2021 8th Intl. Conference on Soft Computing & Machine Intelligence

Attention-Based Anomaly Detection in Hospital Process Event Data

2021 8th International Conference on Soft Computing & Machine Intelligence (ISCMI) | 978-1-7281-8683-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/ISCMI53840.2021.9654821

Philippe Krajsic
Business Information Systems Institute
Leipzig University
Leipzig, Germany
krajsic@wifa.uni-leipzig.de

Bogdan Franczyk
Business Informatics Institute
Wroclaw University of Economics and Business
Wroclaw, Poland
bogdan.franczyk@ue.wroc.pl

***Abstract*—The increasing complexity of processes makes it difficult to monitor them efficiently. This also includes the care paths in hospitals, whose process reliability is highly relevant, since unnoticed process deviations can have major consequences. For this purpose, a classification model based on self-attention is presented, which derives process logics from the underlying event data and allows to detect event-based anomalies in process flows without prior process knowledge. With an F1 score of 0.93, the model achieves a very good detection accuracy compared to other approaches which achieve F1 scores of 0.65 - 0.75 for classification tasks in similar domains. This can support process mining activities and makes the monitoring of hospital processes more reliable.**

***Keywords*—anomaly detection, attention, classification, event data, hospital process**

I. Introduction

Increasing complexity of processes due to a multitude of actors, the integration of information systems into the process flows and a related generation of ever increasing amounts of data leads to new challenges in the management of processes and analysis of the generated data. In order to be able to adequately analyze the data generated from the process flows, new analysis tools, like Process Mining [1], have been developed in recent years, which enable a model-based as well as data-centered analysis of these data and thus enable process flows to be recognized and optimized.

In the context of process mining, so-called event logs, in which the individual process activities are logged in chronological order, are extracted from the information systems and analyzed. So far, it has often been assumed that this event data is free of anomalies and incorrect behavior, but this often does not correspond to reality [2-4]. Incorrect data in the event logs can lead to incorrect results during further processing. For example, the accuracy of drift detection can be negatively affected by stochastic vibrations due to inaccurate event data [5, 6]. Approaches that have been conducted in this area using filtering techniques to eliminate erroneous events from event data show an improvement in the quality of process mining techniques which leads to an optimization of the analysis of the processes [7-9].

The processes within a hospital include different types of processes, such as supply chain processes to ensure the availability of medical equipment, but also health care processes that play an important role in the safety and health of patients. Such "care pathways" are the standard pathways of care embodied in clinical guidelines, clinical practice, and evidence-based care. Such care paths can also be documented by special hospital information systems (HIS), whose data can be evaluated. Since errors in the process flows can have serious consequences for patient care, the aim is to detect errors in the process flow as soon as they occur in order to be able to intervene in the process flow in time if necessary.

In this paper, a deep learning based classification model is presented that uses recent technologies, such as an attention mechanism and deep generative models, to analyze and efficiently classify time series such as event data. Anomalies in the care processes can thus be identified and detected in order to minimize subsequent impacts. As an upstream step in a process mining workflow, the classification of this event data enables the increase of data quality for downstream process mining activities. High-quality process mining results can thus lead to an optimization of processes in different areas of healthcare.

The remainder of this paper is structured as follows: Section II gives an general overview of process mining in health care, a definition of event data and the type of process anomaly detection that is talked about within this work. Section III presents related work in the area of business process anomaly detection in general and within hospital and health care processes in particular. Section IV presents the classification model. In Section V a technical experiment is conducted where the experimental setting as well result of the experiment are described. Section VI concludes this paper and presents future.

II. Process Mining in Health Care

A. Process Mining

Process Mining forms the link between traditional model-based process analysis (e.g. process automation or business process improvement) and data-centered analysis techniques (e.g. statistics and predictive analytics). In addition, it offers new ways to discover, monitor and improve real processes by extracting knowledge from information systems and is applicable in a wide variety of domains. Three types of process mining can be used for analysis purposes: Process Discovery, Conformance Checking and Enhancement [1].

