Text-Aware Predictive Monitoring of Business Processes*

Marco Pegoraro ✉[0000−0002−8997−7517], Merih Seran Uysal[0000−0003−1115−6601], David Benedikt Georgi, and Wil M.P. van der Aalst[0000−0002−0955−6940]

Process and Data Science chair, RWTH Aachen University, Aachen, Germany
{pegoraro, uysal, wvdaalst}@pads.rwth-aachen.de,
david.georgi@rwth-aachen.de
http://www.pads.rwth-aachen.de/

Abstract. The real-time prediction of business processes using historical event data is an important capability of modern business process monitoring systems. Existing process prediction methods are able to also exploit the data perspective of recorded events, in addition to the control-flow perspective. However, while well-structured numerical or categorical attributes are considered in many prediction techniques, almost no technique is able to utilize text documents written in natural language, which can hold information critical to the prediction task. In this paper, we illustrate the design, implementation, and evaluation of a novel *text-aware process prediction model* based on *Long Short-Term Memory* (LSTM) neural networks and natural language models. The proposed model can take categorical, numerical and textual attributes in event data into account to predict the activity and timestamp of the next event, the outcome, and the cycle time of a running process instance. Experiments show that the text-aware model is able to outperform state-of-the-art process prediction methods on simulated and real-world event logs containing textual data.

Keywords: Predictive Monitoring · Process Mining · Natural Language Processing · LSTM Neural Networks.

1 Introduction

In recent years, a progressive and rapid tendency to digital transformation has become apparent in most aspects of industrial production, provision of services, science, education, and leisure. This has, in turn, caused the widespread adoption of new technologies to support human activities. A significant number of these technologies specialize in the management of enterprise business processes.

The need of analysis and compliance in business processes, united to a larger and larger availability of historical event data have stimulated the birth and growth of the scientific discipline of *process mining*. Process mining enables the

---

discovery of process models from historical execution data, the measurement of compliance between data and a process model, and the enhancement of process models with additional information extracted from complete process cases.

Advancements in process mining and other branches of data science have also enabled the possibility of adopting *prediction* techniques, algorithms that train a mathematical model from known data instances and are able to perform accurate estimates of various features of future instances. In the specific context of process mining, *predictive monitoring* is the task of predicting features of *partial process instances*, i.e., cases of the process still in execution, on the basis of recorded information regarding complete process instances. Examples of valuable information on partial process instances are the next activity in the process to be executed for the case, the time until the next activity, the completion time of the entire process instance, and the last activity in the case (outcome). If accurately estimated, these case features can guide process owners in making vital decisions, and improve operations within the organization that hosts the process; as a result, accurate predictive monitoring techniques are widely desirable and a precious asset for companies and organizations.

Existing predictive monitoring techniques typically operate at the merging point between process mining and machine learning, and are able to consider not only the control-flow perspective of event data (i.e., the activity, the case identifier, and the timestamp), but also additional data associated with them. However, few prediction techniques are able to exploit attributes in the form of text associated with events and cases. These textual attributes can hold crucial information regarding a case and its status within the workflow of a process. A general framework describing the problem is shown in Figure 1.
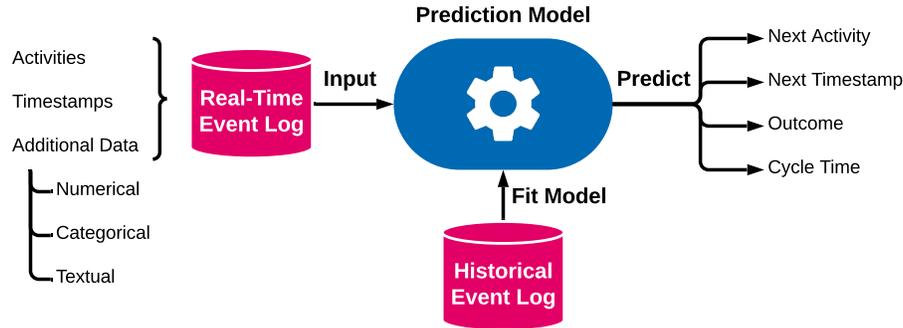

Fig. 1: Problem overview: a general predictive monitoring model. The aim is predicting features of running process instances based on historical data, by exploiting numerical, categorical, and textual data.

The aim of this paper is to assess the extent to which textual information can influence predictive monitoring. To this end, we present a novel predictive monitoring approach able to exploit numerical, categorical, and textual attributes

associated with events, as well as control-flow information. Our prediction model estimates features of cases in execution by combining a set of techniques for sequential and textual data encoding with predictions from an LSTM neural network, a machine learning technique particularly effective on sequential data such as process traces. Validation through experiments on real-life event logs shows that our approach is effective in extracting additional information from textual data, and outperforms state-of-the-art approaches for predictive monitoring.

The remainder of the paper is structured as follows. Section 2 discusses some recent work related to predictive monitoring. Section 3 presents some preliminary definitions. Section 4 illustrates the details and architecture of our text-aware predictive monitoring technique. Section 5 presents the evaluation of the predictor and the results of the experiments. Section 6 concludes the paper.

2   Related Work

The intersection of process mining and machine learning is a rich and influential field of research. Among the numerous applications of machine learning in process mining, feature prediction on partial process traces based on historical complete traces (i.e., predictive monitoring) is particularly prominent.

