced Information Systems Engineering (Forum)* (pp. 109–117). Springer.

Lu, F., Zeng, Q., & Duan, H. (2016). Synchronization-core-based discovery of processes with decomposable cyclic dependencies. *ACM Transactions on Knowledge Discovery from Data*, *10*(3), 1–29.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Proceedings of the 31st Conference on Neural Information Processing Systems* (pp. 4765–4774). ACM.

Maggi, F. M., Bose, R. J., & van der Aalst, W. M. P. (2012). Efficient discovery of understandable declarative process models from event logs. In *Proceedings of the 24th International Conference on Advanced Information Systems Engineering* (pp. 270–285). Springer.

Maggi, F. M., Di Ciccio, C., Di Francescomarino, C., & Kala, T. (2018). Parallel algorithms for the automated discovery of declarative process models. *Information Systems*, *74*, 136–152.

Maggi, F. M., Di Francescomarino, C., Dumas, M., & Ghidini, C. (2014). Predictive monitoring of business processes. In *Proceedings of the 26th International Conference on Advanced Information Systems Engineering* (pp. 457–472). Springer.

Maier, A. K., Syben, C., Stimpel, B., Würfl, T., Hoffmann, M., Schebesch, F., et al. (2019). Learning with known operators reduces maximum error bounds. *Nature Machine Intelligence*, *1*(8), 373–380.

Maita, A. R. C., Martins, L. C., Paz, C. R. L., Peres, S. M., & Fantinato, M. (2015). Process mining through artificial neural networks and support vector machines: A systematic literature review. *Business Process Management Journal*, *21*(6), 1391–1415.

Mannhardt, F., de Leoni, M., Reijers, H. A., & van der Aalst, W. M. P. (2016). Decision mining revisited – discovering overlapping rules. In *Proceedings of the 28th International Conference on Advanced Information Systems Engineering* (pp. 377–392). Springer.

Márquez-Chamorro, A. E., Resinas, M., & Ruiz-Cortés, A. (2017). Predictive monitoring of business processes: A survey. *IEEE Transactions on Services Computing*, *11*(6), 962–977. http://dx.doi.org/10.1109/TSC.2017.2772256.

Márquez-Chamorro, A. E., Resinas, M., Ruiz-Cortés, A., & Toro, M. (2017). Run-time prediction of business process indicators using evolutionary decision rules. *Expert Systems with Applications*, *87*, 1–14.

Martens, A., Fettke, P., & Loos, P. (2014). A genetic algorithm for the inductive derivation of reference models using minimal graph-edit distance applied to real-world business process data. In *Proceedings of the 2014 Multikonferenz Wirtschaftsinformatik* (pp. 1613–1626). Universität Paderborn.

Măruşter, L., Weijters, A. J., van der Aalst, W. M. P., & Van Den Bosch, A. (2002). Process mining: Discovering direct successors in process logs. In *Proceedings of the 5th International Conference on Discovery Science* (pp. 364–373). Springer.

Măruşter, L., Weijters, A. J., van der Aalst, W. M. P., & Van Den Bosch, A. (2006). A rule-based approach for process discovery: Dealing with noise and imbalance in process logs. *Data Mining and Knowledge Discovery*, *13*(1), 67–87.

McCoy, D. W. (2002). Business activity monitoring: Calm before the storm. URL https://www.gartner.com/resources/105500/105562/105562.pdf.

Mehdiyev, N., Evermann, J., & Fettke, P. (2020). A novel business process prediction model using a deep learning method. *Business & Information Systems Engineering*, *62*(2), 143–157. http://dx.doi.org/10.1007/s12599-018-0551-3.

Mendling, J., Decker, G., Hull, R., Reijers, H. A., & Weber, I. (2018). How do machine learning, robotic process automation, and blockchains affect the human factor in business process management? *Communications of the Association for Information Systems*, *43*(1), 297–320. http://dx.doi.org/10.17705/1CAIS.04319.

Metzger, A., Kley, T., & Palm, A. (2020). Triggering proactive business process adaptations via online reinforcement learning. In *Proceedings of the 18th International Conference on Business Process Management* (pp. 273–290). Springer.

