On the Potential of Textual Data for Explainable Predictive Process Monitoring

Christian Warmuth[1,2] and Henrik Leopold[1,3]

[1] Hasso Plattner Institute, University of Potsdam, Germany
[2] SAP Signavio, Berlin, Germany
[3] Kühne Logistics University, Hamburg, Germany
christian.warmuth@sap.com, henrik.leopold@the-klu.org

Abstract. Predictive process monitoring techniques leverage machine learning (ML) to predict future characteristics of a case, such as the process outcome or the remaining run time. Available techniques employ various models and different types of input data to produce accurate predictions. However, from a practical perspective, explainability is another important requirement besides accuracy since predictive process monitoring techniques frequently support decision-making in critical domains. Techniques from the area of explainable artificial intelligence (XAI) aim to provide this capability and create transparency and interpretability for black-box ML models. While several explainable predictive process monitoring techniques exist, none of them leverages textual data. This is surprising since textual data can provide a rich context to a process that numerical features cannot capture. Recognizing this, we use this paper to investigate how the combination of textual and non-textual data can be used for explainable predictive process monitoring and analyze how the incorporation of textual data affects both the predictions and the explainability. Our experiments show that using textual data requires more computation time but can lead to a notable improvement in prediction quality with comparable results for explainability.

Keywords: Predictive Process Monitoring · Explainable Artificial Intelligence (XAI) · Natural Language Processing · Machine Learning

1 Introduction

In recent years, machine learning (ML) techniques have become a key enabler for automating data-driven decision-making [14]. Machine learning has also found its way into the broader context of business process management. Here, an important application is to predict the future of business process executions - commonly known as predictive business process monitoring [7]. For example, a machine learning model can be used to predict the process outcome [20], the next activity [9] or the remaining time of a running process [21].

From a practical point of view, one of the critical shortcomings of many existing predictive process monitoring techniques is that their results are not explainable, i.e., it remains unclear to the user how or why a certain prediction

was made [17]. Especially in critical domains, such as healthcare, explainability, therefore, has become a central concern. Techniques in the area of explainable artificial intelligence (XAI) aim to shed light on black box ML models and provide transparency and interpretability [1]. Recognizing this, several so-called explainable predictive process monitoring techniques have been proposed [10,18,14]. They rely on well-established explainability approaches such as SHAP [12] and LIME [16] to support users in better understanding the predictions of the employed techniques.

What existing explainable predictive process monitoring techniques have in common is that they solely rely on numerical and categorical attributes and do not leverage textual data. This is surprising given that textual data often provides rich context to a process. Recognizing the potential value of textual data for explainable predictive process monitoring, we use this paper to empirically explore how the combination of textual and non-textual data affects the prediction quality, the explainability analysis, and the computational effort. To this end, we propose two novel strategies to combine textual and non-textual data for explainable predictive process monitoring and conduct extensive experiments based on an artificial dataset.

The remainder of this paper is organized as follows: Section 2 illustrates the problem and the potential of using textual data for explainable predictive process monitoring. Section 3 elaborates on our study design. The code for all experiments can be found on GitHub[4]. Section 4 presents the results. Section 5 discusses related work before Section 6 concludes our paper.

2 Problem Illustration

Predictive process monitoring techniques aim to predict the future state of current process executions based on the activities performed so far and process executions in the past [7]. Given a trace, we might, for instance, aim to predict the outcome of a trace [20]. Depending on the context, such an outcome could relate to the successful completion of a production process or the successful curing of a patient. Predicting the outcome of a process execution at an early stage enables early interventions, such as allocating additional resources or taking a different course of action still to reach the desired process outcome [22].

A central problem in process monitoring techniques leveraging ML is that it is nearly impossible for humans to understand why a particular prediction was made. This led to the development of techniques for explainable artificial intelligence, which aim to produce more explainable models without deterioration of the predictive performance. The goal is to help humans comprehend, effectively use, and trust artificial intelligence systems [1]. One widely employed XAI strategy is to produce a simpler, understandable model that approximates the results of the original prediction model [12] such as SHAP [10,18] or LIME [14] which are commonly used in the context of predictive process monitoring.