Earlier techniques for prediction in process mining focused on white-box and human-interpretable models, largely drawn from statistics. Many proposals have been put forward to compute an estimate of the cycle time of a process instance, including decision trees [6] and simulation through stochastic Petri nets [13]. Additionally, Teinemaa et al. [16] proposed a process outcome prediction method based on random forests and logistic regression. Van der Aalst et al. [1] exploit process discovery as a step of the prediction process, obtaining estimations through replay on an annotated transition system; this technique is then extended by Polato et al. [12] by annotating a discovered transition system with an ensemble of naïve Bayes and support vector regressors, allowing for the data-aware prediction of cycle time and next activity.

The second half of the 2010s saw a sharp turn from ensemble learning to single prediction models, and from white-box to black-box models – specifically, recurrent neural networks. This is due to the fact that recurrent neural networks have been shown to be very accurate in learning from sequential data. However, they are not interpretable, and the training efficiency is often lower.

This family of prediction methods employs LSTM neural networks to estimate process instance features. Evermann et al. [7] proposed the use of LSTMs for next activity prediction; Tax et al. [15] trained LSTMs to predict cycle time of process instances. Navarin et al. [9] extended this approach by feeding additional attributes in the LSTM, attaining data-aware prediction. More recently, Park and Song [10] merged system-level information from a process model with a compact trace representation based on deep neural networks to attain performance prediction.

No existing predictive monitoring technique, to the best of our knowledge, incorporates information from free text, recorded as event or trace attribute,

with the control-flow perspective of the process into a state-of-the-art LSTM neural network model for predictive monitoring: this motivates the approach we present in this paper.

3   Preliminaries

Let us first introduce some preliminary definitions and notations.

Definition 1 (Sequence). *A* sequence *of length $n \in \mathbb{N}_0$ over a set X is an ordered collection of elements defined by a function $\sigma\colon \{1,\ldots,n\} \to X$, which assigns each index an element of X. A sequence of length n is represented explicitly as $\sigma = \langle x_1, x_2, \ldots, x_n \rangle$ with $x_i \in X$ for $1 \leq i \leq n$. In addition, $\langle\ \rangle$ is the empty sequence of length 0. Over the sequence σ we define $|\sigma| = n$, $\sigma(i) = x_i$, and $x \in \sigma \Leftrightarrow \exists_{1 \leq i \leq n}\colon x = x_i$. X^* denotes the set of all sequences over X.*

The function $hd^k\colon X^* \to X^*$ gives the head or prefix of length k of σ for $0 \leq k \leq n$: $hd^k(\sigma) = \langle x_1, x_2, \ldots, x_k \rangle$. For instance, $hd^2(\sigma) = \langle x_1, x_2 \rangle$.

Definition 2 (Event, Trace, Event Log, Prefix Log). *Let $\mathcal{A}$ be the universe of* activity labels*. Let $\mathcal{T}$ be the closed under subtraction and totally ordered universe of* timestamps*. Let $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_m$ be the domains of* additional attributes*. An* event *is a tuple $e = (a, t, d_1, \ldots, d_m) \in \mathcal{A} \times \mathcal{T} \times \mathcal{D}_1 \times \cdots \times \mathcal{D}_m = \mathcal{E}$. Over an event e we define the projection functions $\pi_{\mathcal{A}}(e) = a$, $\pi_{\mathcal{T}}(e) = t$, and $\pi_{\mathcal{D}_i}(e) = d_i$. A trace $\sigma \in \mathcal{E}^*$ is a sequence of events such that timestamps are non-decreasing: $\pi_{\mathcal{T}}(e_i) \leq \pi_{\mathcal{T}}(e_j)$ for $1 \leq i < j \leq |\sigma|$. An* event log $L \in \mathcal{B}(\mathcal{E}^*)$ *is a multiset of traces. Given an event log L, we define the* prefix log $\mathbb{L} = \{hd^k(\sigma) \mid \sigma \in L \wedge 1 \leq k \leq |\sigma|\}$.*

Additional attributes $d_i \in \mathcal{D}_i$ may be in the form of text, i.e., its domain is the set of sequences $\mathcal{D}_i = \Sigma^*$ from a fixed and known alphabet Σ.

Next, let us define the target functions for our predictions:

Definition 3 (Target Functions). *Let $\sigma \in \mathcal{E}^*$ be a non-empty trace, and let $1 \leq k \leq |\sigma|$. The* next activity *function $f_{\mathrm{a}}\colon \mathcal{E}^* \times \mathbb{N} \to \mathcal{A} \cup \{\blacksquare\}$ returns the activity of the next event, or an artificial activity $\blacksquare$ if the given trace is complete:*

$$f_{\mathrm{a}}(\sigma, k) = \begin{cases} \blacksquare & \text{if } k = |\sigma| \\ \pi_{\mathcal{A}}(\sigma(k+1)) & \text{else} \end{cases}$$

The next timestamp *function $f_{\mathrm{t}}\colon \mathcal{E}^* \times \mathbb{N} \to \mathcal{T}$ returns the time difference between the next event and last event in the prefix:*

$$f_{\mathrm{t}}(\sigma, k) = \begin{cases} 0 & \text{if } k = |\sigma| \\ \pi_{\mathcal{T}}(\sigma(k+1)) - \pi_{\mathcal{T}}(\sigma(k)) & \text{else} \end{cases}$$