Metzger, A., Neubauer, A., Bohn, P., & Pohl, K. (2019). Proactive process adaptation using deep learning ensembles. In *Proceedings of the 31st International Conference on Advanced Information Systems Engineering* (pp. 547–562). Springer.

Mitchell, T. M. (1997). *Machine learning*. McGraw-Hill Education.

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). *Foundations of machine learning*. MIT Press.

Montani, S., & Leonardi, G. (2012). Retrieval and clustering for business process monitoring: Results and improvements. In *Proceedings of the 20th International Conference on Case-based Reasoning* (pp. 269–283). Springer.

Mothilal, R. K., Sharma, A., & Tan, C. (2020). Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency* (pp. 607–617). ACM.

zur Muehlen, M., & Ho, D. T.-Y. (2005). Risk management in the BPM lifecycle. In *Proceedings of the 3rd International Conference on Business Process Management (Workshops)* (pp. 454–466). Springer, http://dx.doi.org/10.1007/11678564_42.

Murphy, K. P. (2012). *Machine learning: A probabilistic perspective*. MIT Press.

Mustansir, A., Shahzad, K., & Malik, M. K. (2022). Towards automatic business process redesign: An NLP based approach to extract redesign suggestions. *Automated Software Engineering*, *29*(12), 1–24. http://dx.doi.org/10.1007/s10515-021-00316-8.

Neu, D. A., Lahann, J., & Fettke, P. (2021). A systematic literature review on state-of-the-art deep learning methods for process prediction. *Artificial Intelligence Review*, *55*, 801–827. http://dx.doi.org/10.1007/s10462-021-09960-8.

Neuendorf, K. A. (2017). *The content analysis guidebook*. Sage Publications.

Nguyen, H. T. C., Lee, S., Kim, J., Ko, J., & Comuzzi, M. (2019). Autoencoders for improving quality of process event logs. *Expert Systems with Applications*, *131*, 132–147. http://dx.doi.org/10.1016/j.eswa.2019.04.052.

Nguyen, P., Slominski, A., Muthusamy, V., Ishakian, V., & Nahrstedt, K. (2016). Process trace clustering: A heterogeneous information network approach. In *Proceedings of the 2016 SIAM International Conference on Data Mining* (pp. 279–287). SIAM.

Nichol, A., Achiam, J., & Schulman, J. (2018). On first-order meta-learning algorithms. (pp. 1–15). arXiv preprint arXiv:1803.02999, arXiv:1803.02999.

Nigam, A., & Caswell, N. S. (2003). Business artifacts: An approach to operational specification. *IBM Systems Journal*, *42*(3), 428–445.

Nolle, T., Luettgen, S., Seeliger, A., & Mühlhäuser, M. (2018). Analyzing business process anomalies using autoencoders. *Machine Learning*, *107*(11), 1875–1893. http://dx.doi.org/10.1007/s10994-018-5702-8.

Nolle, T., Luettgen, S., Seeliger, A., & Mühlhäuser, M. (2019). BINet: Multi-perspective business process anomaly classification. *Information Systems*, *103*, Article 101458. http://dx.doi.org/10.1016/j.is.2019.101458.

Nolle, T., Seeliger, A., & Mühlhäuser, M. (2018). BINet: Multivariate business process anomaly detection using deep learning. In *Proceedings of the 16th International Conference on Business Process Management* (pp. 271–287). Springer.

Nolle, T., Seeliger, A., Thoma, N., & Mühlhäuser, M. (2020). DeepAlign: Alignment-based process anomaly correction using recurrent neural networks. In *Proceedings of the 32nd International Conference on Advanced Information Systems Engineering* (pp. 319–333). Springer.

Oberdorf, F., Schaschek, M., Weinzierl, S., Stein, N., Matzner, M., & Flath, C. M. (2023). Predictive end-to-end enterprise process network monitoring. *Business & Information Systems Engineering*, 49–64.

Okoli, C. (2015). A guide to conducting a standalone systematic literature review. *Communications of the Association for Information Systems*, *37*(1), 879–910.

OpenAI (2023). GPT-4 technical report. (pp. 1–100). arXiv preprint arXiv:2303.08774v3, arXiv:2303.08774v3.

Otter, D. W., Medina, J. R., & Kalita, J. K. (2020). A survey of the usages of deep learning for natural language processing. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(2), 604–624.

Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345–1359. http://dx.doi.org/10.1109/TKDE.2009.191.

Paré, G., Trudel, M.-C., Jaana, M., & Kitsiou, S. (2015). Synthesizing information systems knowledge: A typology of literature reviews. *Information & Management*, *52*(2), 183–199. http://dx.doi.org/10.1016/j.im.2014.08.008.

Park, G., & Song, M. (2019). Prediction-based resource allocation using LSTM and minimum cost and maximum flow algorithm. In *Proceedings of the 1st International Conference on Process Mining* (pp. 121–128). IEEE.

Park, G., & Song, M. (2020). Predicting performances in business processes using deep neural networks. *Decision Support Systems*, *129*, Article 113191.

Pasquadibisceglie, V., Appice, A., Castellano, G., & Malerba, D. (2019). Using convolutional neural networks for predictive process analytics. In *Proceedings of the 1st International Conference on Process Mining* (pp. 129–136). IEEE.

Pasquadibisceglie, V., Appice, A., Castellano, G., & Malerba, D. (2020). Predictive process mining meets computer vision. In *Proceedings of the 18th International Conference of Business Process Management (Forum)* (pp. 176–192). Springer.

Pasquadibisceglie, V., Appice, A., Castellano, G., & Malerba, D. (2021). A multi-view deep learning approach for predictive business process monitoring. *IEEE Transactions on Services Computing*, *15*(4), 2382–2395.

Pasquadibisceglie, V., Appice, A., Castellano, G., Malerba, D., & Modugno, G. (2020). ORANGE: Outcome-oriented predictive process monitoring based on image encoding and CNNs. *IEEE Access*, *8*, 184073–184086.

Pasquadibisceglie, V., Castellano, G., Appice, A., & Malerba, D. (2021). FOX: A neuro-fuzzy model for process outcome prediction and explanation. In *Proceedings of the 3rd International Conference on Process Mining* (pp. 112–119). IEEE.

Pauwels, S., & Calders, T. (2020). Bayesian network based predictions of business processes. In *Proceedings of the 18th International Conference of Business Process Management (Forum)* (pp. 159–175). Springer.

Pauwels, S., & Calders, T. (2021). Incremental predictive process monitoring: The next activity case. In *Proceedings of the 19th International Conference on Business Process Management* (pp. 123–140). Springer.

Pegoraro, M., Uysal, M. S., Georgi, D. B., & van der Aalst, W. M. P. (2021). Text-aware predictive monitoring of business processes. In *Proceedings of 24th International Conference on Business Information Systems* (pp. 120–129). Springer.

Pery, A., Rafiei, M., Simon, M., & van der Aalst, W. (2022). Trustworthy artificial intelligence and process mining: Challenges and opportunities. In *Proceedings of 3rd International Conference on Process Mining (Workshops)* (pp. 395–407). IEEE.

Pfeiffer, P., Lahann, J., & Fettke, P. (2021). Multivariate business process representation learning utilizing gramian angular fields and convolutional neural networks. In *Proceedings of the 19th International Conference on Business Process Management* (pp. 327–344). Springer.

Pflug, J., & Rinderle-Ma, S. (2016). Application of dynamic instance queuing to activity sequences in cooperative business process scenarios. *International Journal of Cooperative Information Systems*, *25*(01), Article 1650002.

Pham, D.-L., Ahn, H., Kim, K.-S., & Kim, K. P. (2021). Process-aware enterprise social network prediction and experiment using LSTM neural network models. *IEEE Access*, *9*, 57922–57940.

Philipp, P., Jacob, R., Robert, S., & Beyerer, J. (2020). Predictive analysis of business processes using neural networks with attention mechanism. In *Proceedings of the 2020 International Conference on Artificial Intelligence in Information and Communication* (pp. 225–230). IEEE.

Polančič, G., Jagečič, S., & Kous, K. (2020). An empirical investigation of the effectiveness of optical recognition of hand-drawn business process elements by applying machine learning. *IEEE Access*, *8*, 206118–206131. http://dx.doi.org/10.1109/ACCESS.2020.3034603.

Polato, M., Sperduti, A., Burattin, A., & de Leoni, M. (2014). Data-aware remaining time prediction of business process instances. In *Proceedings of the 2014 International Joint Conference on Neural Networks* (pp. 816–823). IEEE.

