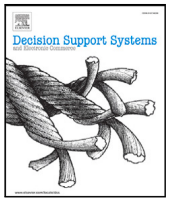
Decision Support Systems 175 (2023) 114043

Contents lists available at ScienceDirect

Decision Support Systems

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



Utilizing the omnipresent: Incorporating digital documents into predictive process monitoring using deep neural networks

Sergej Levich *, Bernhard Lutz, Dirk Neumann

University of Freiburg, Rempartstr. 16, 79098 Freiburg, Germany

ARTICLE INFO

Dataset link: https://github.com/serge724/ppm_docs

Keywords:
Predictive process monitoring
Document processing
Business process management
Deep learning
Natural language processing
Explainable AI

ABSTRACT

Predictive process monitoring (PPM) allows companies to improve the efficiency of their business processes by predicting aspects such as the process outcome, the next event, or the time until the next event. So far, existing studies have mainly focused on developing novel predictive models while using features solely from event logs. In this study, we aim to go beyond log data and increase the focus of PPM research towards external context information. To this end, we consider digital documents as they are omnipresent in many business processes and their inclusion can often be justified by a business rationale. However, incorporating digital documents into PPM models poses considerable challenges as they present unstructured data that can contain visual and textual cues of future process behavior, while manual feature extraction is generally not feasible. Therefore, we propose an approach that processes digital documents based on automated visual and textual feature extraction methods. Furthermore, we design a tailored integration module which transforms the extracted features from multiple document pages into a fixed-size representation that subsequently serves as input for the predictive models. Our evaluation, based on a real-world dataset of insurance claims from a mid-sized German insurance company, featuring 5131 process instances with 32,058 events and 39,242 document pages, shows that incorporating digital documents improves the performance by significant margins in predicting the damage type, the next event, and the time until the next event. Finally, we analyze how digital documents contribute to the model's predictions in terms of Shapley additive explanations.

1. Introduction

Excelling at operational business processes is a key competitive goal for organizations. This is prominently reflected in the size of the theoretical body of literature about business process management [BPM, 1, 2]. Enabled by continuous technological advances, the focus in BPM is shifting from manual, labor-intense activities towards the implementation of predictive analytics [3–5], which can be observed within the field of predictive process monitoring (PPM). Research on PPM is concerned with the prediction of various characteristics of running processes instances, like the process outcome (e.g., [6,7]), the next event (e.g., [8,9]), or the time until completion of a process instance (e.g., [10,11]). Accurate PPM models can benefit process managers in several ways. For instance, accurate predictions at the instance-level of a process like the outcome (e.g., whether an insurance claim is valid) are useful to ensure process compliance. Similarly, accurate predictions at the event-level, such as predictions for the next event or the time until the next event, are valuable for efficient task assignments and early identification of temporal delays in process execution [12–14].

So far, PPM studies have primarily focused on the architecture of the underlying predictive model (e.g., [7–9]). Although these studies achieved remarkable results in predicting different aspects of business process instances, the vast majority of the proposed models rely solely on structured data from event logs. The proposed models are already based on powerful deep learning architectures like the long short-term memory [LSTM, 8,15], or convolutional neural networks [CNN, 16, 17], which are well suited for processing sequential data as given by event logs. Further improvement of model architectures without the consideration of context information is likely to lead to a performance saturation as pure process logs cannot be expected to contain all information that is required to predict future process behavior. Against this background, the growing concept of "data-centric AI" [18] stipulates that researchers should shift from advancing model architectures towards improving predictive performance in terms of the input data. In the context of PPM, this refers to complementing log data in regard to context data that contains relevant cues of future process behavior. However, although BPM and PPM studies have long acknowledged the importance of context data [19–21], external context

* Corresponding author.
E-mail addresses: sergej.levich@is.uni-freiburg.de (S. Levich), bernhard.lutz@is.uni-freiburg.de (B. Lutz), dirk.neumann@is.uni-freiburg.de (D. Neumann).

https://doi.org/10.1016/j.dss.2023.114043
Received 5 August 2022; Received in revised form 22 June 2023; Accepted 26 June 2023
Available online 1 July 2023

information outside of log files received little attention in the PPM literature (see [22,23]).

In this study, we aim to increase the focus of PPM towards external context information given by digital documents. Digital documents like scanned letters, emails, or invoices stored in the form of PDF or image files are omnipresent in many business processes [20,24,25], and their relevance for future process behavior can often be justified by a particular business rationale. For instance, the business rationale in an insurance claims management process may stipulate that the presence of an invoice leads to a payment. The absence of this document can, in contrast, result in the clerk requesting additional information from the client. Features from digital documents may hence complement event log data to serve as input for a PPM model. Fig. 1 presents several examples of digital documents (i.e., an email, an invoice, a police report, and a German vehicle registration). All documents contain different textual and visual cues, characterized by a particular layout, like checkboxes, tabularized content, headers, and footers, which may help to predict various aspects of a running process instance.

However, incorporating digital documents into state-of-the-art PPM models is not straightforward. Digital documents present unstructured data with no predefined features. Manual identification and extraction of relevant features is often not feasible as it might not be clear which features of a document are relevant in a given situation. Moreover, process managers would have to identify relevant features of all occurring document types, which is not possible for a large variety of business processes. Manual feature extraction also requires further human effort and can be subject to mistakes. Given these considerations and the fact that process logs from practice are already subject to data collection errors [26], a fully automated integration of digital documents appears as a promising alternative. Yet, integrating PDF or image files into established deep learning architectures creates its own challenges. Sequential architectures like the LSTM typically operate at the event-level as the log data describes the sequence of events [8,24,27]. However, one event can be linked to multiple document pages, which requires several document pages to be transformed into a single representation of fixed size.

We present a PPM approach that directly incorporates digital documents into an LSTM model. For this purpose, we propose and evaluate several publicly available and pretrained neural network architectures as feature extractors. Specifically, we consider CNNs to extract visual features, bidirectional encoder representations from transformers [BERT, 28] to extract textual features, and a hybrid approach [29] to extract both textual and visual features. The extracted features are processed by a tailored integration module that is able to learn a fixed-size document embedding. The sequence of events and document embeddings subsequently serves as input to the LSTM model, which predicts the process outcome, the next event, and the time until the next event. For our evaluation, we acquire a dataset from a mid-sized German insurance company with 5131 insurance claims, 32,058 events, and 39,242 document pages. We find significant performance improvements across all three prediction tasks. The results suggest up to 54.17% relative performance improvements (or 22.1 percentage points) in predicting the damage type of a running process instance. In addition, we find relative improvements of between 6.81% and 10% in predicting the next event and the time until the next event. Finally, we apply methods from explainable AI (XAI) by calculating Shapley additive explanations [30] to show how digital documents contribute to the model's predictions and validate the business rationale of including digital documents.

Our study contributes to the PPM literature in two ways. To the best of our knowledge, we are the first to extend PPM research towards an inclusion of digital documents into PPM models and quantify the resulting benefit in predictive performance for an insurance claims process. Thereby, we hope to increase the focus of PPM research towards external context information as prior studies mainly focused on improving model architectures based on a limited number of established benchmark datasets consisting of log data only (e.g., [31–33]). In addition, we publish our dataset that consists of real-world log data and the extracted visual and textual features from digital documents. Thereby, we follow best practices in regard to open research [34] and aim to stimulate future PPM research on digital documents.

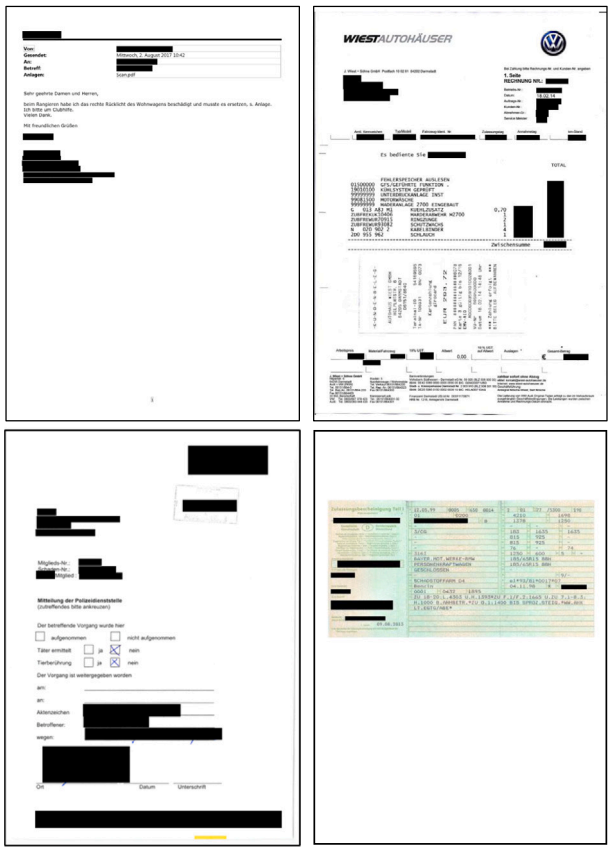


Fig. 1. Examples of digital documents (sensitive information is anonymized).

The remainder of this study is organized as follows. Section 2 provides an overview of the literature on PPM in general as well as related studies, with a particular focus on context information. Section 3 describes our approach in regard to the model architecture and the extraction of features from digital documents. Section 4 presents the dataset and the results of our evaluation. Section 5 discusses our findings and provides an outlook on future research. Section 6 concludes.

2. Related work

2.1. Predictive process monitoring

PPM aims to support process managers by predicting future aspects of a running process instance [35]. Table 1 summarizes prior studies about PPM. These studies focused on predicting the next event (e.g., [8,9,11,24]), process delays (e.g., [36]), timestamp of next event (e.g., [11,23]), remaining time to completion (e.g., [10,37]), or the outcome of a process instance (e.g., [6,7,22,38]). As shown in the second column of Table 1, the majority of prior studies focused on the predictive model.

Early approaches in PPM generated explicit models of the underlying process such as hidden Markov models [25], probabilistic finite automatons [12,39], or evolutionary decision rules [40]. However, these approaches need to make several ex-ante assumptions about the

Table 1
Related studies about predictive process monitoring.