[4] `https://github.com/christianwarmuth/explainable-predictive-process-monitoring-with-text`

All existing techniques for explainable predictive process monitoring have in common that they rely on numerical and categorical features only and do not consider textual data. This is surprising since textual data often can provide rich insights into the context of a process execution.

For example, consider a loan application process where customers may provide written statements about their financial situation, the purpose of the requested loan, and details of the repayment plan. This data might allow to more accurately predict whether the customer will pay back the loan and explain that prediction better. Figure 1 illustrates such a setting using an exemplary event log. We can see two cases where one applicant intends to spend the money on a wedding and the other on a new car. From the bank's perspective, this might make quite a difference since purchasing a car results in a physical asset that can be resold if the customer cannot pay it back.

case_id	activity	timestamp	loan amount	credit score	loan goal description
1566432	Create Application	15.03.2022 15:04	1.000$	0.93	I recently proposed to my wife so I ...
1566432	Review Application	17.03.2022 13:18	1.000$	0.93	/
1566432	Re-Negotiate Terms	17.03.2022 16:21	900$	0.93	/
1566432	Application Accepted	23.03.2022 09:15	900$	0.93	/
1748744	Create Application	16.03.2022 10:20	3.000$	0.87	I am planning to buy a new car and...
1748744	Review Application	17.03.2022 17:04	3.000$	0.87	/

Fig. 1: Exemplary Eventlog with textual context data

Recognizing the potential value of textual data in the context of explainable predictive process monitoring, we use this paper to investigate how the combination of textual and non-textual data can be used for explainable predictive business process monitoring and analyze how the incorporation of textual data affects both the prediction quality and the explainability.

3 Study on the Impact of Textual Data on Explainable Predictive Process Monitoring

In this section, we describe the design of our study to investigate the potential of textual data for explainable predictive process monitoring. In Section 3.1, we first explain the different strategies we use for combining textual and non-textual data and the models chosen for their instantiation. In Section 3.2, we introduce the dataset and its creation. In Section 3.3, we elaborate on the preprocessing and in Section 3.4 we explain the training and explanation setup for the experiments.

3.1 Strategies and Models

Combining textual and non-textual data for explainable predictive process monitoring is not trivial. That is because these different types of input data must

be combined in a useful way for both model building and inference and the explainability analysis. We propose two novel strategies:

Class Label or Probability Combination. Strategy one is to have two models (one for the textual data and one specific for the non-textual data). For inference, we can combine the class labels or the class probabilities output by the different models for prediction on real input. We have two separate explainability analyses as we have two individual models.

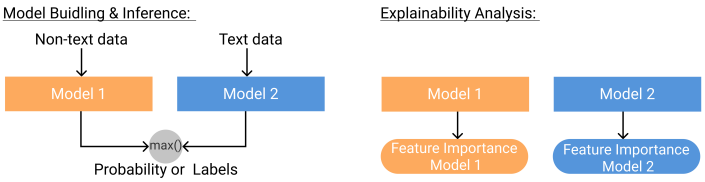

Fig. 2: Conceptual architecture Strategy 1

Two-stage model. In a two-stage model approach, we have one model using solely textual information as stage 1. We then filter out the n most important features (e.g., words or smaller parts of a sentence) and feed them into the stage 2 model alongside non-textual information. The explainability analysis would be performed on the second-stage model, considering both data sources.

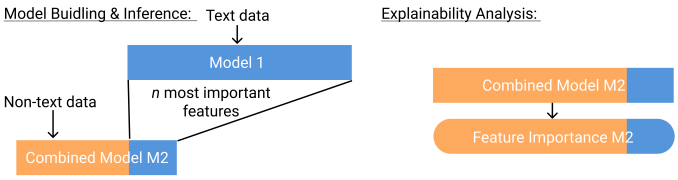

Fig. 3: Conceptual architecture Strategy 2

We needed to choose a model for each input type to instantiate these strategies. For *non-textual data*, i.e., categorical and numerical input, we selected the XGBoost model since it has been found to deliver the best average performance in predictive process monitoring across various datasets with good scalability for large datasets [20]. XGBoost uses gradient tree boosting, a common ensemble learning technique (i.e., combining multiple machine learning models to derive a prediction) which performs boosting on decision trees [4]. For *textual data*, we use BERT (Bidirectional Encoder Representations from Transformers), a state-of-the-art NLP model introduced by Devlin et al., which outperforms previous methods on various NLP tasks and datasets. BERT can be described as a large language model and belongs to the family of transformer models, the current state-of-the-art models dealing with sequences [6].