The case outcome *function $f_{\mathrm{o}}\colon \mathcal{E}^* \to \mathcal{A}$ returns the last activity of the trace: $f_{\mathrm{o}}(\sigma) = \pi_{\mathcal{A}}(\sigma(|\sigma|))$. The* cycle time *function $f_{\mathrm{c}}\colon \mathcal{E}^* \to \mathcal{T}$ returns the total duration of the case, i.e., the time difference between the first and the last event of the trace: $f_{\mathrm{c}}(\sigma) = \pi_{\mathcal{T}}(\sigma(|\sigma|)) - \pi_{\mathcal{T}}(\sigma(1))$.*

The prediction techniques we show include the information contained in textual attributes of events. In order to be readable by a prediction model, the text needs to be processed by a *text model*. Text models rely on a *text corpus*, a collection of text fragments called *documents*. Before computing the text model, the documents in the corpus are preprocessed with a number of normalization steps: conversion to lowercase, tokenization (separation in distinct terms), lemmatization (mapping words with similar meaning, such as "diagnose" and "diagnosed" into a single lemma), and stop word removal (deletion of uninformative parts of speech, such as articles and adverbs). These transformation steps are shown in Table 1.

Table 1: Text preprocessing transformation of an example document containing a single sentence.

Step	Transformation	Example Document
0	Original	"The patient has been diagnosed with high blood pressure."
1	Lowercase	"the patient has been diagnosed with high blood pressure."
2	Tokenization	⟨"the", "patient", "has", "been", "diagnosed", "with", "high", "blood", "pressure", "."⟩
3	Lemmatization	⟨"the", "patient", "have", "be", "diagnose", "with", "high", "blood", "pressure", "."⟩
4	Stop word filtering	⟨"patient", "diagnose", "high", "blood", "pressure"⟩

In order to represent text in a structured way, we consider four different text models:

Bag of Words (BoW) [5]: a model where, given a vocabulary V, we encode a document with a vector of length $|V|$ where the i-th component is the *term frequency* (tf), the number of occurrences of the i-th term in the vocabulary, normalized with its *inverse document frequency* (idf), the inverse of the number of documents that contain the term. This tf-idf score accounts for term specificity and rare terms in the corpus. This model disregards the order between words.

Bag of N-Grams (BoNG) [5]: this model is a generalization of the BoW model. Instead of one term, the vocabulary consists of n-tuples of consecutive terms in the corpus. The unigram model ($n = 1$) is equivalent to the BoW model. For the bigram model ($n = 2$), the vocabulary consists of pairs of words that appear next to each other in the documents. The documents are encoded with the td-idf scores of their n-grams. This model is able to account for word order.

Paragraph Vector (Doc2Vec) [8]: in this model, a feedforward neural network is trained to predict one-hot encodings of words from their context, i.e., words that appear before or after the target word in the training documents. An additional vector, of a chosen size and unique for each document, is trained together with the word vectors. When the network converges, the additional vector carries information regarding the words in the corresponding document and their relationship, and is thus a fixed-length representation of the document.

Latent Dirichlet Allocation (LDA) [4]: a generative statistical text model, representing documents as a set of topics, which size is fixed and specified a priori. Topics are multinomial (i.e., categorical) probability distributions over all words in the vocabulary and are learned by the model in an unsupervised manner. The underlying assumption of the LDA model is that the text documents were created by a statistical process that first samples topic from a multinomial distribution associated with a document, then sample words from the sampled topics. Using the LDA model, a document is encoded as a vector by its topic distribution: each component indicates the probability that the corresponding topic was chosen to sample a word in the document. LDA does not account for word order.

In the next section, we will describe the use of text models in an architecture allowing to process a log to obtain a data- and text-aware prediction model.

4 Prediction Model Architecture

The goal of predictive monitoring is to estimate a target feature of a running process instance based on historical execution data. In order to do so, predictive monitoring algorithms examine *partial traces*, which are the events related to a process case at a certain point throughout its execution. Obtaining partial traces for an event log is equivalent to computing the set of all prefixes for the traces in the log. Prefix logs will be the basis for training our predictive model.

In this paper, we specifically address the challenge of managing additional attributes that are textual in nature. In order to account for textual information, we need to define a construction method for fixed-length vectors that encode activity labels, timestamps, and numerical, categorical, and textual attributes.

Given an event $e = (a, t, d_1, \ldots, d_m)$, its activity label a is represented by a vector $\vec{a}$ using *one-hot encoding*. Given the set of possible activity labels $\mathcal{A}$, an arbitrary but fixed ordering over $\mathcal{A}$ is introduced with a bijective index function $index_{\mathcal{A}} \colon \mathcal{A} \to \{1, \ldots, |\mathcal{A}|\}$. Using this function, the activity is encoded as a vector of size $|\mathcal{A}|$, where the component $index_{\mathcal{A}}(\pi_{\mathcal{A}}(e))$ has value 1 and all the other components have value 0. The function $\mathbb{1}_{\mathcal{A}} \colon \mathcal{A} \to \{0, 1\}^{\mathcal{A}}$ is used to describe the realization of such one-hot encoding $\vec{a} = \mathbb{1}_{\mathcal{A}}(\pi_{\mathcal{A}}(e))$ for the activity label of the event e.

In order to capture time-related correlations, a set of time-based features is utilized to encode the timestamp t of the event. We compute a time vector $\vec{t} = (\hat{t}_1, \hat{t}_2, \hat{t}_3, \hat{t}_4, \hat{t}_5, \hat{t}_6)$ of min-max normalized time features, where t_1 is the time since the previous event, t_2 is the time since the first event of the case, t_3 is the time since the first event of the log, t_4 is the time since midnight, t_5 is the time since previous Monday, and t_6 is the time since the first of January. The min-max normalization is obtained through the formula

$$\hat{x} = \frac{x - \min(x)}{\max(x) - \min(x)}$$

where $\min(x)$ is the lowest and $\max(x)$ is the highest value for the attribute x.