Study	Focus	Model	Target	Dataset(s)	External context	XAI
Lakshmanan et al. [25]	Context	HMM	PO, NE	Simulated	Structured attributes	✓
Leontjeva et al. [42]	Context	RF	PO	AI, BP'11		
Breuker et al. [12]	Model	PFA	NE	BP'12,13, simulated		✓
Teinemaa et al. [22]	Context	RF, LR	PO	DR, LtC	Short texts	
Evermann et al. [8]	Model	LSTM	NE	BP'12,13		✓
Márquez-Chamorro et al. [40]	Model	EA	PI	BP'13, AHS		✓
Di Francescomarino et al. [43]	Model	LSTM	NE	BP'11-13,17, CSL, HD		
Tax et al. [11]	Model	LSTM	NE, TNE	BP'12, HD		
Yeshchenko et al. [23]	Context	GB	TNE	BP'12,13,17, RTF	Twitter sentiment	
Borkowski et al. [27]	Context	LSTM	NE, PI	BP'17, simulated	Structured attributes	
Camargo et al. [44]	Model	LSTM	NE, TNE, RO	BP'12,13,15, HD		
Di Mauro et al. [45]	Model	CNN	NE	BP'12, CSL, HD		
Pasquadibisceglie et al. [46]	Model	CNN	NE	BP'12, HD		
Brunk et al. [24]	Context	LSTM	NE	BP'12,13,17, HD		
Harl et al. [6]	Model	GGNN	PO	BP'17		✓
Mehdiyev et al. [14]	Model	AE	NE	BP'12,13, HD		
Park and Song [36]	Model	CNN+LSTM	PI	BP'12, HD, own		
Taymouri et al. [47]	Model	GAN+LSTM	NE, TNE	BP'12,17, HD		
Brunk et al. [48]	Context	DBN	NE	BP'12,13		✓
Heinrich et al. [9]	Model	GCNN	NE	BP'11-13, CSL, HD		
Kim et al. [7]	Context	RF, GB	PO	BP'11,12,15, RTF		✓
This study	Context	CNN+LSTM	PO, NE, TNE	Own	Digital documents	✓

Models: HMM = hidden Markov model, RF = random forest, PFA = probabilistic finite automation, LR = logistic regression, EA = evolutionary algorithm, GB = gradient boosting, GGNN = gated graph neural network, AE = autoencoder, GAN = generative adversarial network, DBN = dynamic Bayesian network, GCNN = gated convolutional neural network Targets: PO = process outcome, NE = next event, PI = performance indicators, TNE = time until the next event, RO = role Datasets: AI = Australian insurer, BP = BPI Challenge, DR = debt recovery, LtC = lead-to-contract, AHS = Andalusian Health Service, CSL = CoSeLoG project, HD = HelpDesk, RTF = road traffic fines.

generated explicit process model, which is not required by machine learning methods as they learn the underlying process model implicitly [8,9]. Therefore, researchers shifted their attention to machine learning models that consider the prediction task as a classification or regression problem [41].

More recent studies developed deep learning approaches for PPM due to their superior predictive performance over traditional machine learning models [37,38]. Evermann et al. [8], Mehdiyev et al. [49], and Tax et al. [11] were among the first to consider the LSTM model for the purpose of PPM. Recurrent neural networks like the LSTM are a natural choice for this task as they are specifically designed to capture temporal dependencies among a sequence of inputs [15]. Tax et al. [11] used the LSTM model to predict a sequence of future events (instead of only the next event), and the remaining time until process completion. Taymouri et al. [47] combined the LSTM with generative adversarial networks to predict the next event and the time until the next event.

However, the LSTM architecture is not without competition. Mehdiyev et al. [14] proposed an approach based on stacked autoencoders and a particular feature hashing technique to reduce the size of the input vector given by event log entries. Their evaluation of established PPM datasets suggests superior performance over the LSTM approach by Evermann et al. [8]. In a similar vein, the study by Heinrich et al. [9] argues that the respective deep learning architecture should be chosen according to the corresponding data. Therefore, the authors proposed approaches based on a gated convolutional neural network (GCNN) and a key–value-predict attention network. Their experiments using a total of 11 benchmark datasets showed that their proposed architectures largely outperform prior methods. Harl et al. [6] presented a gated graph neural network (GGNN) to predict the outcome of a loan application process. The proposed architecture was found to be particularly useful for explaining approval or denial decisions.

Taken together, we observe that prior studies mainly focused on improving PPM model architectures based on a limited number of established benchmark datasets like "BPI Challenge" or "HelpDesk" containing log data exclusively, while external context information received comparatively little attention. Our study intends to increase the focus of PPM research towards external context information from digital documents as they often guide the subsequent process behavior. In the following, we review the PPM literature that specifically focused on context information.

2.2. Context information

The idea of context information is not new to PPM in general as business processes involve the creation of intermediate data that governs future process execution [19–21]. The inclusion of external context information into PPM models is known to be particularly promising when it can be substantiated by a business rationale [24,48]. However, Table 1 shows that prior studies mainly evaluated their approaches based on established PPM datasets consisting of pure log data like the "BPI Challenges" [50] and "HelpDesk" [33]. The few works that explicitly focus on the integration of context information into PPM models can be divided into two groups as follows.

The first group [7,24,42,48] extracted additional features from event attributes in the process log. Event attributes from log data can be represented as categorical variables, like the entity performing the activity or the client organization that is involved. Leontjeva et al. [42] analyzed different encodings of context attributes. They proposed an encoding based on hidden Markov models (HMM) that specifies the log-likelihood ratio of the sequence belonging to models describing positive or negative process outcomes. Their results suggest that the encoding based on HMM improves the predictive performance, but not to a significant extent. Brunk et al. [24] evaluated the influence of including context information on the predictive performance of LSTM models. The authors show that context from event attributes can improve the quality of predictions depending on the selected features and datasets. The more recent study by Brunk et al. [48] on context information distinguishes between context as cause or effect in next event prediction. The authors developed a dynamic belief network that allows the user to specify the relationship between context attributes and the control flow. They showed that, if this cause–effect relationship is correctly specified, the approach can achieve state-of-the-art results in predicting the next event. Both studies [24,48] support the premise that context information has a greater utility if its impact on the process can be substantiated in terms of a business rationale. Kim et al. [7] argue that the common approach of encoding the human resource as a categorical variable neglects valuable information about prior

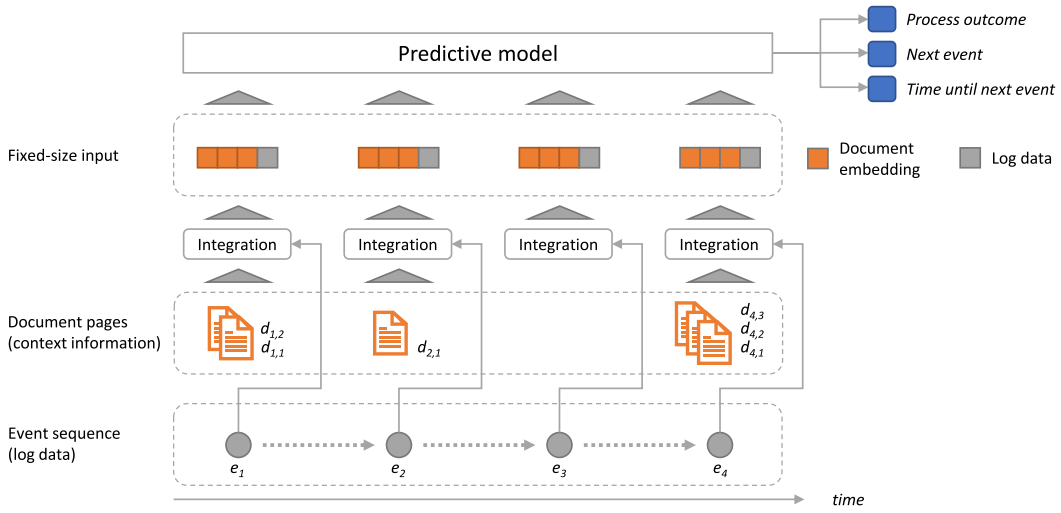


Fig. 2. Conceptual framework. The event log sequence (bottom) consists of four events $e_1, \ldots, e_4$. Events e_1, e_2 and e_4 are linked to one or more document pages. Our approach first learns a fixed-size representation over all document pages, which can subsequently be used together with the log data to predict several aspects of the running process instance (process outcome, next event, and time until the next event).

experiences of the human worker. Therefore, the authors generated additional features based on the human resource that operates on the respective process instance. Their evaluation suggests that the extracted features may improve the accuracy of predictive models depending on the process type.

The second group [22,23,25,27] acquires data from *external* sources, following a specific rationale as to why the additional information qualifies as context. Lakshmanan et al. [25] studied the influence of simulated structured document data in an insurance claims process. Their results suggest that the prediction accuracy of an instance-specific PPM increases if more (simulated) document data becomes available. Teinemaa et al. [22] developed a framework that integrates structured attributes and unstructured attributes (e.g., textual data from emails or comments) for sequence classification models. Yeshchenko et al. [23] acquire news content from online media. Based on this, the authors employed sentiment analysis to enrich the process log data. Their results suggest that including sentiment from external news reduces the error when predicting the remaining time until completion. Borkowski et al. [27] developed and evaluated a framework to incorporate context events as produced by external sources, including internet-of-things devices, sensors, third-party collaborators, and others. Their evaluation suggests that including additional features from context events is particularly useful in internet-of-things applications.

In summary, we find that the PPM literature with a focus on context information has so far neglected digital documents even though they are an omnipresent, rich source of information. In particular, no study so far has attempted to (i) integrate visual and textual cues from digital documents into PPM models, and (ii) estimate the benefits in regard to the predictive performance, which presents the focus of our study.

3. Method

3.1. Overview

A business process instance p with outcome $o_p \in \mathcal{O}$ can be represented as a sequence of events $e_1, \ldots, e_{T_p}$ with length T_p. The whole set of event types is denoted by $\mathcal{E}$. The discrete time component $t = 1, 2, \ldots$ denotes the order in which events occurred in the running process instance. Each event e_t is described by log data l_t and can be associated with an arbitrary number of unlabeled document pages $d_{t,1}, \ldots, d_{t,m}$ with $m \leq M$, where M denotes the maximum possible number of document pages that can be linked to an event.[1] Since we do not require any metadata on the documents, we cannot group multiple pages according to different documents. Instead, we treat all document pages that correspond to an event as a single document with multiple pages. Accordingly, there are two types of sequential inputs, namely, (i) the sequence of events, and (ii) up to M document pages linked to an event.

Fig. 2 shows the conceptual framework, which illustrates a running process instance with a sequence of four events. The events are linked to different numbers of document pages presenting the context information. Following prior PPM research (e.g., [8,24]), we employ a predictive model that processes a sequence of events plus context information as the entire history of log data and context can influence future process behavior. However, this also requires the sequential input of the predictive model to be of fixed size. For this purpose, we design an integration module that learns a document embedding of fixed size over multiple document pages. Given the sequence of events and document embeddings of the running process instance, we aim to predict the process outcome, the next event, and the time until the next event occurs.

The proposed neural network architecture is shown in Fig. 3 and consists of three components as follows.[2] First, a feature extraction method is used to extract features from all document pages linked to the current event. For this purpose, we evaluate a total of three publicly available pretrained neural networks; namely (i) the convolutional neural network VGG-16 [17] to extract visual features, (ii) a BERT model [28] to extract textual features, and (iii) a multimodal approach LayoutXLM [29] to extract textual and visual features. Second, we design an integration module to process the extracted features from all document pages linked to the current event to learn a document embedding of fixed size. The integration module also combines the representation of the document pages with the log data, which then serves as fixed-size input for the predictive model. Finally, we follow prior studies in implementing the predictive model based on two LSTM layers (e.g., [8,24,27]). The predictive model processes the output sequence of the integration module (i.e., document embedding and log

[1] In general, each document page is associated with a single event, thereby qualifying the document as a dynamic attribute of the process instance [51]. Although not considered in this work, there can also be documents that constitute the static context of an entire process instance. Such static documents could be mapped to all individual process events.

[2] The code is publicly available at https://github.com/serge724/ppm_docs.