3.2 Dataset

There is no public event log dataset available that contains rich textual context data. We, therefore, artificially augment an existing event log with textual data. We chose to augment the BPIC17 dataset with textual context data on case level in a parameterizable fashion with the LendingClub dataset. The BPI Challenge dataset from 2017 refers to a credit application process filed by customers of a Dutch financial institution through an online system [8]. Overall, 12792 of the 31413 loans were granted, which leaves us with a 0.41 minority class ratio for this binary process outcome prediction problem on loan acceptance. The LendingClub dataset we use for dataset augmentation only includes textual descriptions of accepted loan applications, and we therefore have to redistribute the existing textual loan goal descriptions [11]. The redistribution is based on the topics discussed by the loan applicants in their loan goal description. In an initial data analysis, we identified the dominant topics using Latent Dirichlet Allocation, an NLP technique to retrieve topics in text corpora [2]. We assigned multiple topics to the two process outcomes and thus introduced in a controlled fashion, for example, that people who talk about medical issues in their loan goal description tend to be less likely to receive a loan offer. This approach creates a latent structure for the machine learning model to pick up in the prediction process. The topic attribution is performed based on the word occurrences per topic in the document. After determining the topic memberships, the dataset is augmented with the schematic depicted in Figure 4 with a varying parameter of impurity, which adjusts the proportion of randomly assigned texts samples from the dataset during the data augmentation process. The loan goal descriptions are added to the original BPIC17 event log as an additional feature in the first event for each case (the filing of the loan application).

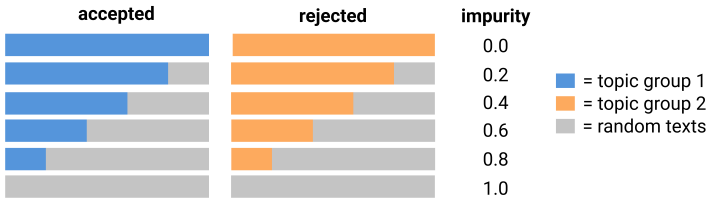

Fig. 4: Dataset augmentation strategy with impurity parameter

With an impurity of zero, the accepted cases are solely assigned the textual descriptions talking about topics in topic group 1. As the newly introduced textual features do not correlate with existing features, we thus introduce an additional dimension to differentiate between accepted and rejected cases. An impurity of 0.0 allows for an apparent differentiation in textual features. In contrast, an impurity of 1.0 would be a baseline with purely randomly sampled text for both outcomes, so there is no way to differentiate between the outcomes on the textual data. We henceforward define $purity = 1 - impurity$. For all

experiments described in the following, we create 11 synthetically augmented dataset variants with an impurity ranging from 0.0 to 1.0 in steps of 0.1. We reduce measurement deviations by running each experiment 10 times and taking the arithmetic mean.

3.3 Data Preprocessing

We conduct several preprocessing steps. First, we need to retrieve the class labels "accepted" and "rejected" by choosing respective end activities. Then, we need to transform the input such as it is suitable for the employed models. For the XGBoost model, we have multiple events per case with various attributes that change during the process executions. However, the XGBoost model expects static (non-sequential input). We, therefore, preprocess the data to derive static properties (i.e., one n-dimensional vector of features per case) and convert all activities performed into categorical variables (encoding whether they occurred or not). All further categorical variables are one-hot-encoded (resulting in one additional feature per category level) to represent categorical variables using numerical values. Numerical variables are then standardized by removing the mean and scaling them to unit variance. Since we use BERT models for the textual data, we do not need extensive preprocessing steps. The model can process the textual data without significant assumptions and in considerable length. We, however, need to tokenize the dataset before feeding it into the BERT model with the model-specific tokenizer (in our case "BERT base model (uncased)").