Polato, M., Sperduti, A., Burattin, A., & de Leoni, M. (2018). Time and activity sequence prediction of business process instances. *Computing*, *100*(9), 1005–1031.

Qafari, M. S., & van der Aalst, W. M. P. (2019). Fairness-aware process mining. In *Proceedings of the 2019 OTM Confederated International Conferences "On The Move to Meaningful Internet Systems"* (pp. 182–192). Springer.

Qian, C., Wen, L., Kumar, A., Lin, L., Lin, L., Zong, Z., et al. (2020). An approach for process model extraction by multi-grained text classification. In *Proceedings of the 32nd International Conference on Advanced Information Systems Engineering* (pp. 268–282). Springer.

Qiao, M., Akkiraju, R., & Rembert, A. J. (2011). Towards efficient business process clustering and retrieval: combining language modeling and structure matching. In *Proceedings of the 9th International Conference on Business Process Management* (pp. 199–214). Springer.

Rama-Maneiro, E., Vidal, J. C., & Lama, M. (2021). Deep learning for predictive business process monitoring: Review and benchmark. *IEEE Transactions on Services Computing*, 1–18. http://dx.doi.org/10.1109/TSC.2021.3139807.

Recker, J., & Mendling, J. (2016). The state of the art of business process management research as published in the BPM conference: Recommendations for progressing the field. *Business & Information Systems Engineering*, *58*(1), 55–72. http://dx.doi.org/10.1007/s12599-015-0411-3.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Why should I trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd International Conference on Knowledge Discovery and Data Mining* (pp. 1135–1144). ACM.

Rizzi, W., Di Francescomarino, C., & Maggi, F. M. (2020). Explainability in predictive process monitoring: When understanding helps improving. In *Proceedings of the 18th International Conference on Business Process Management (Forum)* (pp. 141–158). Springer.

Rogge-Solti, A., & Kasneci, G. (2014). Temporal anomaly detection in business processes. In *Proceedings of the 12th International Conference on Business Process Management* (pp. 234–249). Springer.

Romano, J. D., Le, T. T., La Cava, W., Gregg, J. T., Goldberg, D. J., Chakraborty, P., et al. (2021). PMLB v1.0: an open source dataset collection for benchmarking machine learning methods. arXiv preprint arXiv:2012.00058v2.

Rozinat, A., & van der Aalst, W. M. P. (2006). Decision mining in ProM. In *Proceedings of the 4th International Conference on Business Process Management* (pp. 420–425). Springer.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). Learning internal representations by error propagation. USA: California University.

Russel, S., & Norvig, P. (2016). *Artificial intelligence: A modern approach* (3rd ed.). Prentice Hall.

Samiri, M. Y., Najib, M., El Fazziki, A., & Boukachour, J. (2017). Toward a self-adaptive workflow management system through learning and prediction models. *Arabian Journal for Science and Engineering*, *42*(2), 897–912.

Saraeian, S., Shirazi, B., & Motameni, H. (2019). Optimal autonomous architecture for uncertain processes management. *Information Sciences*, *501*, 84–99.

Sarno, R., Ginardi, H., Pamungkas, E. W., & Sunaryono, D. (2013). Clustering of ERP business process fragments. In *Proceedings of the 2013 International Conference on Computer, Control, Informatics and its Applications* (pp. 319–324). IEEE.

Sarno, R., Sari, P. L. I., Ginardi, H., Sunaryono, D., & Mukhlash, I. (2013). Decision mining for multi choice workflow patterns. In *Proceedings of the 2013 International Conference on Computer, Control, Informatics and its Applications* (pp. 337–342). IEEE.

Sarno, R., & Sungkono, K. R. (2016a). Hidden markov model for process mining of parallel business processes. *International Review on Computers and Software*, *11*(4), 290–300.

Sarno, R., & Sungkono, K. R. (2016b). Coupled hidden markov model for process mining of invisible prime tasks. *International Review on Computers and Software*, *11*(6), 539–547.

Savickas, T., & Vasilecas, O. (2018). Belief network discovery from event logs for business process analysis. *Computers in Industry*, *100*, 258–266.