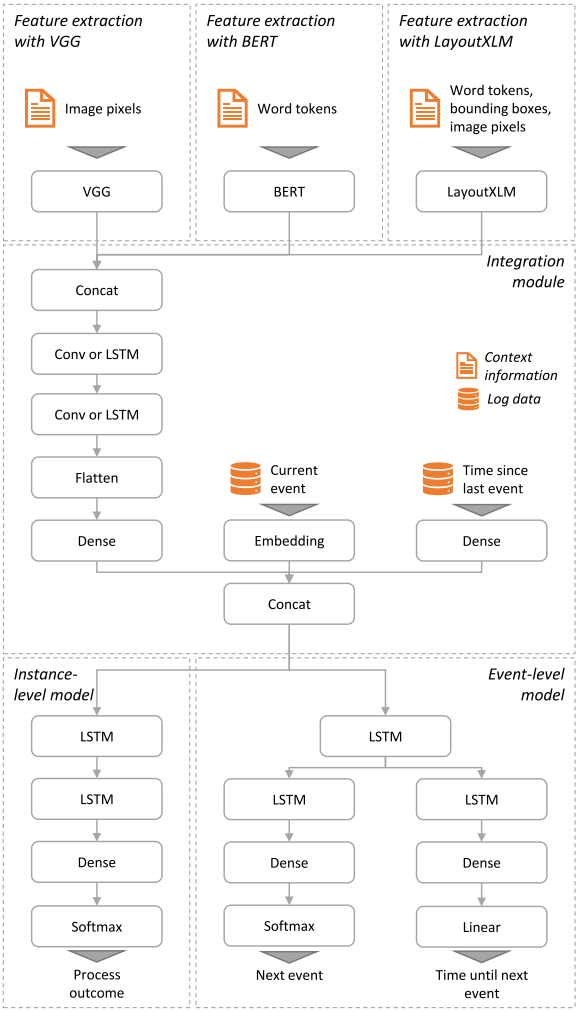


Fig. 3. Model architecture. We train and evaluate a total of three feature extractors (visual features, textual features, and hybrid of visual and textual features). The integration module depends on the type of feature extractor (convolutional or LSTM layers). The architecture also varies according to the level at which predictions are made. The bottom left shows the layers used for predictions at the process instance-level, while the bottom right layers are used for predictions at the event-level.

data) to predict the process outcome, or the next event and the time until next event. In the following, we describe each component of the architecture in detail.

3.2. Feature extraction from digital documents

We first need a meaningful representation of the document pages $d_{t,1}, \ldots, d_{t,m}$ linked to the current event e_t. This representation should yield rich but abstract features of the pages that are invariant to the individual differences between the pages that do not contain useful information [52]. Neural networks appear to be a suitable choice for this task as they have been shown to perform well in extracting features from unstructured data like texts or images [53]. Researchers have proposed several network architectures, like VGG-16 [17] and BERT [28], which have been pretrained on specific tasks and can now be used for the purpose of transfer learning. We rely on pretrained feature extractors for two reasons. First, transfer learning is the current paradigm in document processing and produces state-of-the-art results [29]. Second, using pretrained feature extractors is favorable from a practical perspective since smaller organizations may not have the data volume or the computational resources needed to train models with several hundred million parameters.

For a generic approach like ours, we cannot determine the optimal feature extraction method ex-ante. Specifically, we do not know in advance whether textual features or the layout of the document page are more relevant for predicting future aspects of a running process. For instance, in the context of car damage insurance claims, a picture of a car's damage does not contain any textual features at all, but it contains visual cues that explain the particular damage type. Conversely, textual data may be indicative of missing documents that have to be requested by the clerk. We therefore identify and evaluate a total of three feature extraction methods which aim to extract visual and textual features. An overview of all considered feature extraction methods is presented in Table 2. Note that the parameters of the feature extractors are not updated during training.

We assume that digital documents are stored in the form of PDF files (otherwise, documents can easily be converted into a PDF). PDF files can then be of two different types as follows. (i) The PDF contains text, images, and bounding boxes, which applies to documents created from Word documents or emails that are exported as a PDF file. In this case, we can directly extract all information. (ii) The PDF contains only an image, which frequently occurs when a printed document (e.g., a letter from a customer) is scanned. In the latter case, we additionally use OCR to extract text and bounding boxes. We apply this procedure to each document page separately, so we can ultimately retrieve text, image and bounding boxes from each document page.[3] We then concatenate the extracted features for each document $d_{t,1}, \ldots, d_{t,m}$ linked to the current event e_t using feature extractor $\mathcal{F}$

$$f_t = \text{concat}\left(\mathcal{F}(d_{t,1}). \ldots, \mathcal{F}(d_{t,m})\right). \quad (1)$$

Let $D_{\mathcal{F}}$ denote the dimension of the features extracted by $\mathcal{F}$. We pad f_t with zeros according to the maximum number of document pages M linked to an event. Therefore, f_t is of dimension $M \cdot D_{\mathcal{F}}$, where the first $m \cdot D_{\mathcal{F}}$ are non-zero.

Visual features

We use the VGG-16 [17] architecture to extract visual features due to its strong performance in image classification. The VGG-16 has several fully-connected layers prior to the final output layer that performs the image classification. We remove these layers so that VGG-16 only acts as a feature extractor [54]. We evaluate two pretrained models; one is pretrained on the ImageNet dataset [55] while the other is pretrained on the RVL-CDIP dataset (Ryerson Vision Lab Complex Document Information Processing, [56]). The domain of the data used for pretraining has a decisive influence on the learned representations. By considering a model that is pretrained on general image data and one that is pretrained exclusively on document images, we can empirically evaluate this effect. In order to match the input dimensions of the VGG-16 architecture, we resize the image of each document page to 224 × 224 pixels. The dimension of the extracted features of one document page is $D_{\mathcal{F}} = 7 \times 7 \times 512$.

Textual features

To extract textual features, we make use of the BERT model [28], which presents the current state-of-the-art model in natural language processing. A major advantage of transformer architectures over recurrent neural networks is that they perform better in learning long-term dependencies between words in a document [57]. The BERT model operates on the level of tokens (i.e., word pieces like "dam" and "age"). It is trained based on two objectives: next sentence prediction (NSP), and missing token prediction. For the latter, BERT randomly masks 15% of the input tokens and uses the context tokens for predicting the masked token. The NSP task focuses on learning relationships between sentences. Given two input sequences, it aims to predict whether one

[3] We use the Python libraries doc2data version 0.2.0 for handling PDF files and docTR version 0.6.0 for OCR.

Table 2
Overview of feature extraction methods.

Method	Feature type	Feature dimension	Training data	Parameters
VGG-RVL	Visual	7 × 7 × 512	RVL-CDIP	14,714,688
VGG-Imagenet	Visual	7 × 7 × 512	ImageNet	14,714,688
BERT-German	Text	768	German Wikipedia, news data and Open Legal Data	109,081,344
LayoutXLM	Text & Visual	768	Proprietary PDF data, Subset of IIT-CDIP	368,651,456

sentence follows another [28]. In contrast to context-free language models like word2vec [58], the word embeddings generated by BERT are context-dependent. Since words can be polysemic, contextualized models provide more accurate language representations. For instance, consider the sentences "They won the game by accident" and "It was a severe accident". Contextualized language models generate a different embedding for the word "accident", while context-free models create the same word embedding for "accident" in every sentence. While BERT generates embeddings at token-level, it also allows us to extract embeddings of dimension $D_F = 768$ for an entire token sequence. We use a version of the BERT model that was pretrained entirely on texts in the German language, thereby matching the language of our dataset.[4]

Hybrid of visual and textual features

Finally, we also consider a hybrid representation based on both visual and textual features. To this end, we implement the LayoutXLM model as feature extractor [59]. The LayoutXLM is the multilingual successor model to LayoutLM and LayoutLMv2, which were designed specifically for document processing tasks. The main assumption behind the aforementioned models is that, in contrast to pure textual data, documents are a visually rich source of information and cannot be fully understood by relying solely on the text. The simplest example is a form that contains key–value pairs, such as the text "case number:" before the actual number. To correctly infer the relationship between such units of information, the layout of the document needs to be considered. LayoutXLM is fundamentally a BERT-based architecture, which enhances the core transformer blocks with two additional input channels to capture layout information. First, it adds the bounding boxes of each token as 2D position embeddings to the input sequence. Second, it uses a pretrained CNN to create and integrate visual embeddings of the document. Furthermore, the attention mechanism in the transformer blocks and the pretraining tasks are adapted to allow for the effective learning of a rich representation from visual and textual content. Analogously to the BERT model, LayoutXLM yields an embedding for the entire input sequence at the first output position. The embedding dimension is the same dimension as for the BERT model that we use to extract textual features only ($D_F = 768$).[5]

3.3. Integration module

The integration module combines the extracted features from the document pages f_t and the log data l_t of the current event. The integration module is applied at each event in the process sequence. During model training, we replicate the integration module along the temporal event sequence, so that the document pages linked to each event are processed with the same parameters. While this keeps the number of model parameters low, the integration module can still learn a useful representation of the corresponding document pages. The integration module hence contains several trainable layers that learn one document embedding v_t for all document pages $d_{t,1}, \ldots, d_{t,m}$ depending on the type of the extracted features f_t.

For the visual features, the document embedding is generated based on two convolution operations. We first employ a 1 × 1 convolution C_{1x1} to reduce the dimension of the input [60]. Specifically, the 1 × 1 convolution reduces the size of the input from 7 m × 7 × 512 to 7 m × 7 × D. Note that D is a hyperparameter that can be tuned according to the given dataset.[6] Subsequently, we apply a 7 × 7 convolution C_{7x7} with stride 7 × 1, which means that the convolution is applied to each document page vector exactly once. This further reduces the dimension to m × 1 × D. Applying the 7 × 7 convolution operation only once per document page ensures that the kernels do not convolve input data from two different document pages. In other words, we do not convolve features from the bottom of one page with the top of the next page.

For the textual and hybrid feature approach, we employ two LSTM layers with D neurons. The choice of two LSTM layers is motivated by prior research that successfully used LSTM layers to process text [61, 62]. Since the LSTM layers output sequences of length M, we use the last state of the second LSTM layer

$$c_t = \begin{cases} \text{convolve}(\text{convolve}\left(f_t, C_{1\times1}, 1\times1\right), C_{7\times7}, 7\times1) \\ \qquad \text{if visual feature extraction,} \\ \text{LSTM}(\text{LSTM}(f_t)) \\ \qquad \text{otherwise.} \end{cases} \tag{2}$$

The tensor c_t is finally flattened and processed by a dense layer to obtain the embedding representing all document pages $d_{t,1}, \ldots, d_{t,m}$

$$embedding(d_{t,1}, \ldots, d_{t,m}) = W\ \text{flatten}(c_t) + b, \tag{3}$$

with weight matrix W of dimension $D \times M \cdot D$ and bias b of dimension D. An overview of the intermediate results from the integration module and their dimensions is provided in Appendix B of the supplementary material.

The right branch of the integration module incorporates the log data of event e_t. We include a standard embedding layer of dimension D for the type of the current event.[7] In addition, we include the elapsed time since the last event y^T_{t-1} (for the first event, we set $y^T_0 = 0$). Altogether, the output of the integration module (i.e., the input for the predictive model) is given as

$$x_t = \text{concat}(embedding(d_{t,1}, \ldots, d_{t,m}), \text{embedding}(e_t), y^T_{t-1}). \tag{4}$$

Other event attributes (e.g., resource, organization) that are stored in the process log could also be added to (4).