This work was supported by the German Federal Ministry of Education and Research (BMBF, 01IS18026B) by funding the competence center for Big Data and AI "ScaDS.AI Dresden/Leipzig".

B. *Business Process Event Data*

An event is any data that is recorded during the execution of a process and is considered the smallest unit within a process. The granularity of an event depends on the application domain as well as the way it is recorded. For example, an event can describe which activity of a process was executed at what time. In the same way, it can also describe the different stages of the execution of an activity, e.g. events refer to the scheduling, starting, suspending, continuing or completing of an activity [10]. An event is an assignment of values to a set of attributes. Table I shows an example event log.

TABLE I. EXAMPLE EVENT LOG

Case-ID	Event-ID	Activity	Resource	Timestamp
1	1001	Register	Martin	06-08-2019:15.02
1	1002	Triage	Max	06-08-2019:16.52
1	1003	Blood test	Jonas	12-08-2019:09.05
1	1004	Register	Martin	18-08-2019:11.36
2	2001	X-ray	Max	20-08-2019:13.26
2	2002	Blood Test	Anne	22-08-2019:11.20
2	2003	X-ray	Max	26-08-2019:15.57
	...			

C. *Health Care Process*

Health care processes are characterized by the fact that various organizational units may be involved in the treatment of patients. These organizational units generally have their own specific IT applications (i.e. Hospital Information Systems HIS), although it is clear that extracting data related to healthcare processes presents challenges. Nevertheless, the systems used in hospitals must provide an integrated view of all these IT applications, as it must be guaranteed that, for instance, treatment and billing processes can be carried out properly. These systems therefore contain process-related information about health care processes and can therefore be used for process mining activities and the analysis of these information. Fig. 1 shows a general impression of the application of process mining in the health care sector. Every activity carried out in a hospital by a doctor, nurse, technician or any other resource for the care of patients is normally stored in a HIS (network of databases, systems, logs, events, etc.). The activities are recorded in so-called event logs for support, control and further analysis activities. Process models are created to specify the order in which health professionals act in a particular process or for critical analysis of the process design. These process models can in turn be used to support the development of HIS, for example to understand how the information system can support process execution.

D. *Process Anomaly Detection*

Based on the extracted event data from the HIS, analyses can be performed. The analysis of the event data in this work includes a classification and differentiation of the event data into error-free and erroneous event data, which can be used for subsequent process mining activities. Thus, for example, process conformance checking can be supported in the detection of process deviations or it enables the detection of errors in the event data if no process conformance checking can be performed due to missing target process models.

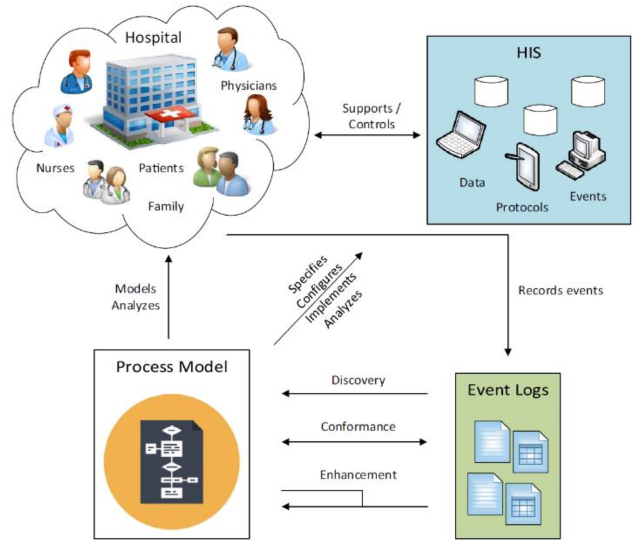

Fig. 1. Process mining in health care adapted from [1].