3.4 Model Training and Explanation

For strategy 1, we focus on combining the class attribution probability of an XGBoost Model and a BERT model, which is fine-tuned on our dataset. We then decide per case which of the models' predictions results in a more significant absolute difference to the probability of 0.5 and, therefore, provide a clearer decision. Both models are fed into the SHAP explainer module and are individually explained. The SHAP framework is generally model-agnostic, but model-specific optimizations for faster calculation exist. The SHAP framework relies for BERT on the so-called PartitionExplainer and for XGBoost on TreeExplainer.

For strategy 2, we first use the identical BERT setup described above. However, we then perform an explainability analysis using the SHAP framework to filter out the n most important words. We then feed these n features into an XGBoost model as the second stage to derive the final prediction. As mentioned above, BERT will be explained using the SHAP PartitionExplainer. As we use XGBoost in the second stage, we delete the stopwords before feeding these features into the XGBoost model. XGBoost disregards a word's left and right context and its sequential nature. The n most important features of the BERT explainability analysis after stopwords removal are represented using the well-known TF-IDF approach before using the XGBoost model. For the explainability analysis of strategy 2, we only consider the second-stage XGBoost model.

4 Results

Effect on Model Performance. The two strategies and their performance on the different augmented datasets are assessed using an F1-score and ROC AUC, which are common evaluation metrics for classification problems. We also introduce another baseline with "baseline unilateral" predicting all inputs with the majority class. Overall, we differentiate between strategies 1 and 2 on the augmented dataset and a baseline model on non-textual data only. The results in Figure 5 show that already for purity of above 0.1, the proposed strategies lead to a net improvement of both ROC AUC and F1-score. The results suggest that the strategies provide a benefit even at low levels of textual data purity and improve the model performance. The combined incorporation of textual and non-textual information shows value in light of a low level of textual data purity as neither model alone can score these results. Using a pure textual model also creates similar results for high textual data purity (around 1.0), as shown by the pure BERT performance. Therefore, we can conclude that both strategies are valuable in that they provide higher predictive quality, especially for low levels of textual data purity, while the performance of the models converges for a very high purity on textual features. There is a slight difference discernible between strategies 1 and 2.

Effect on Rediscovery Rate. We calculate a metric of rediscovery to determine whether the artificial latent structures introduced during the dataset augmentation are uncovered and manifested in the explainability analysis. The rediscovery rate will be measured by the overlap between the most important textual features derived by the SHAP calculations and the input features used during the dataset augmentation via word2vec vector similarity. Word2vec represents words in a high-dimensional vector space [13]. We used the pre-trained word2vec vectors based on the Google News dataset[5]. In our rediscovery calculation, we consider two words as rediscovered if the cosine similarity between the two words on the pre-trained word2vec vectors is above 0.3 and if the mean absolute feature importance via SHAP is above 0.005. Since both strategies show high rediscovery rates, one can conclude that the right latent structures seem to be found, and the strategies seem to work as intended. There is a difference between strategies 1 and 2, which indicates that strategy 1 rediscovers more of the latent features introduced during dataset augmentation. Strategy 2 incorporates a limited amount of features and thus leads to a lower yet still considerable rediscovery rate.

Effect on Quantitative Explainability Metrics. Stevens et al. propose an approach to quantitatively evaluate the explainability of ML models, particularly for the process domain. Their approach distinguishes interpretability (measured by parsimony), as well as faithfulness (measured by monotonicity) [18].

Parsimony. Parsimony as a property can describe the explainability models' complexity. Parsimony describes the number of features in the final model and

[5] `https://code.google.com/archive/p/word2vec/`

can quantify the simplicity of a model. For post-hoc explainability analysis using feature importance, the non-zero feature weights are considered. The maximal value of the parsimony property is the number of features. A simple (or parsimonious) model is characterized by a small parsimony value [18]. To compare the parsimony, we take the parsimony for the baseline model, for strategy 1 (as a sum of both models' feature counts), and the second-stage model of strategy 2. We can see a significant difference between the baseline model and strategy 1 in Figure 5. For strategy 2, the parsimony is only slightly higher than the baseline and converges against an upper boundary since we limit the number of textual features n in the second-stage model.