3.4. Predictive model

The predictive model generates the actual predictions of the process outcome, the next event, and the time until the next event. Consistent with prior studies [8,11,24,27], the predictive model is based on two LSTM layers to reflect the sequential nature of log data. We implement two model variants as shown in Fig. 3. Specifically, we implement one model for instance-level predictions (e.g., process outcome), and another model for simultaneous event-level predictions (e.g., next event and the time until the next event). The joint prediction of next event and time until the next event is motivated by Tax et al. [11], who found performance improvements when using two outcome branches over a single outcome branch. During the training process, the instance-level

[4] The model is available via the Python library transformers version 4.25.1 under the name "bert-base-german-cased".

[5] The model is available via the Python library transformers version 4.25.1 under the name "layoutxlm-base".

[6] Further information on our parameters is provided in Appendix A.

[7] Note that D is a hyperparameter that can be tuned based on the number of distinct events in the dataset.

model is exposed to the same target value at each step in the LSTM as the process outcome is an attribute of the entire instance and does not change over time. The dimension of all LSTM layers is set to the embedding dimension parameter D to keep the model complexity low.

Process outcome

The first process model predicts the outcome of a process instance. Accordingly, the output layer is given by a softmax layer with $|\mathcal{O}|$ outputs. A softmax layer presents a probability distribution over all possible outcomes, so that $\hat{o}_i$ denotes the predicted probability for outcome i. The target output o is given as the one-hot encoded vector of the outcome type. This means that o consists of exactly one non-zero entry which encodes the particular damage type. Based on this, we use the cross entropy loss as

$$L(o, \hat{o}) = -\sum_{i=1}^{|\mathcal{O}|} o_i \, log(\hat{o}_i). \tag{5}$$

Next event and time until the next event

The second process model predicts the next event and the time until the next event. The loss function is defined as the sum of the cross entropy loss for the next event and the squared error in the predicted time until the next event. Let y_t^E denote the one-hot encoded target vector for the next event and y_t^T the time until the next event. Further, let $\hat{y}_{t,i}^E$ denote the predicted probability that the next event is of type i and $\hat{y}_t^T$ the predicted time until the next event. We define a joint loss function as

$$L(y_t^E, y_t^T, \hat{y}_t^E, \hat{y}_t^T) = -\sum_{i=1}^{|\mathcal{E}|} y_{t,i}^E \, log(\hat{y}_{t,i}^E) + (y_t^T - \hat{y}_t^T)^2. \tag{6}$$

4. Evaluation

4.1. Dataset

We obtained a dataset of log files and documents from a claims management process of a mid-sized German company from the insurance sector. The company offers a complementary service to its clients for minor car damages (i.e., rodent bites, glass damage, parking damage, animal damage, hit and run, scorch damage) where the costs are below the given threshold of the main car insurance. These damages are not reimbursed by the main insurer but can be partly covered by the company. Here, predicting process characteristics is useful in different ways. Knowledge of the next event or the time until the next event can be used to improve workflows by routing processes to specialized resources. This also applies to the damage type. Especially in larger organizations, different variations of a process are handled by corresponding teams. Moreover, early identification of the damage type in the event sequence can help to identify fraud if one damage type is claimed but in reality, the documents indicate another. Finally, the prediction of the time until the next event can be used to anticipate delays which, in turn, can be used for proactive client communication.

The claims management process is illustrated in Fig. 4. The figure shows the intended process behavior as stipulated by the company.[8] When a claim is filed, the "Notification of claim" event begins a new process instance. Clients may also submit multiple refund requests or change the requested claim type, which can lead to sequences of "notification of claim" events. The claim is then handled by the clerk during the "Processing" event. Here, documents are checked for completeness and then filed during the "Damage documents" event. At this point, incomplete applications may require a "Document request", leading to another iteration in this part of the process. Once the complete application is reviewed, a decision on the reimbursement is made, resulting in the "Notification of payment" or "Rejection" events. Finally, the application receives an audit. While this shows the main flow of events, further clarification with the client, e.g., on the damage type, takes place in events such as "Further inquiry" or "Correspondence". Note that the event type "Other" is not shown in the figure as it denotes that the event does not match any of the other types.

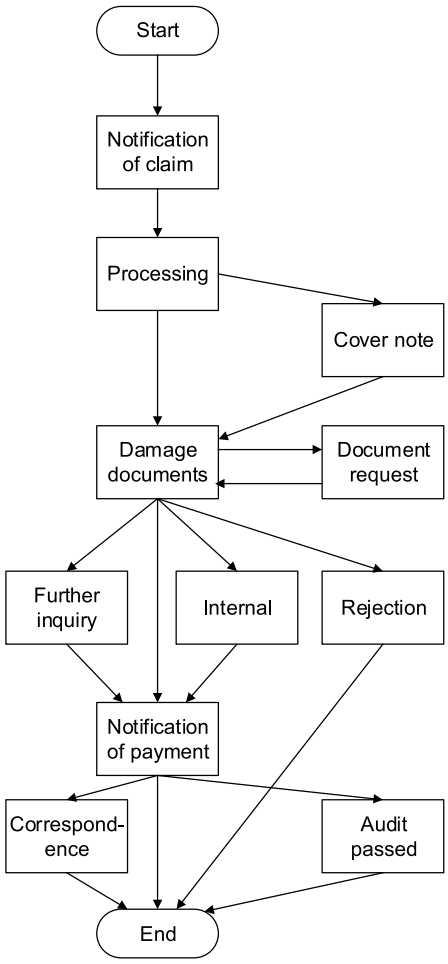


Fig. 4. Intended process behavior as stipulated by the insurance company.

The business rationale for why digital documents guide future process behavior can also be observed in Fig. 4. For instance, predicting whether the event "Damage documents" is followed by a "Document request" requires knowledge of which documents are already present. Even more importantly, a claim is accepted or rejected solely based on the information provided in the damage documents. Here, each damage type has its own requirements for eligibility of a claim, so not only must the relevant documents be present, but their contents also have to support the claim. A PPM model is hence likely to fall short if relying solely on log data. Accordingly, this insurance claims process seems well suited to analyze and quantify the benefit of incorporating digital documents.

The document types that occur in the claims management process are the cover letter, vehicle registration, insurance policy, photos of damages and official certificates by the police. The clerks can send letters to request missing documents or to notify about the acceptance or rejection of a claim. Note that the documents in our dataset are not labeled according to their particular type as the documents are summarized into a single document of multiple pages.

[8] Note that real-world process logs are often noisy [26]. Therefore, we provide the relative frequencies of the transitions between two events in Appendix C of the supplementary material.

Table 3
Descriptive statistics.

	Mean	SD	Min	Max
Events per process instance	6.25	2.46	2	16
Documents per process instance	3.14	1.76	0	14
Document pages per event (> 0 only)	2.43	2.17	1	10
Time until the next event (100 s)	1944	10,782	0.04	618,800
Log time until the next event (s)	5.77	4.32	1.39	17.94

Note: Number of process instances = 5131, number of events = 32,058, number of document pages = 39,242.

The complete dataset consists of 32,058 events from 5131 process instances with 16,132 events (50.3%) linked to a document, and a total of 39,242 document pages.[9] The business process consists of twelve event types. We add an "end of sequence" (EOS) event after the last event of each process instance which yields a total of thirteen event types ($|\mathcal{E}| = 13$). Being able to predict the EOS event is useful in general. For instance, it allows managers to predict when the process ends and to estimate the remaining time of a running process. Table 3 provides further descriptive statistics. The process instances have between 2 and 16 events, while a process instance involves up to 14 documents. Given that an event is linked to a digital document, the number of pages ranges between 1 and 10. The top-10 most frequent sequence patterns of the log data are provided in Appendix C of the supplementary material.

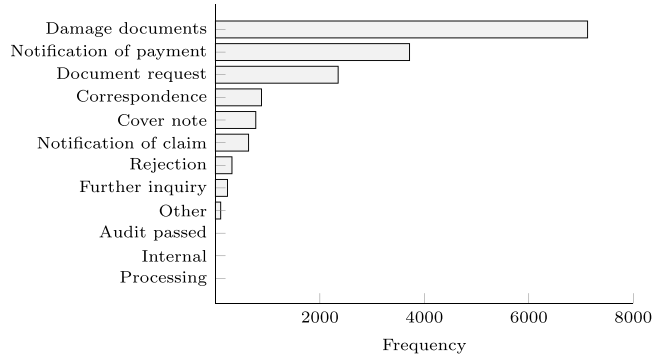


Fig. 5. Number of documents linked to event types.

Fig. 5 shows how often documents are associated with a particular event type. We observe that the vast majority (81.78%) of digital documents are linked to the events "damage documents", "notification of payment", and "document request". Conversely, the events "audit passed", "internal", and "processing" are never linked to a document.

4.2. Procedure

We aim to study the specific benefits of incorporating digital documents into PPM models. For this purpose, we evaluate three baselines which use the same process model as shown in Fig. 3 but without context information from digital documents. The main baseline is a context-free model that does not include any information from digital documents. The other two baselines add very limited count data from digital documents. The second baseline uses a binary flag that signals the presence of at least one document page linked to an event. The third baseline includes the number of document pages m linked to an event. We measure the performance in terms of accuracy for the prediction of the damage type and next event, and mean squared error (MSE) of the predicted time until the next event.

We apply ten-fold cross-validation (e.g., [8,9,14,47,48]) to evaluate the out-of-sample performance of all considered approaches. In each iteration, we select eight folds to train the neural network using RMSProp optimizer [64] while monitoring the loss using a validation fold. We train each model for a maximum of 100 epochs and apply early stopping if no improvement was observed during the last five epochs. Moreover, we apply a grid search on a discrete parameter space to tune parameters for each model architecture.[10] Again, we use the validation loss to select the best model from the search. The out-of-sample performance is finally measured on the remaining test fold. This procedure is repeated ten times so that each fold has constituted the test set exactly once. Additionally, we perform a rolling window evaluation [7] to assess the robustness of the approach towards data drift.

4.3. Performance metrics

We consider several common performance metrics to assess the predictive performance of the implemented models. First, we calculate accuracy to measure the performance in predicting damage and event types. Let N denote the number of samples in the test set, C the number of types, N_i the number of samples of type i, and TP_i, TN_i, FP_i, FN_i the number of true positive, true negative, false positive and false negative predictions of the one-vs-all confusion matrix of type i. Based on this, accuracy is defined as

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{C}\mathrm{TP}_i. \tag{7}$$

We also consider precision and recall for each individual type i

$$\text{Precision}_i = \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FP}_i}, \tag{8}$$

$$\text{Recall}_i = \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FN}_i}. \tag{9}$$

The prediction of the time until the next event is a regression task. Let $y^T_{i,j}$ denote the logarithm of the time until the next event in seconds for the jth observation of event type i. To measure the predictive performance, we calculate the overall mean squared error over all event types as

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{C}\sum_{j=1}^{N_i}(y^T_{i,j} - \hat{y}^T_{i,j})^2. \tag{10}$$

Since each time until the next event prediction is also linked to the next event, we also consider the MSE conditionally on the individual event types. Given that the next event is of type i, the corresponding MSE is defined as

$$\text{MSE}_i = \frac{1}{N_i}\sum_{j=1}^{N_i}(y^T_{i,j} - \hat{y}^T_{i,j})^2. \tag{11}$$

4.4. Results

The results for the overall predictive performance are presented in Table 4. We refer to the models by the name of the used feature extractor and report the performance metrics as mean ± standard deviation. We find that the models using context information from digital documents processed by any feature extractor outperform the baselines on all predictions tasks. The relative performance improvements of BERT-German over the context-free model is 54.17% (22.1 percentage points) in predicting the damage type of the current process

[9] The dataset is available at [63]. Due to privacy constraints, the documents are provided as the extracted features obtained from VGG, the BERT model and LayoutXLM.