Every additional attribute d_i of e is encoded in a vector $\vec{d_i}$ as follows:

$$\vec{d_i} = \begin{cases} \mathbb{1}_{\mathcal{D}_i}(d_i) & \text{if } \mathcal{D}_i \text{ is categorical} \\ \hat{d}_i & \text{if } \mathcal{D}_i \text{ is numerical} \\ \textsc{TextModel}(d_1) & \text{if } \mathcal{D}_i \text{ is textual} \end{cases}$$

The encoding technique depends on the type of the attribute. Categorical attributes are one-hot encoded similarly to the activity label. Numerical attributes are min-max normalized: if the minimum and maximum are not bounded conceptually, the lowest or highest value of the attribute in the historical event log is used for scaling. Finally, if $\mathcal{D}_i$ is a textual model, it is encoded in a fixed-length vector with one of the four text models presented in Section 3; the documents in the text corpus for the text model consist of all instances of the textual attribute $\mathcal{D}_i$ contained in the historical log. This technique allows to build a complete fixed-length encoding for the event $e = (a, t, d_1, \ldots, d_m)$, which we indicate with the tuple of vectors $enc(e) = (\vec{a}, \vec{t}, \vec{d_1}, \ldots, \vec{d_m})$.

This encoding procedure allows us to build a training set for the prediction of the target functions presented in Section 3 utilizing an LSTM neural network.

Figure 2 illustrates the entire encoding architecture, and the fit/predict pipeline for our final LSTM model. The schematic distinguishes between the *offline* (fitting) phase, where we train the LSTM with encoded historical event data, and the *online* (real-time prediction) phase, where we utilize the trained model to estimate the four target features on running process instances. Given an event log L, the structure of the training set is based on the partial traces in its prefix log $\mathbb{L} = \{hd^k(\sigma) \mid \sigma \in L \wedge 1 \leq k \leq |\sigma|\}$. For each $\sigma = \langle e_1, e_2, \ldots e_n \rangle \in L$ and $1 \leq k \leq n$, we build an instance of the LSTM training set. The network input $\langle \vec{vecx_1}, \vec{x_2}, \ldots, \vec{x_k} \rangle$ is given by the event encodings $vecx_1 = enc(e_1)$, $vecx_2 = enc(e_2)$, through $vecx_k = enc(e_k)$. The targets $(\vec{y_a}, y_t, \vec{y_o}, y_c)$ are given by $\vec{y_a} = f_a(\sigma, k)$, $\vec{y_t} = f_t(\sigma, k)$, $\vec{y_o} = f_o(\sigma)$, and $y_c = f_c(\sigma, k)$.

Figure 3 shows the topology of the network. The training utilizes gradient descent and backpropagation through time (BPTT). The loss for numerical prediction values $\hat{y}$ and the true value y is the absolute error $\text{AE}(\hat{y}, y) = |\hat{y} - y|$, while the loss for categorical prediction values is computed using the categorical cross-entropy error $\text{CE}(\vec{\hat{y}}, \vec{y}) = -\sum_{i=1}^{k} y_i \cdot \log \hat{y}_i$.

5   Evaluation

The predictive monitoring approach presented in this paper has been implemented for validation, utilizing a Python-based, fully open-source technological stack. PM4Py [3] is a process mining Python tool developed by Fraunhofer FIT. It is used for event log parsing and its internal event log representation. The neural network framework Tensorflow [2], originally developed by Google, and its API Keras[1] were utilized to implement the final LSTM model. Furthermore,

---

[1] https://keras.io/
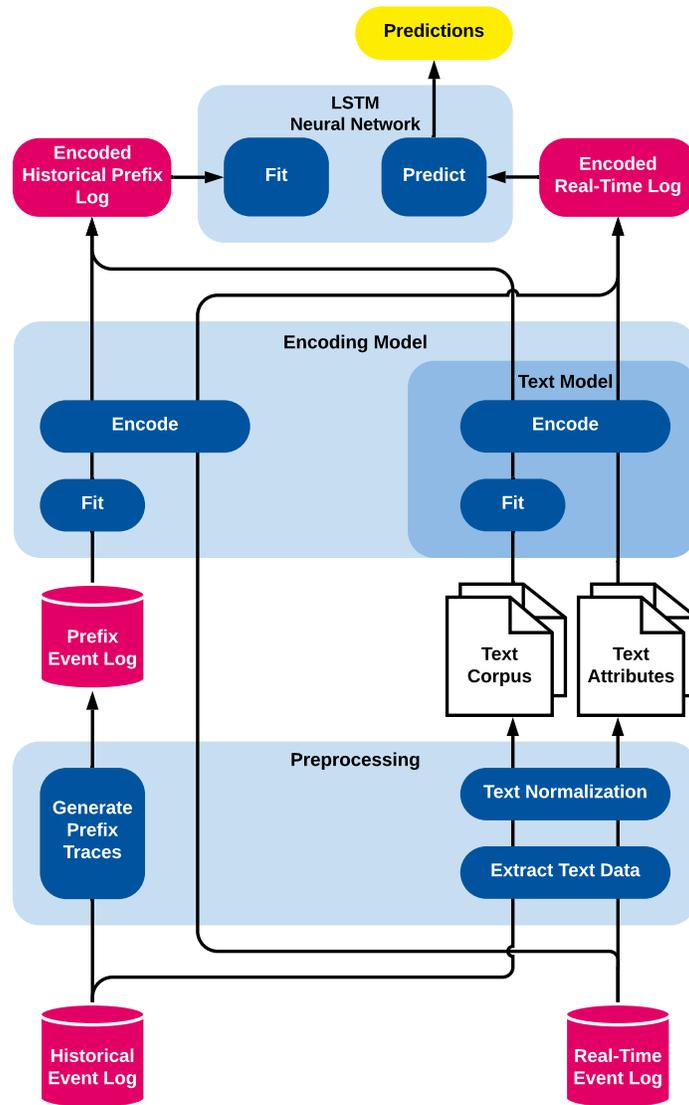
Fig. 2: Overview of the text-aware process prediction model. Predictions for real-time processes are realized by an LSTM model that is fitted using an encoded representation of all prefixes of the historical event log. The encoding of textual attributes is realized by a text preprocessing pipeline and an exchangeable text encoding model.