Seeliger, A., Luettgen, S., Nolle, T., & Mühlhäuser, M. (2021). Learning of process representations using recurrent neural networks. In *Proceedings of the 33rd International Conference on Advanced Information Systems Engineering* (pp. 109–124). Springer.

Seeliger, A., Nolle, T., & Mühlhäuser, M. (2018). Finding structure in the unstructured: Hybrid feature set clustering for process discovery. In *Proceedings of the 16th International Conference on Business Process Management* (pp. 288–304). Springer.

Senderovich, A., Di Francescomarino, C., Ghidini, C., Jorbina, K., & Maggi, F. M. (2017). Intra and inter-case features in predictive process monitoring: A tale of two dimensions. In *Proceedings of the 15th International Conference on Business Process Management* (pp. 306–323). Springer.

Senderovich, A., Weidlich, M., Gal, A., & Mandelbaum, A. (2014). Mining resource scheduling protocols. In *Proceedings of the 12th International Conference on Business Process Management* (pp. 200–216). Springer.

Senderovich, A., Weidlich, M., Gal, A., & Mandelbaum, A. (2015). Queue mining for delay prediction in multi-class service processes. *Information Systems*, *53*, 278–295.

Settles, B. (2009). *Active learning literature survey*: *Technical Report*, USA: University of Wisconsin-Madison.

Shraga, R., Gal, A., Schumacher, D., Senderovich, A., & Weidlich, M. (2020). Process discovery with context-aware process trees. *Information Systems*, *106*, Article 101533.

Silva, R., Zhang, J., & Shanahan, J. G. (2005). Probabilistic workflow mining. In *Proceedings of the 11th International Conference on Knowledge Discovery in Data Mining* (pp. 275–284). ACM.

Sim, S., Sutrisnowati, R. A., Won, S., Lee, S., & Bae, H. (2022). Automatic conversion of event data to event logs using CNN and event density embedding. *IEEE Access*, *10*, 15994–16009.

Sindhgatta, R., Moreira, C., Ouyang, C., & Barros, A. (2020). Exploring interpretable predictive models for business processes. In *Proceedings of the 18th International Conference on Business Process Management* (pp. 257–272). Springer.

Sommers, D., Menkovski, V., & Fahland, D. (2021). Process discovery using graph neural networks. In *Proceedings of the 3rd International Conference on Process Mining* (pp. 40–47). IEEE.

Stierle, M., Brunk, J., Weinzierl, S., Zilker, S., Matzner, M., & Becker, J. (2021). Bringing light into the darkness – A systematic literature review on explainable predictive business process monitoring techniques. In *Proceedings of the 29th European Conference on Information Systems* (pp. 1–12). AIS.

Stierle, M., Weinzierl, S., Harl, M., & Matzner, M. (2021). A technique for determining relevance scores of process activities using graph-based neural networks. *Decision Support Systems*, *144*, Article 113511. http://dx.doi.org/10.1016/j.dss.2021.113511.

Tavares, G. M., Ceravolo, P., da Costa, V. G. T., Damiani, E., & Junior, S. B. (2019). Overlapping analytic stages in online process mining. In *Proceedings of the 2019 IEEE International Conference on Services Computing* (pp. 167–175). IEEE.

Tax, N., Sidorova, N., Haakma, R., & van der Aalst, W. M. P. (2016). Event abstraction for process mining using supervised learning techniques. In *Proceedings of SAI Intelligent Systems Conference* (pp. 251–269). Springer.

Tax, N., Verenich, I., La Rosa, M., & Dumas, M. (2017). Predictive business process monitoring with LSTM neural networks. In *Proceedings of the 29th International Conference on Advanced Information Systems Engineering* (pp. 477–492). Springer.

Taymouri, F., La Rosa, M., Dumas, M., & Maggi, F. M. (2021). Business process variant analysis: Survey and classification. *Knowledge-Based Systems*, *211*, Article 106557. http://dx.doi.org/10.1016/j.knosys.2020.106557.

Taymouri, F., La Rosa, M., & Erfani, S. (2021). A deep adversarial model for suffix and remaining time prediction of event sequences. In *Proceedings of the 2021 SIAM International Conference on Data Mining* (pp. 522–530). SIAM.