[10] The search grids, the resulting parameter choices for each fold and further details on our implementation are presented in Appendix A of the supplementary material.

Table 4
Overall predictive performance.

Architecture	Damage type (accuracy)	Next event (accuracy)	Time until next event (MSE)
Baselines			
Context-free	0.408 ± 0.013	0.558 ± 0.011	13.0 ± 0.266
Binary flag	0.413 ± 0.015	0.563 ± 0.013	12.8 ± 0.290
Page numbers	0.427 ± 0.015	0.576 ± 0.009	12.9 ± 0.306
Digital documents			
VGG-ImageNet	0.538 ± 0.023	0.592 ± 0.013	12.1 ± 0.298
VGG-RVL	0.550 ± 0.017	0.589 ± 0.01	12.0 ± 0.241
LayoutXLM	0.506 ± 0.019	0.577 ± 0.007	12.7 ± 0.239
BERT-German	**0.629 ± 0.016**	**0.596 ± 0.008**	**11.7 ± 0.239**

Note: Number of process instances = 5131, number of events = 32,058, number of document pages = 39,242.
The results are given as mean ± standard deviation over ten-fold cross-validation. MSE refers to the natural logarithm of the seconds until the next event.

instance, 6.81% (4.1 percentage points) in predicting the next event and 10% (1.3 MSE reduction) in predicting the time until the next event. In addition, we find that extracting visual or textual features from digital documents yields significantly greater performance than including features in terms of flags or page numbers. All Wilcoxon rank-sum tests [65] on accuracy and MSE are significant with $p < 0.001$.

Interestingly, we find that the visual feature extractors based on VGG achieve slightly lower accuracy than BERT-German for the prediction of the next event but considerably lower accuracy for the damage type. This indicates that visual and textual features are similarly well suited for event-level predictions, but for instance-level predictions, textual features lead to superior performance. The model using the LayoutXLM features performs worse than other feature extractors, which is unexpected given that the LayoutXLM is a hybrid of both textual and visual features. We see two possible explanations for this unexpected finding. First, LayoutXLM is primarily a text-based model that is extended to incorporate visual features. Therefore, the visual part of its representation might not be as rich as what is provided by purely image-based models. This would explain the performance difference between LayoutXLM and VGG models. Second, the BERT-German model was specifically pretrained on German texts, which matches the language of the documents in our dataset.

We also perform a rolling window analysis to assess the presence of a temporal drift in our dataset. For this purpose, we split the data along the temporal axis and perform a total of four train-test evaluations. For the first evaluation, we use the first 50% of the data for training, the next 10% for validation, and the subsequent 10% for out-of-sample evaluation. For the second evaluation, we add the validation data to the training data and "roll the window" forward by 10% along the temporal axis. This procedure is repeated until the last 10% of the data was used for out-of-sample evaluation. We finally report the means and standard deviations of all four evaluations. The results suggest a lower predictive performance than that of the ten-fold cross-validation (Table 4), although the relative performance ranking of all approaches persists. The decrease in predictive performance suggests the presence of a temporal trend in the dataset. Additionally, lower predictive performance is expected due to the models having considerably less training data compared to ten-fold cross-validation. Detailed results of the rolling window analysis are provided in Appendix E of the supplementary material.

Finally, we perform an ablation study to analyze the importance of prior events from the log data. Intuitively, we expect that the damage type prediction depends less on the event sequence than the predictions of the next event and time until next event. Therefore, we retrain the BERT-German model without log data. The results suggest a reduction in accuracy of 1.4 percentage points to 61.5% in predicting the damage type and a 2.4 percentage points reduction to 57.2% accuracy in predicting the next event. Regarding the logarithmic time until the next event, the MSE increases to 13.2, which yields the worst model overall. Taken together, these results suggest that features extracted from digital documents are mainly useful for predicting the damage type and next event, while log data is particularly relevant for predicting the time between events. This is an intuitive result as our business rationale suggests that documents are more important for the process outcome and the workflow, rather than the timing of the process.

So far, we have considered the overall predictive performance of multi-class prediction problems. In the following, we assess the predictive performance in greater depth depending on the particular damage or event type to provide more useful insights for real-world application.

Damage type

Next, we assess the predictive performances of the context-free baseline and the model using BERT-German for each individual damage type in regard to precision and recall. The results are shown in Table 5. If a model did not predict a particular damage type in each iteration, we cannot properly calculate the performance metrics. This is denoted by "–". We find that including context features consistently improves precision and recall over the model without context from digital documents. The difference is significant, with at least $p < 0.05$, except for the improvement in recall for "rodent bites". The damage types "hit and run" and "scorch damage" occur rather infrequently, so the models do not make a prediction for this type in all iterations.

The damage type is an outcome at the level of the entire process instance. From a practical view, it seems useful to assess the accuracy of predictions depending on the current event number. Intuitively, as more context information becomes available during process execution, accuracy in predicting an instance-level aspect should increase when the model has seen a greater number of events. Accordingly, we calculate the accuracy values of the model using BERT-German features (Fig. 6) and the context-free baseline (Fig. 7) over different event numbers and instance lengths. Darker colors indicate greater accuracy. For instance, in Fig. 6, the value 0.67 in the third row from the bottom indicates that the correct damage is predicted with 67% accuracy after seeing the third event of a process instance with length 4. There are

Table 5
Predictive performance depending on damage type.

Damage type	Prevalence	Precision			Recall		
		Context-free	Context BERT-German	Imp. (%)	Context-free	Context BERT-German	Imp. (%)
Rodent bites	0.400	0.415 ± 0.02	**0.583 ± 0.023**	40.6***	0.745 ± 0.19	**0.813 ± 0.048**	9.14
Glass damage	0.371	0.411 ± 0.10	**0.646 ± 0.049**	57.4***	0.247 ± 0.19	**0.511 ± 0.055**	106.0**
Parking damage	0.146	0.359 ± 0.08	**0.775 ± 0.048**	116.0***	0.142 ± 0.041	**0.538 ± 0.033**	280.0***
Animal damage	0.074	–	**0.808 ± 0.113**	–	–	**0.473 ± 0.080**	–
Hit and run	0.005	–	–	–	–	–	–
Scorch damage	0.004	–	–	–	–	–	–

Note: Number of process instances = 5131, number of events = 32,058, number of document pages = 39,242.
The results are given as mean ± standard deviation over ten-fold cross-validation. "–" denotes that the particular damage type was not predicted in at least one fold. Significance: $^{*}p < 0.05$,$^{**}p < 0.01$,$^{***}p < 0.001$.

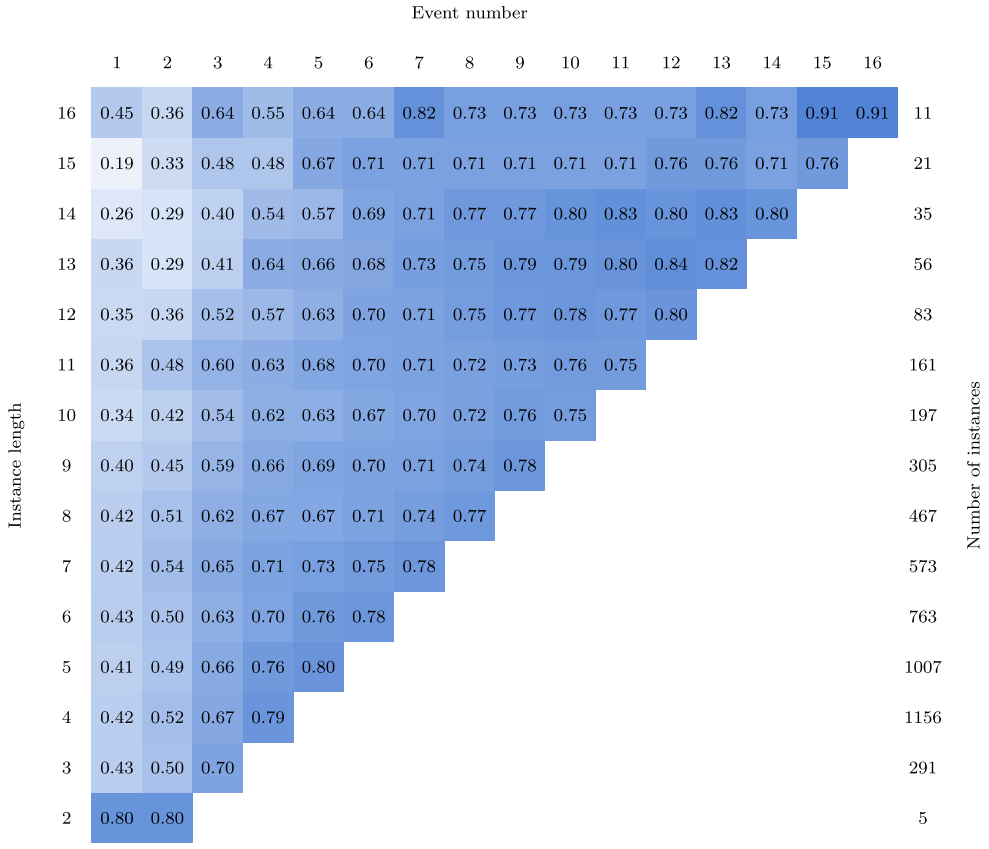


Fig. 6. Accuracy in predicting damage type after seeing different numbers of events given the total instance length for the context-based model BERT-German.