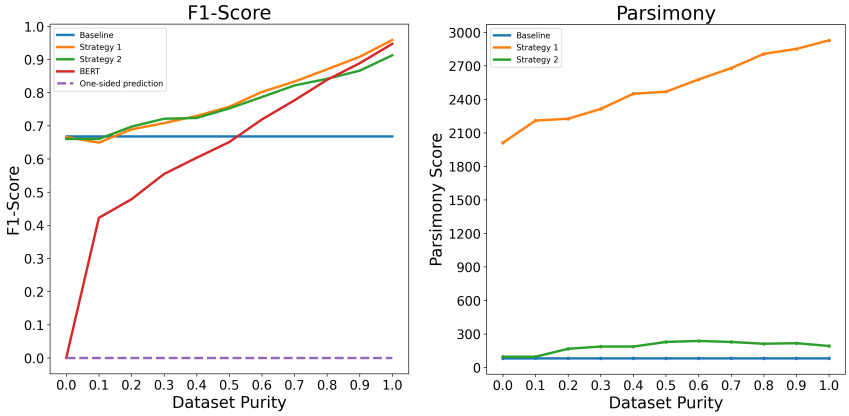

Fig. 5: F1-score and parsimony for augmented datasets with varying impurity

This implies that strategies 1 and 2 naturally consider substantially more features than the baseline. For strategy 1, even more features are incorporated in an explainability analysis with a higher purity of the augmented datasets and overall better model performance. As parsimony is a metric to determine how interpretable an explainability analysis is, this consequently means that models considering textual information (strategy 1 and strategy 2) are more challenging to interpret. We have to note here that the parsimony of strategy 2 is significantly below the parsimony of strategy 1. Therefore, the interpretability of strategy 2 is better as we limit the number of features to incorporate by the parameter n. In their elaboration on feature importance techniques specifically in the area of NLP, Danilevsky et al. argue in their work that "[t]ext-based features are inherently more interpretable by humans [...]" [5]. Following this line of reasoning, it is not entirely correct to assign non-textual and textual features the same negative impact on interpretability, which puts the results into relative terms.

Monotonicity. Monotonicity can be used as a metric to describe the faithfulness between the model and the explanation. Monotonicity describes the faithfulness

between the feature importance resulting from the explainability analysis and the feature importance of the task model. For models that require post-hoc explainability, the monotonicity is denoted by the Spearman's correlation coefficient between the absolute values of the feature weights for the task model and the absolute values of the feature weights of the explainability model [18]. The range of the monotonicity lies between $[-1, 1]$ and describes the association of rank, where a perfectly faithful model would have a Monotonicity M of $+1$. In contrast, a less faithful model would score values closer to 0. A negative Spearman correlation coefficient implies a negative association of rank between the task model's feature importance and the explainability model's feature importance. For strategy 2 in the second stage and the baseline model, we use XGBoost as a model of choice, which provides inherent task model-specific feature importance. While there are multiple ways to assess XGBoost-specific feature importance, we will focus on the importance by the number of times a feature is used to split the data across all trees of the decision tree approach. We will not consider the monotonicity metric for strategy 1 because it is a BERT model for which task model-specific feature importance cannot be directly obtained.

We see that the monotonicity of the baseline model and the second-stage model in strategy 2 are almost similar. While there is only a small difference in monotonicity initially, it disappears with higher dataset purity. The results on monotonicity showed little to no difference between strategy 2 and the baseline. This indicates no notable difference in the faithfulness of the explainability analysis in comparison with the original prediction model. As elaborated before, we cannot calculate the monotonicity score for strategy 1 due to a lack of task model feature importance from the BERT model. Therefore, the statement relates to strategy 2 only.

Effect on Computation Time. For strategy 1, we add up both models' training time and the explanation time. For strategy 2, we add the training time of both stages together for training. At the same time, we only consider the explanation time of the second stage as we only perform an explanation computation via SHAP for this second stage.