the libraries Scikit-learn [11], NLTK[2], and Gensim[3] provided the natural lan-

---

[2] https://nltk.org/
[3] https://radimrehurek.com/gensim/
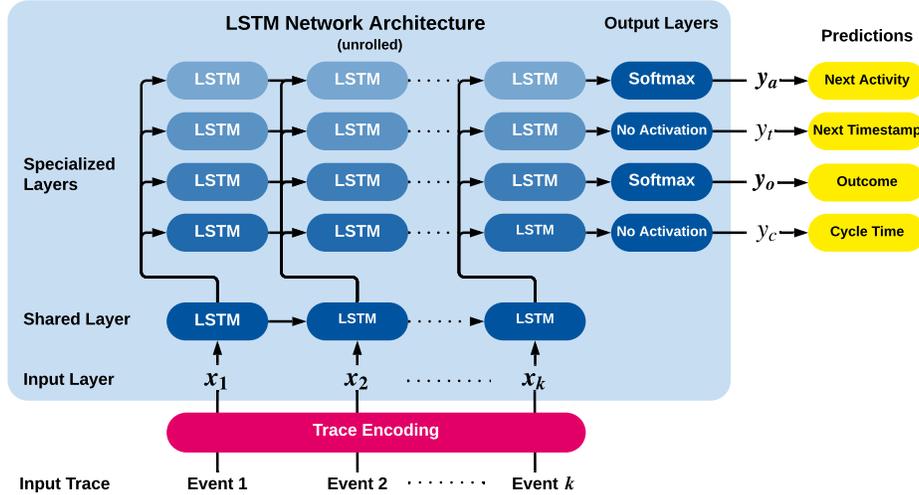
Fig. 3: LSTM model architecture to simultaneously predict the next activity $(\vec{y_a})$, next event time (y_t), outcome $(\vec{y_o})$ and cycle time (y_c) for an encoded prefix trace $\vec{x_1}, \vec{x_2}, \ldots, \vec{x_k}$.

guage processing capabilities required to preprocess and normalize text, as well as build and train the text models.

The text-aware model is compared to two other process prediction methods. First, the pure LSTM approach based on the ideas of Navarin et al. [9] is considered, which only uses the activity, timestamp, and additional non-textual attributes of each event. This approach can be considered the state of the art in predictive monitoring with respect to prediction accuracy. The second baseline is the process model-based prediction method originally presented by van der Aalst et al. [1]. This approach builds an annotated transition system for a log using a sequence, bag, or set abstraction. Each state of the transition system is annotated with measurements of historical traces that can be used to predict target values for unseen traces. During the prediction phase, running traces are mapped to the corresponding state of the transition system, and the measurements of the state are used to compute a prediction. We adopt the improvement of this method described in [14] to apply it to classification tasks and obtain the next activity and outcome predictions. The first 8 events of a trace are considered for the construction of the state space. Experiments with different horizon lengths (1, 2, 4, 16) mostly led to inferior results, and are thus not reported.

We evaluate the two baseline methods against our approach considering all four text models presented here, with a varying dimension of vector size (50, 100 and 500 for BoW and BoNG, 10, 20 and 100 for PV and LDA). The BoNG model is built with bigrams $(n = 2)$. Of the four target functions presented in Section 3, classification tasks (next activity and outcome) are evaluated with a weighted-average class-wise F_1 score; regression tasks (next timestamp and cycle time) are evaluated on Mean Absolute Error (MAE). The first 2/3 of the chronologically

Table 2: Overview of the evaluated event logs with their key properties.

Event Log	Customer Journey	Hospital Admission
Cases	15 001	46 520
Trace variants	1001	2784
Events	55 220	117 952
Events per case (mean)	3.681	2.536
Median case duration (days)	0.224	7.579
Mean case duration (days)	0.713	121.154
Activities	18	26
Words before preprocessing	247 010	171 938
Words after preprocessing	98 915	165 285
Vocabulary before preprocessing	1203	4973
Vocabulary after preprocessing	817	4633
Text attribute	Customer question	Diagnosis
Additional non-textual attributes	Gender	Admission type
	Age	Insurance

ordered traces is used to fit the prediction model to the historical event data. The remaining 1/3 of traces are used to measure the prediction performance.

The process prediction models are evaluated on two real-world event logs, of which the general characteristics are given in Table 2. Additionally, snippets of the datasets are shown in Tables 3 and 4. The first describes the customer journeys of the Employee Insurance Agency commissioned by the Dutch Ministry of Social Affairs and Employment. The log is aggregated from two anonymized data sets provided in the BPI Challenge 2016, containing click data of customers logged in the official website werk.nl and phone call data from their call center.