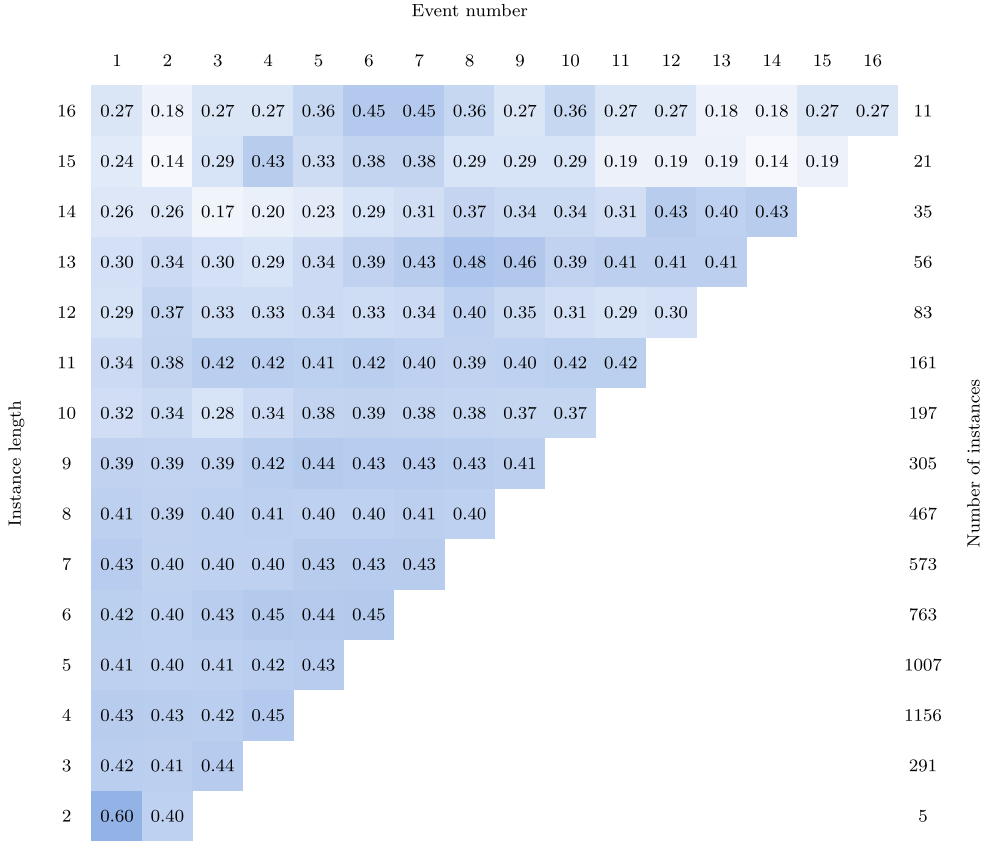


Fig. 7. Accuracy in predicting damage type after seeing different numbers of events given the total instance length for the context-free model.

1156 instances in the dataset with instance length four, as indicated on the right-hand side.

Fig. 6 shows that accuracy generally increases when the model has seen a higher number of events. However, this increase is not consistently monotonic for the instances with more than nine events. There appear to be two possible reasons for this unexpected observation. First, longer instances appear less frequently than shorter instances. Accordingly, the inherent variance in the model's predictions has a larger detrimental impact on the resulting accuracy. Second, longer instances indicate a more complex and unintended process execution. If a claim cannot be decided based on the available documents, the clerk must make further document requests. An example of a long instance is provided in Appendix E of the supplementary material.

Fig. 7 shows that the context-free model has uniformly lower accuracy in predicting the damage type over different event numbers. In contrast to the results shown in Fig. 6, we observe no monotonic increases in accuracy after seeing a greater number of events within a process instance. This observation shows that relevant context information accumulates over time, which is consistent with the business rationale that digital documents govern future process behavior. Taken together, the results suggest a clear advantage of the model incorporating digital documents, which can directly inform practitioners' decisions. Specifically, these results assist PPM practitioners in their decisions about if and when they should rely on the PPM model, provided they have a reasonable estimate about the total length of a process instance.

Next event

Next, we analyze the performance in predicting the next event in terms of recall and precision for each individual event type, as shown in Table 6. For instance, the first row states that if the models predict that the next event is "damage documents", their predictions are correct with 59.1% probability (precision). In addition, the first rows suggests that, if the next event is of type "damage documents", the models predict this with 60.9% and 64.4% probability respectively (recall). The difference in recall is significant with $p < 0.05$. Overall, we find that including context features increases the predictive performance in terms of recall and precision for most event types.

We also consider the improvements in predicting the time until the next event for all individual event types. The results are shown in Table 7. The MSE refers to the natural logarithm of the seconds until the next event. We observe improvements in the mean squared error when including context features for most event types except for the less frequent types "further inquiry", "notification of claim", and "other". However, the improvements are often not significant and smaller in comparison to the improvements in the accuracy of predicting the damage type. Significant improvements of more than ten 10% lower MSE are achieved for "end of sequence" and "audit passed". We provide the boxplots of the mean squared errors for all event types in Appendix D of the supplementary material.

Table 7
Predictive performance for time until the next event (mean squared error).

Next event type	Prevalence	Context-free	Context BERT-German	Imp. (%)
Damage documents	0.222	12.9 ± 0.61	**12.80 ± 0.52**	0.772
End of sequence	0.160	22.2 ± 2.51	**18.6 ± 1.39**	16.4**
Processing	0.160	**1.20 ± 0.01**	1.28 ± 0.191	−7.12
Notification of payment	0.120	7.31 ± 0.51	**6.86 ± 0.43**	6.20
Audit passed	0.103	18.1 ± 1.95	**12.7 ± 1.20**	30.0***
Document request	0.074	9.68 ± 1.07	**9.07 ± 1.31**	6.28
Further inquiry	0.044	**17.8 ± 2.67**	19.3 ± 2.27	−7.90
Correspondence	0.033	29.7 ± 5.44	**29.4 ± 3.84**	0.996
Cover note	0.024	**10.9 ± 2.03**	11.0 ± 2.03	−1.37
Internal	0.023	15.6 ± 2.54	**14.1 ± 2.91**	9.68
Notification of claim	0.019	**9.12 ± 1.34**	9.22 ± 1.33	−1.12
Rejection	0.010	13.5 ± 3.82	**12.6 ± 3.37**	6.17
Other	0.008	26.0 ± 5.54	**24.4 ± 4.73**	6.10

Note: Number of process instances = 5131, number of events = 32,058, number of document pages = 39,242.
MSE refers to the natural logarithm of the seconds until the next event. The results are given as mean ± standard deviation over ten-fold cross-validation.
Significance: $^{*}p < 0.05$,$^{**}p < 0.01$,$^{***}p < 0.001$.

4.5. Prediction contribution of digital documents

We now explain how the proposed PPM models arrive at decisions. Specifically, we analyze the contribution of digital documents to the damage prediction in terms of SHAP values [Shapley Additive Explanations, 30]. SHAP values build on the theoretical concept of Shapley values from game theory [66], which quantifies the contribution of each player in a game to the game's outcome. Analogously, the SHAP value of a feature indicates the contribution of the feature to the prediction of a model [67].

We analyze the contribution of digital documents to the damage type prediction of VGG-RVL (i.e., the best model using visual features), and BERT-German (i.e., the best model using textual features). For this purpose, we focus on the predicted damage after the model has seen the last event of an event sequence as this ensures that SHAP values are calculated for all documents of a process instance. To measure the contribution of a document, we implement a masking function that replaces the document embedding with the learned embedding vector that indicates the absence of a document. We finally calculate the SHAP

Table 6
Predictive performance depending on next event.

Event type	Prevalence	Precision			Recall		
		Context-free	Context BERT-German	Imp. (%)	Context-free	Context BERT-German	Imp. (%)
Damage documents	0.222	0.591 ± 0.023	**0.604 ± 0.015**	2.19	0.609 ± 0.032	**0.644 ± 0.030**	5.83*
End of sequence	0.160	0.661 ± 0.031	**0.717 ± 0.014**	8.40***	0.674 ± 0.058	**0.730 ± 0.011**	8.42*
Processing	0.160	0.673 ± 0.023	**0.682 ± 0.022**	1.37	**0.978 ± 0.011**	0.954 ± 0.016	−2.44**
Notification of payment	0.120	0.397 ± 0.019	**0.437 ± 0.023**	10.0**	0.712 ± 0.060	**0.760 ± 0.042**	6.77*
Audit passed	0.103	0.488 ± 0.04	**0.558 ± 0.027**	14.4**	0.587 ± 0.071	**0.714 ± 0.022**	21.7***
Document request	0.074	0.459 ± 0.084	**0.614 ± 0.061**	33.8**	0.157 ± 0.051	**0.243 ± 0.039**	54.3**
Further inquiry	0.044	–	**0.138 ± 0.151**	–	0.014 ± 0.018	**0.016 ± 0.014**	16.9
Correspondence	0.033	0.225 ± 0.169	**0.231 ± 0.299**	2.6	**0.039 ± 0.031**	0.021 ± 0.017	−47.4
Cover note	0.024	–	–	–	0.001 ± 0.003	0.001 ± 0.005	41.2
Internal	0.023	–	–	–	–	–	–

Note: Number of process instances = 5131, number of events = 32,058, number of document pages = 39,242.
The results are given as mean ± standard deviation over ten-fold cross-validation. "–" denotes that the particular event was not predicted in at least one fold. Note that "notification of claim", "rejection", and "other" are never predicted by any model. Significance: $^{*}p < 0.05$,$^{**}p < 0.01$,$^{***}p < 0.001$.

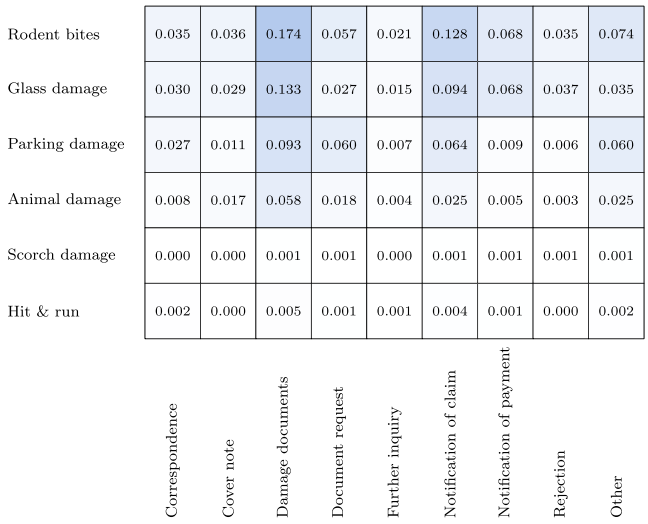


Fig. 8. Average absolute SHAP values of BERT-German model for documents linked to different event types for damage type prediction.

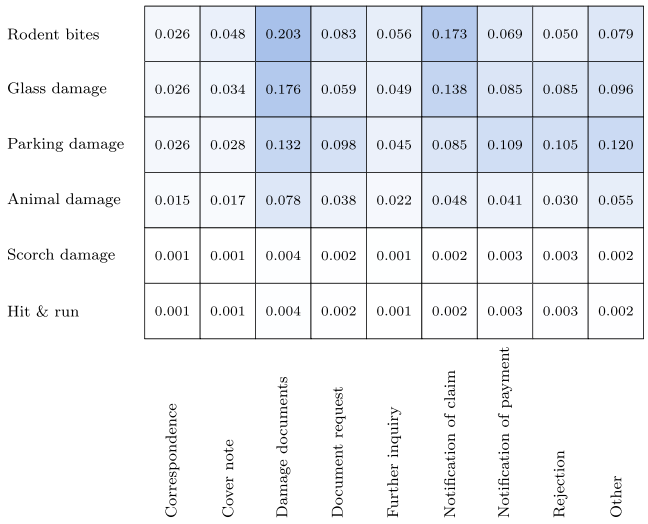


Fig. 9. Average absolute SHAP values of VGG-RVL model for documents linked to different event types for damage type prediction.

values of each document for the first fold of the dataset.[11] Recall that the outputs of the process model for damage type prediction is a softmax layer. This means that the model predicts a probability distribution over all damage types. Therefore, a SHAP value indicates the average change in the predicted probability of the particular damage type.