Taymouri, F., La Rosa, M., Erfani, S., Bozorgi, Z. D., & Verenich, I. (2020). Predictive business process monitoring via generative adversarial nets: The case of next event prediction. In *Proceedings of the 18th International Conference on Business Process Management* (pp. 237–256). Springer.

Teinemaa, I., Dumas, M., La Rosa, M., & Maggi, F. M. (2019). Outcome-oriented predictive process monitoring: Review and benchmark. *ACM Transactions on Knowledge Discovery from Data*, *13*(2), 1–57. http://dx.doi.org/10.1145/3301300.

Teinemaa, I., Dumas, M., Maggi, F. M., & Di Francescomarino, C. (2016). Predictive business process monitoring with structured and unstructured data. In *Proceedings of the 14th International Conference on Business Process Management* (pp. 401–417). Springer.

Teinemaa, I., Tax, N., de Leoni, M., Dumas, M., & Maggi, F. M. (2018). Alarm-based prescriptive process monitoring. In *Proceedings of the 16th International Conference on Business Process Management (Forum)* (pp. 91–107). Springer.

Tello, G., Gianini, G., Mizouni, R., & Damiani, E. (2019). Machine learning-based framework for log-lifting in business process mining applications. In *Proceedings of the 17th International Conference on Business Process Management* (pp. 232–249). Springer.

Theis, J., & Darabi, H. (2019). Decay replay mining to predict next process events. *IEEE Access*, *7*, 119787–119803.

Theis, J., & Darabi, H. (2020). Adversarial system variant approximation to quantify process model generalization. *IEEE Access*, *8*, 194410–194427.

Tiwari, A., Turner, C. J., & Majeed, B. (2008). A review of business process mining: State-of-the-art and future trends. *Business Process Management Journal*, *14*(1), 5–22.

Trkman, P. (2010). The critical success factors of business process management. *International Journal of Information Management*, *30*, 125–134. http://dx.doi.org/10.1016/j.ijinfomgt.2009.07.003.

Turner, C. J., Tiwari, A., & Mehnen, J. (2008). A genetic programming approach to business process mining. In *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation* (pp. 1307–1314).

Valdés, J. J., Céspedes-González, Y., & Pou, A. (2022a). Process mining as a time series analysis tool via conformance checking. In *Proceedings of the 2021 Future Technologies Conference* (pp. 636–649). Springer, http://dx.doi.org/10.1007/978-3-030-89906-6_42.

Valdés, J. J., Céspedes-González, Y., & Pou, A. (2022b). Process mining as a time series analysis tool via graph kernels. In *Proceedings of the 2021 Future Technologies Conference* (pp. 832–847). Springer, http://dx.doi.org/10.1007/978-3-030-89906-6_53.

van der Aa, H., Carmona Vargas, J., Leopold, H., Mendling, J., & Padró, L. (2018). Challenges and opportunities of applying natural language processing in business process management. In *Proceedings of the 27th International Conference on Computational Linguistics* (pp. 2791–2801). ACL.

van der Aalst, W. M. P. (2016). Process mining. In *Data science in action* (2nd ed.). Springer.

van der Aalst, W. (2019). Object-centric process mining: Dealing with divergence and convergence in event data. In *Proceedings of the 17th International Conference on Software Engineering and Formal Methods* (pp. 3–25). Springer.

van der Aalst, W. M. P., Adriansyah, A., De Medeiros, A. K. A., Arcieri, F., Baier, T., Blickle, T., et al. (2011). Process mining manifesto. In *Proceedings of the 9th International Conference of Business Process Management (Workshops)* (pp. 169–194). Springer.

van der Aalst, W. M. P., Bichler, M., & Heinzl, A. (2017). Responsible data science. *Business & Information Systems Engineering*, *59*(5), 311–313.

van der Aalst, W. M. P., De Medeiros, A. K. A., & Weijters, A. J. (2005). Genetic process mining. In *Proceedings of the 26th International Conference on Application and Theory of Petri Nets* (pp. 48–69). Springer.

van Dongen, B. F., Jansen-Vullers, M. H., Verbeek, H., & van der Aalst, W. M. (2007). Verification of the SAP reference models using EPC reduction, state-space analysis, and invariants. *Computers in Industry*, *58*(6), 578–601.