The second log is generated from the MIMIC-III (Medical Information Mart for Intensive Care) database and contains hospital admission and discharge events of patients in the Beth Israel Deaconess Medical Center between 2001 and 2012.

The results of the experiments are shown in Table 5. The next activity prediction shows an improvement of 2.83% and 4.09% on the two logs, respectively, showing that text can carry information on the next task in the process. While the impact of our method on next timestamp prediction is negligible in the customer journey log, it lowers the absolute error by approximately 11 hours in the hospital admission log. The improvement shown in the outcome prediction is small but present: 1.52% in the customer journey log and 2.11% in the hospital admission log. Finally, the improvement in cycle time prediction is particularly notable in the hospital admission log, where the error decreases by 27.63 hours. In general, compared to the baseline approaches, the text-aware model can improve the predictions on both event logs with at least one parametrization.

In addition, the prediction performance is evaluated per prefix length for each event log. Figure 4 shows the F_1 score and next timestamp MAE for every

Table 3: Snippet from the customer journey log.

Case	Activity	Timestamp	Age	Gender	Message
40154127	question	2015/12/15 12:24:42.000	50-65	M	Can you send me a copy of the decision?
40154127	taken	2015/12/30 15:39:36.000	50-65	M	
40154127	mijn_sollicitaties	2015/12/30 15:39:42.000	50-65	M	
40154127	taken	2015/12/30 15:39:46.000	50-65	M	
40154127	home	2015/12/30 15:39:51.000	50-65	M	
23245109	question	2015/07/21 09:49:32.000	50-65	M	Law: How is the GAA (Average Number of Labor)?
23245109	question	2015/07/21 09:54:28.000	50-65	M	Dismissal Procedure: Stops my contract automatically after two years of illness?
23245109	question	2015/07/21 10:05:43.000	50-65	M	Dismissal: Am I entitled to a transitional allowance?
23245109	question	2015/07/21 10:05:56.000	50-65	M	Chain Determination: How often may be extended a fixed-term contract?
23245109	mijn_werkmap	2015/07/27 09:54:03.000	50-65	M	
23245109	mijn_berichten	2015/07/27 09:54:13.000	50-65	M	
23245109	mijn_cv	2015/07/27 10:04:20.000	50-65	M	
21537056	taken	2015/10/30 13:16:48.000	50-65	M	
21537056	question	2015/10/30 13:22:00.000	50-65	M	How can I add a document/share with my consultant work through the workbook?
21537056	taken	2015/10/30 13:23:24.000	50-65	M	
21537056	mijn_werkmap	2015/10/30 13:24:39.000	50-65	M	
19290768	question	2015/09/21 12:41:21.000	30-39	V	Filling: What should I do if I made a mistake when filling out the Income Problem?
19290768	home	2015/09/22 10:09:53.000	30-39	V	
19290768	taken	2015/09/22 10:10:14.000	30-39	V	
19290768	home	2015/09/22 10:11:12.000	30-39	V	
53244594	mijn_berichten	2016/02/25 09:10:40.000	40-49	M	
53244594	question	2016/02/25 13:27:38.000	40-49	M	When is/are transferred my unemployment benefits?
53244594	question	2016/02/29 10:04:23.000	40-49	M	Problem: I have to pay sv € 0 and further fill only the amount of holiday pay. What should I do if I get an error?
53244594	question	2016/02/29 10:10:52.000	40-49	M	Why did you change the amount of my payment?

Table 4: Snippet from the hospital admission log.

Case	Activity	Timestamp	Admission Type	Insurance	Diagnosis
8	PHYS REFERRAL/NORMAL DELI	2117-11-20 10:22:00	NEWBORN	Private	NEWBORN
8	HOME	2117-11-24 14:20:00	NEWBORN	Private	
9	EMERGENCY ROOM ADMIT	2149-11-09 13:06:00	EMERGENCY	Medicaid	HEMORRHAGIC CVA
9	DEAD/EXPIRED	2149-11-14 10:15:00	EMERGENCY	Medicaid	
10	PHYS REFERRAL/NORMAL DELI	2103-06-28 11:36:00	NEWBORN	Medicaid	NEWBORN
10	SHORT TERM HOSPITAL	2103-07-06 12:10:00	NEWBORN	Medicaid	
11	EMERGENCY ROOM ADMIT	2178-04-16 06:18:00	EMERGENCY	Private	BRAIN MASS
11	HOME HEALTH CARE	2178-05-11 19:00:00	EMERGENCY	Private	
12	PHYS REFERRAL/NORMAL DELI	2104-08-07 10:15:00	ELECTIVE	Medicare	PANCREATIC CANCER SDA
12	DEAD/EXPIRED	2104-08-20 02:57:00	ELECTIVE	Medicare	
13	TRANSFER FROM HOSP/EXTRAM	2167-01-08 18:43:00	EMERGENCY	Medicaid	CORONARY ARTERY DISEASE
13	HOME HEALTH CARE	2167-01-15 15:15:00	EMERGENCY	Medicaid	
16	PHYS REFERRAL/NORMAL DELI	2178-02-03 06:35:00	NEWBORN	Private	NEWBORN
16	HOME	2178-02-05 10:51:00	NEWBORN	Private	
17	PHYS REFERRAL/NORMAL DELI	2134-12-27 07:15:00	ELECTIVE	Private	PATIENT FORAMEN OVALE PATENT FORAMEN OVALE MINIMALLY INVASIVE SDA
17	HOME HEALTH CARE	2134-12-31 16:05:00	ELECTIVE	Private	
17	EMERGENCY ROOM ADMIT	2135-05-09 14:11:00	EMERGENCY	Private	PERICARDIAL EFFUSION
17	HOME HEALTH CARE	2135-05-13 14:40:00	EMERGENCY	Private	
18	PHYS REFERRAL/NORMAL DELI	2167-10-02 11:18:00	EMERGENCY	Private	HYPOGLYCEMIA SEIZURES
18	HOME	2167-10-04 16:15:00	EMERGENCY	Private	
19	EMERGENCY ROOM ADMIT	2108-08-05 16:25:00	EMERGENCY	Medicare	C 2 FRACTURE
19	REHAB/DISTINCT PART HOSP	2108-08-11 11:29:00	EMERGENCY	Medicare	