Figs. 8 and 9 show the average absolute SHAP values of documents that are associated with different event types for the BERT-German and VGG model, respectively. Darker colors indicate greater SHAP values. For instance, the value 0.174 in the third column of the first row of Fig. 8 indicates that, if a document is associated with the event "damage documents", it changes the predicted probability of a "rodent bite" damage by 0.174 on average. Overall, we observe that, if there are documents linked to the events "damage documents" and "notification of claim", they lead to the largest change in the predicted damage type probabilities. This intuitive result is consistent with the business rationale as documents from these two events should contain the most relevant information on the damage type. Note that the events "end of sequence", "processing", "audit passed", and "internal" are excluded from the plot as they are never linked to a document. Appendix F of the supplementary material contains several examples of SHAP values at the level of individual process instances.

5. Discussion

5.1. Implications for research

Our study has implications for research on BPM in general, and PPM in particular. The main goal of our study is to increase the focus of PPM research towards external context information outside of log data. So far, most prior studies focused on improving the performance of PPM models based on established benchmark datasets consisting of log data only (e.g., [9,36,47]). In contrast to this, we propose incorporating external context information given by digital documents. Digital documents are omnipresent in many business processes [20], which makes them a natural choice for PPM. Our evaluation, based on an insurance claims process from a German company, showed that incorporating digital documents as a complement to log data leads to significant performance improvements of between 4.1 and 22.1 percentage points. Therefore, following the concept of "data-centric AI" [18], we argue that PPM research should focus more on external context information, in particular, when its inclusion can be justified with a business rationale.

Incorporating external context information like digital documents also poses several challenges. Specifically, this requires automated feature extraction from unstructured data and the resulting features must be transformed into a single document embedding of fixed size. We followed prior studies by implementing the predictive model based on LSTM layers that process the log data at the event-level [8,11,27]. Accordingly, any context information must be converted into a fixed-size representation in order to be processed by the sequential predictive model. For this purpose, we evaluated a total of three feature extraction methods based on (i) visual features, (ii) textual features, and (iii) a hybrid of visual and textual features. Interestingly, we found that the hybrid approach, i.e., LayoutXLM [59], did not perform best. Instead, it was outperformed by the purely text-based approach based on a pretrained BERT model. We used a BERT model that was specifically pretrained on German texts, while LayoutXLM is pretrained on 30 million documents in 53 languages. Hence, future PPM research on external context information should consider the model architecture, the modality, and the data used for pretraining when selecting a suitable feature extractor.

Furthermore, we contribute to open research in BPM [34] by providing a novel benchmark dataset that includes digital documents. Although there is a large consensus about the benefit of context information [7,24,48,68], several established benchmark datasets (e.g., [33, 50]) consist of log data only. Here, we see a major challenge in publishing external context information from real-world data. For instance, digital documents often contain sensitive information like private addresses, bank accounts, or even business secrets. As a remedy, we propose publishing the materials in an anonymized way that is still useful for future studies. Specifically, we publish the extracted features from pretrained deep learning models like VGG-16, BERT, or LayoutXLM. This way of publication does not allow reconstruction of the original input, while textual and visual characteristics of the document are preserved. To our knowledge, this also presents the first PPM benchmark dataset containing anonymized digital documents. At the same time, we hope that future studies will also find ways to provide novel PPM benchmark datasets with external context information as machine learning studies largely benefit from multiple datasets to obtain more reliable estimates of the true out-of-sample performance and generalizability of the models (e.g., [8,9,69]).

[11] We use the Python package "shap" version 0.39.0 to calculate SHAP values.

Finally, the findings of our study can inform research in other subfields within BPM. Given that digital documents contain useful information for PPM, they could also complement log data in process mining [70,71], where business processes are discovered based on the given log data. Here, digital documents may help to better reconstruct the actual work and data flow, and discover richer explanations for deviations from the intended process behavior.

5.2. Implications for practice

The findings of our study also have implications for PPM practitioners. First, PPM practitioners should be aware of the potential benefits of complementing log data with external context information. In our study, we focused on an insurance claims management process where the inclusion of digital documents can directly be justified by a business rationale. For instance, if specific documents are present or missing, the clerk can either proceed to making a decision or must first make additional requests. Our evaluation showed considerable performance increases for instance-level and event-level predictions when log data is complemented by digital documents. In particular, we found that including digital documents provides process managers with more accurate instance-level predictions already in the early stages of a process but accuracy improves further after the model has seen a greater number of events and documents. This allows process managers to employ early countermeasures against unintended process behavior and ensure process compliance.

Second, PPM practitioners aiming to incorporate digital documents into their PPM models should evaluate multiple methods for feature extraction. We proposed a total of three methods to extract features automatically based on different visual and textual cues as it seems unlikely that process mangers can specify the relevant features in advance. Here, practitioners should consider their business rationale of including digital documents, and, if in doubt, evaluate multiple feature extraction methods as there is no universal "silver bullet" architecture in PPM that maximizes performance over all business processes [9].

5.3. Limitations and future research

Our study is not free from limitations. The benefit of incorporating digital documents in PPM models was evaluated using three methods for feature extraction and several simple baseline methods based on the number of document pages only. However, we did not specifically perform a comparison against a model that uses manually-selected and extracted features from digital documents. Using manually-selected features may result in even greater performance benefits. Furthermore, an analysis with manually-selected features might allow for deeper insights into how digital documents increase the predictive performance. At the same time, avoiding manual feature selection and extraction is an important advantage of the proposed approach.

Our study provides several avenues for future research. First, to gain further insights into the real-world benefit of including digital documents in PPM models, it seems intuitive to consider other datasets with digital documents from different processes. Second, the scope of the modeling task can be elevated to the level of process models, as proposed by Kim et al. [7]. Instead of predicting characteristics of individual process instances, PPM models could focus on the entire process to predict aggregated performance indicators like average waiting times. For this purpose, PPM models require an aggregated representation of the process. Here, digital documents could be used to enrich such representations in order to improve the predictive performance. Third, one could attempt to implement a two-staged feature selection process. As a first step, a model decides on whether to rely on visual or textual features. Based on this, the particular prediction model is chosen to make the actual predictions. At the same time, this would require an implementation of one predictive model for each prediction task and feature extractor. Fourth, future research could perform an in-depth analysis about what kind of information in digital documents improves the predictive performance. For this purpose, the parameters of the feature extractor could also be updated during the training process instead of relying on pretrained feature extractors. While this approach requires a considerably larger dataset than ours, there appear to be two major benefits. On the one hand, this may further improve the fixed-size document representation for the particular prediction task. On the other hand, methods from XAI (e.g., SHAP values) could be brought down to the level of individual words or even pixels, providing deeper insights into what kinds of information in digital documents improve the predictive performance.

6. Conclusion

Predictive process monitoring constitutes an important pillar for companies on their road to operational excellence. Being able to anticipate undesired anomalies in business processes and enact timely countermeasures can reduce costs and improve firm performance. In our study, we aimed to increase the focus of PPM research towards external context information outside of log data. Although context information has been shown to be an important determinant of future process behavior (e.g., [24,68]), prior research mainly focused on structured context features from the event logs only. Consistent with the concept of "data-centric AI", we proposed a tailored approach to incorporate digital documents into PPM models as they are omnipresent in many of today's business processes and their inclusion can often be justified by a business rationale. Our evaluation based on a claims management process of a mid-sized German insurance company showed considerable improvements in the predictive performance of PPM models when log data is complemented by features extracted from digital documents. We thus expect and hope for a more pronounced focus on external context information in future PPM research.

CRediT authorship contribution statement

Sergej Levich: Conceptualization, Methodology, Software, Formal analysis, Writing – original draft, Writing – review & editing. **Bernhard Lutz:** Conceptualization, Writing – original draft, Writing – review & editing. **Dirk Neumann:** Resources, Supervision.

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 data and code can be downloaded from [63] and https://github.com/serge724/ppm_docs.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.dss.2023.114043.

References

[1] W.M. van der Aalst, M. La Rosa, F.M. Santoro, Business process management: Don't forget to improve the process!, Bus. Inform. Syst. Eng. 58 (1) (2016) 1–6.
[2] J. vom Brocke, J. Mendling, Frameworks for business process management: A taxonomy for business process management cases, in: J. vom Brocke, J. Mendling (Eds.), Business Process Management, Springer International Publishing, Cham, 2018, pp. 1–17.
[3] J. Mendling, B. Baesens, A. Bernstein, M. Fellmann, Challenges of smart business process management: An introduction to the special issue, Decis. Support Syst. 100 (2017) 1–5.

[4] O. Müller, I. Junglas, J. vom Brocke, S. Debortoli, Utilizing big data analytics for information systems research: Challenges, promises and guidelines, Euro. J. Inform. Syst. 25 (4) (2016) 289–302.

[5] G. Shmueli, O.R. Koppius, Predictive analytics in information systems research, MIS Quart. 35 (3) (2011) 553–572.

[6] M. Harl, S. Weinzierl, M. Stierle, M. Matzner, Explainable predictive business process monitoring using gated graph neural networks, J. Decis. Syst. 29 (2020) 312–327.

[7] J. Kim, M. Comuzzi, M. Dumas, F.M. Maggi, I. Teinemaa, Encoding resource experience for predictive process monitoring, Decis. Support Syst. 153 (113669) (2021).

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

[9] K. Heinrich, P. Zschech, C. Janiesch, M. Bonin, Process data properties matter: Introducing gated convolutional neural networks (GCNN) and key-value-predict attention networks (KVP) for next event prediction with deep learning, Decis. Support Syst. 143 (113494) (2021).

[10] A. Rogge-Solti, M. Weske, Prediction of business process durations using non-Markovian stochastic Petri nets, Inf. Syst. 54 (2015) 1–14.

[11] N. Tax, I. Verenich, M. La Rosa, M. Dumas, Predictive business process monitoring with LSTM neural networks, in: E. Dubois, K. Pohl (Eds.), Conference on Advanced Information Systems Engineering, vol. 10253, Springer International Publishing, Cham, 2017, pp. 477–492.

[12] D. Breuker, M. Matzner, P. Delfmann, J. Becker, Comprehensible predictive models for business processes, MIS Quart. 40 (4) (2016) 1009–1034.

[13] A.E. Márquez-Chamorro, M. Resinas, A. Ruiz-Cortés, Predictive monitoring of business processes: A survey, IEEE Trans. Serv. Comput. 11 (6) (2017) 962–977.

[14] N. Mehdiyev, J. Evermann, P. Fettke, A novel business process prediction model using a deep learning method, Bus. Inform. Syst. Eng. 62 (2) (2020) 143–157.

[15] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.

[16] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, vol. 25, 2012, pp. 1097—1105.

[17] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: Y. Bengio, Y. LeCun (Eds.), International Conference on Learning Representations, 2015, pp. 1–14.

[18] D. Zha, Z.P. Bhat, K.-H. Lai, F. Yang, Z. Jiang, S. Zhong, X. Hu, Data-centric artificial intelligence: A survey, 2023, Available at https://arxiv.org/abs/2303.10158.

[19] A.E. Márquez-Chamorro, K. Revoredo, M. Resinas, A. Del-Río-Ortega, F.M. Santoro, A. Ruiz-Cortés, Context-aware process performance indicator prediction, IEEE Access 8 (2020) 222050–222063.