The results show a significant difference between the baseline and strategies 1 and 2 for model training and explainability calculation. For the baseline, the training is performed quicker than the explanation, while this holds not true for strategies 1 and 2. The training and explanation of strategy 2 take only marginally longer than for strategy 1 but are considerably more expensive than for the baseline. There is also a noteworthy difference between training time and time for the SHAP calculations. The evaluations showed that the training times and explainability analyses required significantly more time for the proposed strategies than for the baseline. Our experiments suggest that for a high number of features and complex models, the computation for the explainability analysis far outweighs the training time. We can, however, not draw a conclusion regarding the ratio of training and explainability times, as this is highly dependent on the model choice and the dataset used for evaluation.

Prototype. To contemplate the practical implications of using textual data for explainable predictive monitoring of business processes, we developed a prototype illustrating how this might affect users. We differentiate between local explainability (for individual process instances) and global explainability (overview over all process instances). This screenshot shows a local analysis of strategy 1 divided into two separate models for the prediction as well as the explanation. A red color in the individual explainability plots indicates a positive change (towards a loan acceptance); blue color indicates a negative change in the expected model prediction (towards a loan rejection).

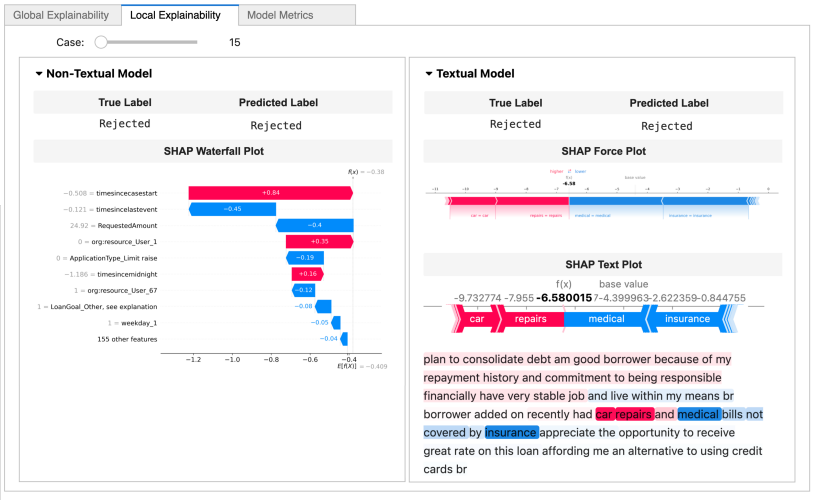

Fig. 6: Prototypical implementation of local explainability analysis (Strategy 1)

5 Related Work

Predictive process monitoring techniques have been developed for a wide range of purposes. The most prominent use cases include the prediction of the process outcome [19,22] and the prediction of future process behavior, such as the next activity [9]. While most techniques build on categorical and numerical features to accomplish their prediction goal, some also take into account textual data. For instance, Pegoraro et al. use different strategies such as TF-IDF, Doc2Vec, or LDA to represent textual information and, in this way, integrate it into an LSTM architecture with further categorical and numerical data [15]. Teinemaa et al. perform predictive monitoring with structured and unstructured data by concatenating the textual features to the feature vector of the non-textual features. The text is represented, among others, using bag-of-n-grams, TF-IDF, and LDA [19]. A recent technique from Cabrera et al. [3] uses contextualized word embeddings to predict the next activity and the next timestamp of running cases.

Recognizing the need for explainability, several so-called explainable predictive process monitoring techniques have been developed. These techniques

mostly rely on model-agnostic approaches such as SHAP [10,18] or LIME [14]. SHAP unifies existing model explanation techniques (which include six existing methods, amongst others, LIME [16]). SHAP is a unified measure to calculate post-hoc feature importance by using the Shapley values of the conditional expectation function of the original model [12]. All explainable predictive process monitoring techniques have in common that they rely on numerical and categorical features only and do not consider textual data. Hence, this paper empirically demonstrates the potential of explainable predictive process monitoring based on textual and non-textual data.