Varela-Vaca, Á. J., Galindo, J. A., Ramos-Gutiérrez, B., Gómez-López, M. T., & Benavides, D. (2019). Process mining to unleash variability management: Discovering configuration workflows using logs. In *Proceedings of the 23rd International Systems and Software Product Line Conference* (pp. 265–276). ACM.

Vázquez-Barreiros, B., Mucientes, M., & Lama, M. (2014). A genetic algorithm for process discovery guided by completeness, precision and simplicity. In *Proceedings of the 12th International Conference on Business Process Management* (pp. 118–133). Springer.

Vázquez-Barreiros, B., Mucientes, M., & Lama, M. (2015). ProDiGen: Mining complete, precise and minimal structure process models with a genetic algorithm. *Information Sciences*, *294*, 315–333.

Verenich, I., Dumas, M., La Rosa, M., Maggi, F. M., & Di Francescomarino, C. (2016). Minimizing overprocessing waste in business processes via predictive activity ordering. In *Proceedings of the 28th International Conference on Advanced Information Systems Engineering* (pp. 186–202). Springer.

Verenich, I., Dumas, M., La Rosa, M., Maggi, F. M., & Teinemaa, I. (2019). Survey and cross-benchmark comparison of remaining time prediction methods in business process monitoring. *ACM Transactions on Intelligent Systems and Technology*, *10*(4), 1–34. http://dx.doi.org/10.1145/3331449.

Vergidis, K., Tiwari, A., Majeed, B., & Roy, R. (2007). Optimisation of business process designs: An algorithmic approach with multiple objectives. *International Journal of Production Economics*, *109*(1–2), 105–121.

Vidgof, M., Bachhofner, S., & Mendling, J. (2023). Large language models for business process management: Opportunities and challenges. (pp. 1–16). arXiv preprint arXiv:2304.04309, arXiv:2304.04309.

vom Brocke, J., Simons, A., Riemer, K., Niehaves, B., Plattfaut, R., & Cleven, A. (2015). Standing on the shoulders of giants: Challenges and recommendations of literature search in information systems research. *Communications of the Association for Information Systems*, *37*, 205–224.

Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. *Computational Intelligence and Neuroscience*, *2018*.

Wahid, N. A., Adi, T. N., Bae, H., & Choi, Y. (2019). Predictive business process monitoring – Remaining time prediction using deep neural network with entity embedding. *Procedia Computer Science*, *161*, 1080–1088.

Wang, J., Wong, R. K., Ding, J., Guo, Q., & Wen, L. (2012). Efficient selection of process mining algorithms. *IEEE Transactions on Services Computing*, *6*(4), 484–496.

Wang, J., Yu, D., Liu, C., & Sun, X. (2019). Outcome-oriented predictive process monitoring with attention-based bidirectional LSTM neural networks. In *Proceedings of the 2019 IEEE International Conference on Web Services* (pp. 360–367). IEEE.

Wanner, J., Wissuchek, C., & Janiesch, C. (2020). Machine learning and complex event processing: A review of real-time data analytics for the industrial internet of things. *Enterprise Modelling and Information Systems Architectures*, *15*, 1.

Watkins, C. J. C. H. (1989). *Learning from delayed rewards*. King's College.

Webster, J., & Watson, R. T. (2002). Analyzing the past to prepare for the future: Writing a literature review. *MIS Quarterly*, xiii–xxiii.

Weinzierl, S., Dunzer, S., Zilker, S., & Matzner, M. (2020). Prescriptive business process monitoring for recommending next best actions. In *Proceedings of the 18th International Conference of Business Process Management (Forum)* (pp. 193–209). Springer.

Weinzierl, S., Wolf, V., Pauli, T., Beverungen, D., & Matzner, M. (2022). Detecting temporal workarounds in business processes – A deep-learning-based method for analysing event log data. *Journal of Business Analytics*, *5*(1), 76–100.

Weinzierl, S., Zilker, S., Stierle, M., Matzner, M., & Park, G. (2020). From predictive to prescriptive process monitoring: Recommending the next best actions instead of calculating the next most likely events. In *Proceedings of the 15th International Conference on Wirtschaftsinformatik* (pp. 364–368).