III. RELATED WORK

Regarding the approaches to filtering methods of event data in the context of process mining, the current state of research can be described as follows. The related work done in the areas of online process mining and anomaly detection is of particular importance for the work presented in this paper. With regard to detection and filtering of anomalies in event logs, there are some approaches described in the literature. In [8] a reference model is used to detect inappropriate behavior and repair the affected log. The approach proposed in [9] is based on an automaton which is modeled on the frequent process behavior recorded in the logs. Events that cannot be reproduced by the automaton are removed. In [10] an approach that uses conditional probabilities between activity sequences to eliminate events that are unlikely in a particular sequence is proposed. While existing techniques for filtering anomalies from event-logs show that they help improve process quality, they cannot be applied to the filtering of event streams in an online context. A real time approach for filtering anomalies in event streams is offered in [11]. It proposes an event processor that allows to effectively filter out unwanted events from an online event stream, based on probabilistic automaton. The basic idea of the approach is that dominant behavior achieves a higher probability of occurrence in the automaton than unlikely behavior. Thus, this filtering technique improves process discovery in real-time process mining. In [12], an architectural approach is presented that enables filtering of faulty process behaviors at the event level. For this purpose, an event filter is embedded in the streaming layer of a lambda architecture. Furthermore, an approach in [13] is presented that uses a hybrid approach to identify anomalies in the latent representations of the event data.

Regarding the application of such techniques with respect to hospital event data, some few work exists. In [14] an outlier behavior repair approach is applied to a hospital billing event log. In [15] a probabilistic topic model is use to identify anomalies in clinical pathways.

These few works in this area leave room for the development of further concepts and technologies regarding anomaly detection in hospital event logs.

IV. Methodology

This section presents the methodology for developing and applying a classification model to identify anomalies in hospital event logs.

A. Data Preparation

Before the event data can be used for training and testing a classification model, they must first be made suitable for processing by machine learning algorithms. Those event data preprocessing steps include:

- Measurement of data set statistics: includes the use of statistics to identify gross outliers in the values
- Imputation of missing values: involves replacing missing or inconsistent data values in the data set
- Trace encoding via node2vec: coding the traces as efficiently as possible leads to better performing classification results. For this purpose, the event data is subjected to a node2vec encoding [16].
- Normalization of the data set: A normalization of the data set is used to unify the scaling of numerical data values with differing value ranges.
- Based on this performed processing and transformation of the data set, a pre-processed data set is obtained, which can be passed to a classification model for training and testing purposes.

B. Classification Model

The classification model receives a sequence of event data $x = (x_1, x_2, x_3, \dots, x_T)$ as input. This input sequence is passed to the encoder. The encoder of the model is based on a bidirectional lstm as an expression of a recurrent neural net. The bidirectional lstm encoder is used to generate sequences of hidden states both forward and backward based on the input sequence. The hidden states of the network form a component that allows the network to store past and already known information and thus can be used for tasks that require verification of long-term dependencies, such as the identification of anomalies in business process data considered here. By using a bidirectional lstm, it is possible to view the input sequences from both sides and thus identify additional structural properties of the input sequences. The hidden states h_T^e generated by the encoder are then combined into a vector and passed to the network's variational layer for further processing.

Within the variational layer of the network the statistical variables μ_z (mean) and σ_z (variance) are generated based on the hidden states h_T^e of the encoder, which represent a normal distribution. From a normal distribution defined this way, a point z is sampled, which is a latent representation of this normal distribution. The classification model will be able to learn this representation of the observed event data und to learn the underlying structure of the event data.

The principle of the self-attention mechanism [17] is based on passing contextual information from the encoder's hidden states to the decoder. The self-attention layer receives a sequence of hidden state vectors as input and gives a sequence of context vectors c_t, which are determined as weighted sum of all input vectors. For this, the relevance of the pairwise hidden states of the encoder h_i^e and h_j^e of dimension d_{h^e} is determined using the scaled dot-product [17]. The higher the scaled dot-product of two hidden states, the more similarities two events have. In this way, the relationship between the events can be determined. A softmax function is applied to this determined score, which is used to normalize the generated weightings:

$$score(h_i^e, h_j^e) = softmax(\frac{(h_i^e)^T h_j^e}{\sqrt{d_{h^e}}}) \tag{1}$$

From the thus weighted hidden states of the encoder, the individual context vectors c_t are determined, which are passed to the decoder for further processing. At each decoding step, the decoder is informed on the basis of the context information from the context vector c_t which information in the hidden states h_T^d has to be considered separately.