Table 5: Experimental results for the next activity, next timestamp, outcome, and cycle time prediction. All MAE scores are in days.

Text Model	Text Vect. Size	BPIC2016 Customer Journey				MIMIC-III Hospital Admission			
		Activity F$_1$	Time MAE	Outcome F$_1$	Cycle MAE	Activity F$_1$	Time MAE	Outcome F$_1$	Cycle MAE
Text-Aware Process Prediction (LSTM + Text Model)									
BoW	50	0.4251	0.1764	0.4732	0.2357	0.5389	29.0819	0.6120	69.2953
BoW	100	0.4304	**0.1763**	0.4690	**0.2337**	0.5487	31.4378	**0.6187**	70.9488
BoW	500	**0.4312**	0.1798	0.4690	0.2354	**0.5596**	27.5495	0.6050	70.1084
BoNG	50	0.4270	0.1767	0.4789	0.2365	0.5309	27.5397	0.6099	69.4456
BoNG	100	0.4237	0.1770	0.4819	0.2373	0.5450	28.3293	0.6094	69.3619
BoNG	500	0.4272	0.1773	0.4692	0.2358	0.5503	27.9720	0.6052	70.6906
PV	10	0.4112	0.1812	0.4670	0.2424	0.5265	29.4610	0.6007	73.5219
PV	20	0.4134	0.1785	0.4732	0.2417	0.5239	**27.2902**	0.5962	69.6191
PV	100	0.4162	0.1789	0.4707	0.2416	0.5292	28.2369	0.6058	69.4793
LDA	10	0.4239	0.1786	0.4755	0.2394	0.5252	28.8553	0.6017	**69.1465**
LDA	20	0.4168	0.1767	0.4747	0.2375	0.5348	27.8830	0.6071	69.6269
LDA	100	0.4264	0.1777	**0.4825**	0.2374	0.5418	27.5084	0.6106	69.3189
LSTM Model Prediction Baseline									
LSTM [9]		0.4029	0.1781	0.4673	0.2455	0.5187	27.7571	0.5976	70.2978
Process Model Prediction Baseline (Annotated Transition System)									
Sequence [1, 13]		0.4005	0.2387	0.4669	0.2799	0.4657	64.0161	0.5479	171.5684
Bag [1, 13]		0.3634	0.2389	0.4394	0.2797	0.4681	64.6567	0.5451	173.7963
Set [1, 13]		0.3565	0.2389	0.4381	0.2796	0.4397	63.2042	0.5588	171.4487
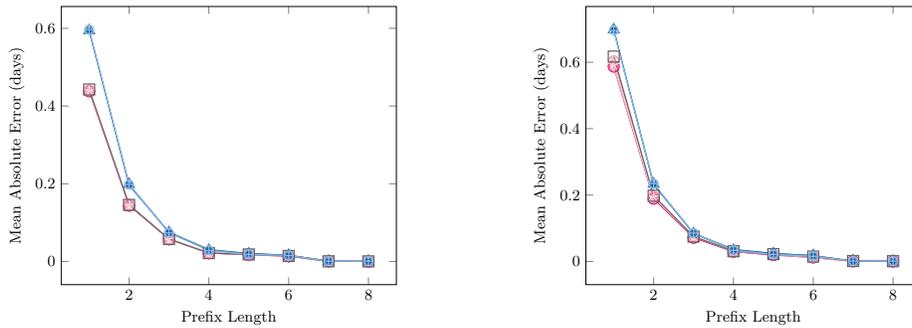
(a) BPIC2016 Customer Journey event log: next timestamp (left), cycle time (right).
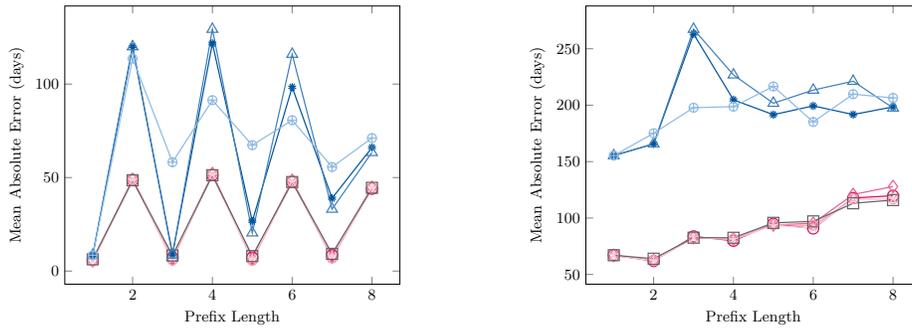
(b) MIMIC-III Hospital Admission event log: next timestamp (left), cycle time (right).