Wetzstein, B., Leitner, P., Rosenberg, F., Brandic, I., Dustdar, S., & Leymann, F. (2009). Monitoring and analyzing influential factors of business process performance. In *Proceedings of the 2009 IEEE International Enterprise Distributed Object Computing Conference* (pp. 141–150). IEEE.

Weytjens, H., & De Weerdt, J. (2021). Learning uncertainty with artificial neural networks for improved remaining time prediction of business processes. In *Proceedings of the 19th International Conference on Business Process Management* (pp. 141–157). Springer.

Weytjens, H., & Weerdt, J. D. (2021). Creating unbiased public benchmark datasets with data leakage prevention for predictive process monitoring. In *Proceedings of the 19th International Conference on Business Process Management (Workshops)* (pp. 18–29). Springer.

Wibisono, A., Nisafani, A. S., Bae, H., & Park, Y.-J. (2015). On-the-fly performance-aware human resource allocation in the business process management systems environment using naïve Bayes. In *Proceedings of the 3rd Asia-Pacific Conference on Business Process Management* (pp. 70–80). Springer.

Wickramanayake, B., He, Z., Ouyang, C., Moreira, C., Xu, Y., & Sindhgatta, R. (2022). Building interpretable models for business process prediction using shared and specialised attention mechanisms. *Knowledge-Based Systems*, *248*, Article 108773.

Wu, X., Xiao, L., Sun, Y., Zhang, J., Ma, T., & He, L. (2022). A survey of human-in-the-loop for machine learning. *Future Generation Computer Systems*, *135*, 364–381.

Xu, J., & Liu, J. (2019). A profile clustering based event logs repairing approach for process mining. *IEEE Access*, *7*, 17872–17881.

Xu, J., Liu, C., Zhao, X., Yongchareon, S., & Ding, Z. (2016). Resource management for business process scheduling in the presence of availability constraints. *ACM Transactions on Management Information Systems*, *7*(3), 1–26.

Xu, C., Tao, D., & Xu, C. (2013). A survey on multi-view learning. (pp. 1–59). arXiv preprint arXiv:1304.5634, arXiv:1304.5634.

Yang, S., Dong, X., Sun, L., Zhou, Y., Farneth, R. A., Xiong, H., et al. (2017). A data-driven process recommender framework. In *Proceedings of the 23rd International Conference on Knowledge Discovery and Data Mining* (pp. 2111–2120). ACM.

Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology*, *10*(2), 1–19. http://dx.doi.org/10.1145/3298981.

Yang, J., Ouyang, C., Pan, M., Yu, Y., & ter Hofstede, A. H. (2018). Finding the "liberos": Discover organizational models with overlaps. In *Proceedings of the 16th International Conference on Business Process Management* (pp. 339–355). Springer.

Yeshchenko, A., Di Ciccio, C., Mendling, J., & Polyvyanyy, A. (2019). Comprehensive process drift detection with visual analytics. In *Proceedings of the 38th International Conference on Conceptual Modeling* (pp. 119–135). Springer.

Yeshchenko, A., Durier, F., Revoredo, K., Mendling, J., & Santoro, F. (2018). Context-aware predictive process monitoring: The impact of news sentiment. In *Proceedings of the 2018 OTM Confederated International Conferences "On The Move to Meaningful Internet Systems"* (pp. 586–603). Springer.

Zairi, M. (1997). Business process management: A boundaryless approach to modern competitiveness. *Business Process Management Journal*, *3*, 64–80. http://dx.doi.org/10.1108/14637159710161585.

Zanzotto, F. M. (2019). Human-in-the-loop artificial intelligence. *Journal of Artificial Intelligence Research*, *64*, 243–252.

Zhang, Y., & Yang, Q. (2021). A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 1–20. http://dx.doi.org/10.1109/TKDE.2021.3070203.

Zhao, W., Liu, H., Dai, W., & Ma, J. (2016). An entropy-based clustering ensemble method to support resource allocation in business process management. *Knowledge and Information Systems*, *48*(2), 305–330. http://dx.doi.org/10.1007/s10115-015-0879-7.

Zhao, W., Pu, S., & Jiang, D. (2020). A human resource allocation method for business processes using team faultlines. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, *50*(9), 2887–2900.

Zhu, X. (2005). *Semi-supervised learning literature survey*: *Technical Report*, USA: University of Wisconsin-Madison.