Together with the latent representation z, the context vectors c_t are passed to the bidirectional lstm decoder for each timestep t and similar to the case of the encoder, hidden states h_T^d are generated. The hidden states h_T^d, determined by means of the bidirectional lstm decoder, are used to generate the output of the network. The output are the reconstructed sequences x_t', of the input sequences x_t considered in the input layer. To obtain the reconstructed sequences x_t' from the hidden states h_T^d of the decoder, the hidden states h_T^d are connected by two fully connected layers.

C. Post-Processing of Classificaion Results

A reconstruction-based approach for anomaly detection in event data is defined as decompression or reconstruction of the previously compressed input data with as little loss of information as possible. Possible losses that occur during the reconstruction of the compressed input data are referred to as reconstruction errors.

The classification model in this work is trained with normal event sequences to learn the normal behavior of the data and the underlying patterns therein. It is assumed that normal event data can be reconstructed better than event data containing anomalies, since these are unknown to the model from the training phase.

Based on the reconstruction of the distribution parameters coming from the variational layer, a reconstruction error ε can be determined from the distance between the input data point and the expected value that this data point comes from the learned distribution

$$\varepsilon = \|x - E[p_\theta(x_i|z_i)]\| \tag{2}$$

The reconstruction error ε is used as an anomaly score to calculate a threshold beyond which a data point is flagged as an anomaly. The threshold to be determined is denoted as τ and is determined as the average reconstruction error over all data points considered.

$$\tau = \frac{\sum_{i=1}^{n} \| x - E[p_\theta(x_i|z_i)] \|}{n} \quad (3)$$

Data points with an above-average reconstruction error are thus classified as an anomaly.

V. Technical Experiment

In order to verify the functionality and quality of the presented approach, a technical experiment is carried out. First, the data set used (hospital event log) is described in more detail. Subsequently, the setup of the experiment is explained. Finally, the results of the experiment are explained and discussed.

A. Hospital Event Log

The event data used for this technical experiment is taken from a Dutch Academic Hospital [18]. This hospital event log contains about 150,000 events in about 1,100 cases. There each case recorded in the event log represents a patient of a gynaecology department. The log contains information about when certain activities took place and which group performed the activity. Some attributes are repeated more than once for a patient, indicating that this patient went through different (maybe overlapping) phases, where a phase consists of the combination diagnosis and treatment.

B. Experimental Setup

A prototype implementation of the classification model was developed for the experiment [19]. For these purposes, a Google Colab environment is used, which provides access to GPU and TPU resources, such as Nvidia K80s, T4s, P4s, or P110s, for training the model.

In order to evaluate the performance of the classification model, the available data sets are divided into 70% training, 20% testing and 10% validation data. Since the model is only learned with normal data over a training dataset, anomalies must be added to the test and validation datasets to verify that the model is working. These anomalies include reworks and inserts:

- *Reworks:* A sequence of up to 3 events has been executed twice.
- *Inserts:* Up to 3 random activities were inserted into the case.

In 10% of all cases in the test and validation datasets, anomalies were inserted this way. Hyperparameter optimization was performed for the classification model. For this purpose, a grid search was performed. The following parameters were chosen for training of the model: batch size of 64 over 50 epochs with a dropout rate of 0.5 and a learning rate of 0.006, adam as an optimizer, softmax and sigmoid activation.

C. Experimental Results

In the following the results of the technical experiment are presented. Fig. 2 shows the confusion matrix of the classification results obtained. The confusion matrix provides an overview of the correct assignment of the events to their respective classes (normal or anomaly). We can see that 19674 events represent normal events and could be correctly classified as such (true negative), whereas 74 events were incorrectly classified as normal although they represent an anomaly (false negative). During the classification process, 2311 erroneous events were correctly identified as such (true positives). 1941 events were incorrectly classified as anomalies and actually represent a normal event (false positive).