Fig. 4: Prediction performance on selected metrics and logs, shown by length of trace prefix.

prefix trace of length $1 \leq k \leq 8$ on a selection of prediction tasks. Note that the results on shorter traces are supported by a much larger set of traces due to prefix generation. For text-aware models, only the best encoding size is shown.

On the customer journey log, the performance of all models correlates positively with the available prefix length of the trace. All text-aware prediction models surpass the baseline approaches on very short prefix traces of length 3 or shorter, for next activity and outcome prediction: we hypothesize that the cause for this is a combination of higher availability of textual attributes in earlier events in the traces, and the high number of training samples of short lengths, which allow text models to generalize. The next timestamp and cycle time predictions show no difference between text-aware models and the LSTM baseline, although they systematically outperform transition system-based methods.

The hospital admission log is characterized by the alternation of admission and discharge events. Therefore, the prediction accuracy varies between odd and even prefix lengths. The text-aware prediction models generate slightly better predictions on admission events since only these contain the diagnosis as text attribute. Regarding the next timestamp prediction, higher errors after discharge events and low errors after admission events are observed. This can be explained by the short hospital stays compared to longer time between two hospitalizations.

6   Conclusion

The prediction of the future course of business processes is a major challenge in business process mining and process monitoring. When textual artifacts in a natural language like emails or documents hold critical information, purely control-flow-oriented approaches are limited in delivering accurate predictions.

To overcome these limitations, we propose a text-aware process predictive monitoring approach. Our model encodes process traces of historical process executions to sequences of meaningful event vectors using the control flow, timestamp, textual, and non-textual data attributes of the events. Given an encoded prefix log of historical process executions, an LSTM neural network is trained to predict the activity and timestamp of the next event, and the outcome and cycle time of a running process instance. The proposed concept of text-aware predictive monitoring has been implemented and evaluated on real-world event data. We show that our approach is able to outperform state-of-the-art methods using insights from textual data.

The intersection between the fields of natural language processing and process mining is a promising avenue of research. Besides validating our approach on more datasets, future research also includes the design of a model able to learn text-aware trace and event embeddings, and the adoption of privacy-preserving analysis techniques able to avoid the disclosure of sensitive information contained in textual attributes.

References

1. van der Aalst, W.M.P., Schonenberg, M.H., Song, M.: Time prediction based on process mining. Information Systems **36**(2), 450–475 (2011)
2. Abadi, M., Barham, P., Chen, J., et al.: Tensorflow: A system for large-scale machine learning. In: 12th USENIX symposium on operating systems design and implementation (OSDI 16). pp. 265–283 (2016)
3. Berti, A., van Zelst, S.J., van der Aalst, W.M.P.: Process Mining for Python (PM4Py): Bridging the Gap Between Process- and Data Science. In: International Conference on Process Mining (ICPM) Demo Track (CEUR 2374). pp. 13–16 (2019)
4. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation. the Journal of Machine Learning Research **3**, 993–1022 (2003)
5. Brown, P.F., Della Pietra, V.J., Desouza, P.V., Lai, J.C., Mercer, R.L.: Class-based n-gram models of natural language. Computational linguistics **18**(4), 467–480 (1992)
6. Ceci, M., Lanotte, P.F., Fumarola, F., Cavallo, D.P., Malerba, D.: Completion time and next activity prediction of processes using sequential pattern mining. In: International Conference on Discovery Science. pp. 49–61. Springer (2014)
7. Evermann, J., Rehse, J.R., Fettke, P.: A deep learning approach for predicting process behaviour at runtime. In: International Conference on Business Process Management (BPM). pp. 327–338. Springer (2016)
8. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International Conference on Machine Learning. pp. 1188–1196. PMLR (2014)
9. Navarin, N., Vincenzi, B., Polato, M., Sperduti, A.: LSTM networks for data-aware remaining time prediction of business process instances. In: 2017 IEEE Symposium Series on Computational Intelligence (SSCI). pp. 1–7. IEEE (2017)
10. Park, G., Song, M.: Predicting performances in business processes using deep neural networks. Decision Support Systems **129**, 113191 (2020)
11. Pedregosa, F., Varoquaux, G., Gramfort, A., et al.: Scikit-learn: Machine learning in Python. the Journal of Machine Learning Research **12**, 2825–2830 (2011)
12. Polato, M., Sperduti, A., Burattin, A., de Leoni, M.: Time and activity sequence prediction of business process instances. Computing **100**(9), 1005–1031 (2018)
13. Rogge-Solti, A., Weske, M.: Prediction of remaining service execution time using stochastic Petri nets with arbitrary firing delays. In: International Conference on Service-Oriented Computing. pp. 389–403. Springer (2013)
14. Tax, N., Teinemaa, I., van Zelst, S.J.: An interdisciplinary comparison of sequence modeling methods for next-element prediction. Software and Systems Modeling **19**(6), 1345–1365 (2020)
15. Tax, N., Verenich, I., La Rosa, M., Dumas, M.: Predictive business process monitoring with LSTM neural networks. In: International Conference on Advanced Information Systems Engineering (CAiSE). pp. 477–492. Springer (2017)
16. Teinemaa, I., Dumas, M., Maggi, F.M., Di Francescomarino, C.: Predictive business process monitoring with structured and unstructured data. In: International Conference on Business Process Management. pp. 401–417. Springer (2016)