[20] S.X. Sun, J.L. Zhao, J.F. Nunamaker, O.R.L. Sheng, Formulating the data-flow perspective for business process management, Inf. Syst. Res. 17 (4) (2006) 374–391.

[21] J. vom Brocke, M.-S. Baier, T. Schmiedel, K. Stelzl, M. Röglinger, C. Wehking, Context-aware business process management: method assessment and selection, Bus. Inform. Syst. Eng. 63 (2020) 533—550.

[22] I. Teinemaa, M. Dumas, F.M. Maggi, C. Di Francescomarino, Predictive business process monitoring with structured and unstructured data, in: M. La Rosa, P. Loos, O. Pastor (Eds.), Business Process Management, vol. 9850, Springer International Publishing, Cham, 2016, pp. 401–417.

[23] A. Yeshchenko, F. Durier, K. Revoredo, J. Mendling, F. Santoro, Context-aware predictive process monitoring: THe impact of news sentiment, in: H. Panetto, C. Debruyne, H.A. Proper, C.A. Ardagna, D. Roman, R. Meersman (Eds.), On the Move To Meaningful Internet Systems. OTM 2018 Conferences, vol. 11229, Springer International Publishing, Cham, 2018, pp. 586–603.

[24] J. Brunk, J. Stottmeister, S. Weinzierl, M. Matzner, J. Becker, Exploring the effect of context information on deep learning business process predictions, J. Decis. Syst. 29 (2020) 328–343.

[25] G.T. Lakshmanan, D. Shamsi, Y.N. Doganata, M. Unuvar, R. Khalaf, A Markov prediction model for data-driven semi-structured business processes, Knowl. Inf. Syst. 42 (1) (2015) 97–126.

[26] Z. Huang, A. Kumar, A study of quality and accuracy trade-offs in process mining, INFORMS J. Comput. 24 (2) (2012) 311–327.

[27] M. Borkowski, W. Fdhila, M. Nardelli, S. Rinderle-Ma, S. Schulte, Event-based failure prediction in distributed business processes, Inf. Syst. 81 (2019) 220–235.

[28] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: PRe-training of deep bidirectional transformers for language understanding, 2018, Available at https://arxiv.org/abs/1810.04805.

[29] Y. Xu, Y. Xu, T. Lv, L. Cui, F. Wei, G. Wang, Y. Lu, D. Florencio, C. Zhang, W. Che, M. Zhang, L. Zhou, LayoutLMv2: Multi-modal pre-training for visually-rich document understanding, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, 2021, pp. 2579–2591.

[30] S.M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, Adv. Neural Inf. Process. Syst. 30 (2017) 4765–4774.

[31] B.F. van Dongen, BPI challenge 2012, 2012, Available at https://doi.org/10.4121/uuid:3926db30-f712-4394-aebc-75976070e91f.

[32] W. Steeman, BPI challenge 2013, 2013, Available at https://doi.org/10.4121/uuid:a7ce5c55-03a7-4583-b855-98b86e1a2b07.

[33] I. Verenich, Helpdesk, 2016, Available at https://doi.org/10.17632/39bp3vv62t.1.

[34] W. van der Aalst, M. Bichler, A. Heinzl, Open research in business and information systems engineering, Bus. Inform. Syst. Eng. 58 (6) (2016) 375–379.

[35] A.E. Márquez-Chamorro, M. Resinas, A. Ruiz-Cortés, Predictive monitoring of business processes: A survey, IEEE Trans. Serv. Comput. 11 (6) (2018) 962–977.

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

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

[38] W. Kratsch, J. Manderscheid, M. Röglinger, J. Seyfried, Machine learning in business process monitoring: A comparison of deep learning and classical approaches used for outcome prediction, Bus. Inform. Syst. Eng. 63 (2020) 261–276.

[39] S. Verwer, R. Eyraud, C. De La Higuera, Pautomac: A probabilistic automata and hidden Markov models learning competition, Mach. Learn. 96 (1) (2014) 129–154.

[40] A.E. Márquez-Chamorro, M. Resinas, A. Ruiz-Cortés, M. Toro, Run-time prediction of business process indicators using evolutionary decision rules, Expert Syst. Appl. 87 (2017) 1–14.

[41] B.A. Tama, M. Comuzzi, An empirical comparison of classification techniques for next event prediction using business process event logs, Expert Syst. Appl. 129 (2019) 233–245.

[42] A. Leontjeva, R. Conforti, C. Di Francescomarino, M. Dumas, F.M. Maggi, Complex symbolic sequence encodings for predictive monitoring of business processes, in: H.R. Motahari-Nezhad, J. Recker, M. Weidlich (Eds.), Business Process Management, vol. 9253, Springer International Publishing, Cham, 2015, pp. 297–313.

[43] C. Di Francescomarino, C. Ghidini, F.M. Maggi, G. Petrucci, A. Yeshchenko, An eye into the future: Leveraging A-priori knowledge in predictive business process monitoring, in: J. Carmona, G. Engels, A. Kumar (Eds.), Business Process Management, vol. 10445, Springer International Publishing, Cham, 2017, pp. 252–268.

[44] M. Camargo, M. Dumas, O. González-Rojas, Learning accurate LSTM models of business processes, in: T. Hildebrandt, B.F. van Dongen, M. Röglinger, J. Mendling (Eds.), Business Process Management, vol. 11675, Springer International Publishing, Cham, 2019, pp. 286–302.

[45] N. Di Mauro, A. Appice, T.M.A. Basile, Activity prediction of business process instances with inception CNN models, in: M. Alviano, G. Greco, F. Scarcello (Eds.), Advances in Artificial Intelligence, vol. 11946, Springer International Publishing, Cham, 2019, pp. 348–361.

[46] V. Pasquadibisceglie, A. Appice, G. Castellano, D. Malerba, Using convolutional neural networks for predictive process analytics, in: International Conference on Process Mining, IEEE, Aachen, Germany, 2019, pp. 129–136.

[47] F. Taymouri, M. La Rosa, S. Erfani, Z.D. Bozorgi, I. Verenich, Predictive business process monitoring via generative adversarial nets: The case of next event prediction, in: International Conference on Business Process Management, Springer, 2020, pp. 237–256.

[48] J. Brunk, M. Stierle, L. Papke, K. Revoredo, M. Matzner, J. Becker, Cause vs. Effect in context-sensitive prediction of business process instances, Inf. Syst. 95 (2021) 101635.

[49] N. Mehdiyev, J. Evermann, P. Fettke, A multi-stage deep learning approach for business process event prediction, in: IEEE Conference on Business Informatics, IEEE, 2017, pp. 119–128.

[50] I.F. Lopes, D.R. Ferreira, A survey of process mining competitions: THe BPI challenges 2011–2018, in: International Conference on Business Process Management, Springer, 2019, pp. 263–274.

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

[52] Y. Bengio, Deep learning of representations for unsupervised and transfer learning, in: Proceedings of ICML Workshop on Unsupervised and Transfer Learning, JMLR Workshop and Conference Proceedings, 2012, pp. 17–36.

[53] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016.

[54] A. Das, S. Roy, U. Bhattacharya, S.K. Parui, Document image classification with intra-domain transfer learning and stacked generalization of deep convolutional neural networks, in: IEEE International Conference on Pattern Recognition, IEEE, 2018, pp. 3180–3185.

[55] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 248–255.
[56] A.W. Harley, A. Ufkes, K.G. Derpanis, Evaluation of deep convolutional nets for document image classification and retrieval, in: IEEE International Conference on Document Analysis and Recognition, IEEE, 2015, pp. 991–995.
[57] T. Trinh, A. Dai, T. Luong, Q. Le, Learning longer-term dependencies in RNNs with auxiliary losses, in: J. Dy, A. Krause (Eds.), International Conference on Machine Learning, 2018, pp. 4965–4974.
[58] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in Neural Information Processing Systems, Neural Information Processing Systems Foundation, La Jolla, CA, 2013, pp. 3111–3119.
[59] Y. Xu, T. Lv, L. Cui, G. Wang, Y. Lu, D. Florêncio, C. Zhang, F. Wei, LayoutXLM: Multimodal pre-training for multilingual visually-rich document understanding, 2021, Available at https://arxiv.org/abs/2104.08836.
[60] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.
[61] M. Kraus, S. Feuerriegel, Decision support from financial disclosures with deep neural networks and transfer learning, Decis. Support Syst. 104 (2017) 38–48.
[62] R. Gruetzemacher, D. Paradice, Deep transfer learning & beyond: Transformer language models in information systems research, ACM Comput. Surv. 54 (10s) (2022).
[63] S. Levich, Claims management log dataset with digital documents, 2023, Available at https://doi.org/10.17632/kdcspz6xtn.1.
[64] T. Tieleman, G. Hinton, et al., Lecture 6.5–Rmsprop: Divide the gradient by a running average of its recent magnitude, COURSERA Neural Netw. Mach. Learn. 4 (2) (2012) 26–31.
[65] F. Wilcoxon, Individual comparisons by ranking methods, Biom. Bull. 1 (6) (1945) 80–83.
[66] L.S. Shapley, A value for N-person games, Contribut. Theory Games 2 (28) (1953) 307–317.
[67] J. Senoner, T. Netland, S. Feuerriegel, Using explainable artificial intelligence to improve process quality: Evidence from semiconductor manufacturing, Manage. Sci. 68 (8) (2022) 5704–5723.
[68] T.d.C. Mattos, F.M. Santoro, K. Revoredo, V.T. Nunes, A formal representation for context-aware business processes, Comput. Ind. 65 (8) (2014) 1193–1214.
[69] B. Padmanabhan, N. Sahoo, A. Burton-Jones, et al., Machine learning in information systems research, MIS Quart. 46 (1) (2022) iii–xix.
[70] H. Chen, R.H. Chiang, V.C. Storey, Business intelligence and analytics: From big data to big impact, MIS Quart. 36 (4) (2012) 1165–1188.
[71] W. van der Aalst, Discovery, Conformance and Enhancement of Business Processes, Germany, Springer, 2011.

Sergej Levich is a Ph.D. candidate at the Chair of Information Systems of the University of Freiburg, Germany. His research interest lies in the application of Machine Learning to operational processes in organizations. He holds a bachelor's degree in Business Administration from the University of St. Gallen and a master's degree in Decision Sciences from the London School of Economics.

Bernhard Lutz is a postdoctoral researcher at the Chair for Information Systems Research of the University of Freiburg. His research focuses on applied machine learning in the area of data analytics. Previously, he completed his Ph.D. in Information Systems at the University of Freiburg. He has coauthored research publications for the European Journal of Operational Research, OR Spectrum, Electronic Markets, Information Sciences, Journal of Business Research, and Expert Systems with Applications.

Dirk Neumann is Full Professor with the Chair of Information Systems of the University of Freiburg, Germany. His research topics include Business Analytics, Text Mining and Cloud Computing. He studied information systems in Giessen (Diploma), Economics in Milwaukee, WI, USA (Master) and received a Ph.D. from Karlsruhe Institute of Technology (KIT) in 2004. Among others, he has (co-)authored research publications at Operations Research, European Journal of Operational Research, Decision Support Systems, Journal of Management Information Systems, and Information & Management.