Based on this information different evaluation metrics can be calculated. Within this work we calculate the metrics precision, recall and F1-Score. For the Precision metric we get a value of 0.95 and for the Recall a value of 0.92. From this we get a F1 score of 0.93. This allows us to infer that the classification model performs well on unbalanced (lots of normal data, few anomalies) classification problems, such as the anomaly detection in event data performed here.

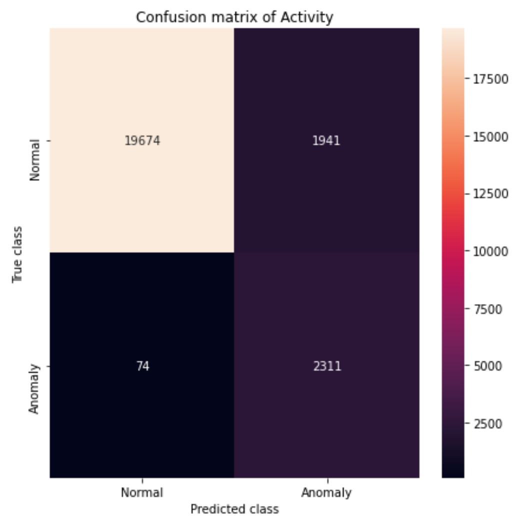

Fig. 2. Confusion matrix of the classification results.

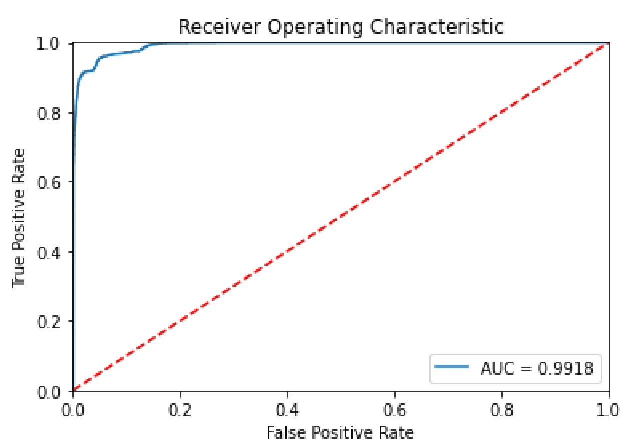

Fig. 3. ROC curve of the classification results.

In Fig. 3 we can see the receiver operating characteristics curve (ROC) as well as the associated area under the curve (AUC). The higher AUC turns out to be, the better the assignment of the event data to their respective classes is. With

AUC = 0.99, the classification model is very good at distinguishing between normal and erroneous event data and thus clearly exceeds a purely random classification (red line). The integrated self-attention layer significantly increases the classification accuracy. In a comparison, the experiment was carried out again without an additional self-attention layer, whereby only an F1 score of 0.71 could be achieved. The integration of such a self-attention mechanism thus has a positive effect on the quality of the classification model.

VI. Conclusion

In this work, a machine learning based classification model was presented that was applied to a hospital event log to identify synthetically inserted anomalies from the hospital process. Due to a high accuracy in the assignment of event data to their respective classes and thus also a precise identification of anomalies in the event data, incorrect behavior in the supply processes of a hospital can be identified and reacted to. The classification model makes it possible to derive process logics and structures from the underlying event data and to learn the rules on which these logics are based. This allows for cross-domain application of this technology and no assumed process knowledge, enabling decision support even for non-process experts. With respect to process mining, the classification model supports process mining activities such as conformance checking, which assumes predefined target process models in order to detect process deviations.

As a starting point for future work, the work presented can be expanded to include additional components. These additional components include, for example, the adaptation of anomaly detection at the activity and resource level to provide more detailed information on erroneous events and thus improve decision support. Furthermore, the presented classification model can be transferred into a real-time environment and thus enable an operational support of process flows in hospitals.

References

[1] W. van der Aalst, “Process Mining. Data Science in Action.” Springer, Berlin, 2nd edition, 2016.

[2] W. van der Aalst, A.J.M.M. Weijters, L. Maruster, “Workflow Mining: Discovering Process Models From Event Logs.”, IEEE Trans. Knowl. Data Eng. 16 (9), pp. 1128-1142, 2004.