6 Conclusion and Future Work

This paper empirically explored the potential of combining textual and non-textual data in the context of explainable predictive process monitoring. To this end, we conducted extensive experiments on a synthetic dataset we created for this purpose. We found that using textual data alongside non-textual data requires more computation time but can lead to better predictions even when the quality of the textual data is poor. While the explainability metrics might decrease slightly depending on the chosen strategy, textual information is inherently more interpretable by humans, which allows for a more human-understandable explanation. Therefore, we conclude that combining textual and non-textual data in the context of explainable predictive process monitoring is a promising approach.

As for future work, we see two main directions. First, after an explainability analysis, it is unclear whether a variable is merely correlated with the outcome or causally related. Therefore, future work should combine the explainability analysis with a subsequent causality analysis. Second, it would be interesting to relate the results of an explainability analysis to real interventions.

References

1. Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., Herrera, F.: Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. Information Fusion **58**, 82–115 (2020)
2. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
3. Cabrera, L., Weinzierl, S., Zilker, S., Matzner, M.: Text-aware predictive process monitoring with contextualized word embeddings
4. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 785–794. KDD '16 (2016)
5. Danilevsky, M., Qian, K., Aharonov, R., Katsis, Y., Kawas, B., Sen, P.: A survey of the state of explainable AI for natural language processing. In: Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing. pp. 447–459. AACL (2020)

6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1. pp. 4171–4186 (2019)
7. Di Francescomarino, C., Ghidini, C., Maggi, F.M., Milani, F.: Predictive process monitoring methods: Which one suits me best? In: Business Process Management. pp. 462–479 (2018)
8. van Dongen, B.: Bpi challenge 2017 (2017), `https://data.4tu.nl/articles/dataset/BPI_Challenge_2017/12696884/1`
9. Evermann, J., Rehse, J.R., Fettke, P.: A deep learning approach for predicting process behaviour at runtime. In: Business Process Management Workshops. pp. 327–338 (2017)
10. Galanti, R., Coma-Puig, B., Leoni, M.d., Carmona, J., Navarin, N.: Explainable predictive process monitoring. In: 2020 2nd International Conference on Process Mining (ICPM). pp. 1–8 (2020)
11. George, N.: Lending club loan application data (2017), retrieved in December 2021 from `https://www.kaggle.com/wordsforthewise/lending-club`
12. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Advances in Neural Information Processing Systems 30, pp. 4765–4774 (2017)
13. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. Proceedings of Workshop at ICLR **2013** (2013)
14. Ouyang, C., Sindhgatta, R., Moreira, C.: Explainable AI enabled inspection of business process prediction models. CoRR **abs/2107.09767** (2021)
15. Pegoraro, M., Uysal, M.S., Georgi, D.B., van der Aalst, W.M.: Text-aware predictive monitoring of business processes. Business Information Systems **1**, 221–232 (2021)
16. Ribeiro, M.T., Singh, S., Guestrin, C.: "Why should I trust you?": Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016. pp. 1135–1144 (2016)
17. Rizzi, W., Di Francescomarino, C., Maggi, F.M.: Explainability in predictive process monitoring: When understanding helps improving. In: Business Process Management Forum. BPM 2020. Lecture Notes in Business Information Processing. pp. 141–158
18. Stevens, A., De Smedt, J., Peeperkorn, J.: Quantifying explainability in outcome-oriented predictive process monitoring. In: Process Mining Workshops. pp. 194–206. Springer International Publishing (2022)
19. Teinemaa, I., Dumas, M., Maggi, F.M., Di Francescomarino, C.: Predictive business process monitoring with structured and unstructured data. In: Business Process Management. pp. 401–417 (2016)
20. Teinemaa, I., Dumas, M., Rosa, M.L., Maggi, F.M.: Outcome-oriented predictive process monitoring: Review and benchmark. ACM Transactions on Knowledge Discovery from Data (TKDD) **13**(2), 1–57 (2019)
21. Verenich, I., Dumas, M., La Rosa, M., Maggi, F., Teinemaa, I.: Survey and cross-benchmark comparison of remaining time prediction methods in business process monitoring. ACM Transactions on Intelligent Systems and Technology **10**, 1–34 (2019)
22. Weytjens, H., De Weerdt, J.: Process outcome prediction: Cnn vs. lstm (with attention). In: International Conference on Business Process Management. pp. 321–333. Springer (2020)