[3] S.J.J. Leemans, D. Fahland, W. van der Aalst, “Discovering Block-Structured Process Models from Event Logs – A Constructive Approach.”, Application and Theory of Petri Nets and Concurrency, pp. 311-329 2013.

[4] M. Hassani, S. Sicca, F. Richter, T. Seidl, “Efficient Process Discovery From Event Streams Using Sequential Pattern Matching.”, IEEE Proc. of SSCI’ 15, pp. 1366-1373, 2015.

[5] A. Maaradji, M. Dumas, M. La Rosa, A. Ostovar, “Detecting Sudden and Gradual Drifts in Business Processes from Execution Traces.”, IEEE Transactions on Knowledge and Data Engineering 29 (10), pp. 2140-2154, 2017.

[6] A. Ostovar, A. Maaradji, M. La Rosa, A.H.M. ter Hofstede, B.F. van Dongen, “Detecting Drift from Event Streams of Unpredictable Business Processes.”, Proc. of ER’16. LNCS vol. 9974, pp. 330-346 , 2016.

[7] L. Ghionna, G. Greco, A. Guzzo, L. Pontieri, “Outlier Detection Techniques for Process Mining Applications.”, Foundations of Intelligent Systems. ISMIS 2008. Lecture Notes in Computer Science vol. 4994, Springer, pp. 150-159, 2008.

[8] J. Wang, S. Song, X. Lin, X. Zhu, J. Pei, “Cleaning Structured Event Logs: A Graph Repair Approach.”, 31st IEEE International Conference on Data Engineering, pp. 30-41, 2015.

[9] R. Conforti, M. La Rosa, A.H.M. ter Hofstede, “Filtering Out Infrequent Behavior from Business Process Event Logs.”, IEEE Transactions on Knowledge and Data Engineering 29 (2), pp. 300-314, 2017.

[10] M.F. Sani, S.J. van Zelst, W. van der Aalst, “Improving Process Discover Results by Filtering Outliers using Conditional Behavioral Probabilities.”, Lecture Notes in Business Information Processing, Springer, 2017.

[11] S.J. van Zelst, M. Fani Sani, A. Ostovar, R. Conforti, M. La Rosa, “Filtering Spurious Events from Event Streams of Business Processes.”, Advanced Information Systems Engineering. CAiSE 2018. Lecture Notes in Computer Science, vol, 10816, pp. 35-52, Tallin, Springer, 2018.

[12] P. Krajsic, B. Franczyk, “Lambda Architecture for Anomaly Detection in Online Process Mining Using Autoencoders”, International Conference on Computational Collective Intelligence, pp. 579-589, 2020.

[13] P. Krajsic, B. Franczyk, “Variational Autoencoder for Anomaly Detection in Event Data in Online Process Mining”, Proceedings of the 23rd International Conference on Enterprise Information Systems, pp. 567-574, 2021.

[14] M.F. Sani, S.J. van Zelst, W. van der Aalst, “Repairing Outlier Behaviour in Event Logs.”, Business Information Systems. BIS 2018. Lecture Notes in Business Information Processing, vol 320., Springer, pp. 115-131, 2018.

[15] Z. Huang, W. Dong, L. Ji, L. Yin, H. Duan, “On local anomaly detection and analysis for clinical pathways.”, Artificial Intelligence in Medicine, Volume 65, Issue 3, pp. 167-177, 2015.

[16] S. Barbon Junior, P. Ceravolo, E. Damiani, G.M. Tavares, “Evaluating Trace Encoding Methods in Process Mining.”, From Data to Models and Back. DataMod 2020. Lecture Notes in Computer Science, vol 12611. Springer, pp. 174-189, 2021.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, “Attention Is All You Need.”, Advances in Neural Information Processing Systems, pp. 5998–6008, 2017.

[18] B. van Dongen, “Real-life event logs - Hospital log.” 4TU.ResearchData. Dataset. https://doi.org/10.4121/uuid:d9769f3d-0ab0-4fb8-803b-0d1120ffcf54 , 2011.

[19] Github, https://github.com/kraphil/filterClassificationModel, 2021.