Business Process Event Log Anomaly Detection based on Statistical Leverage *

Jonghyeon Ko and Marco Comuzzi

Ulsan National Institute of Science and Technology (UNIST)
Ulsan, Republic of Korea
{whd1gus2, mcomuzzi}@unist.ac.kr

Abstract. We present a novel information-theoretic framework to detect anomalous traces in business process event logs. Although information-theoretic approaches to anomaly detection are considered fundamental in data analytics, they have not been considered in the context of event logs. The proposed framework combines a trace-level anomaly score based on statistical leverage, which also gives an indication of the severity of an anomaly, and different ways of setting the value of a threshold to detect anomalous traces. The framework has been first proposed in a traditional offline setting, but we also discuss its extension in the online setting, i.e., when events in a log are considered as a stream.

Keywords: Anomaly Detection · Event Log · Business Process · Information-theoretic Measure · Statistical Leverage

1 Introduction

Business processes span across all departments in an organization, and information logged during the execution of business processes is available in so-called event logs [2]. The events in an event log are the ones that capture the execution of activities that punctuate the flow of each process execution, labeled with their own case identifier. As such, event logs have three key attributes (a case identifier, a timestamp, and an activity label) and additional attributes relevant in a specific domain, such as an identifier of the (human) resource in charge of the execution or supervision of an activity.

Event logs are prone to errors due to different root causes affecting the process execution and/or the logging process, such as system malfunctioning or sub-optimal resource behaviour [2–4]. Low quality event logs can crucially disrupt process mining-based analyses. The process models discovered from a low quality log, in fact, may be highly inaccurate [5] and, more generally, low quality event logs may give a falsified perception of the process that is analysed to stakeholders, resulting in financial loss. In this context, the research field of event log anomaly detection (or anomalous behavior detection) has emerged recently with the aim

of developing methods to identify and possibly correct the anomalies recorded in an event log.

Our work focuses on detecting anomalies at the trace-level in event logs, i.e., anomalous sequences of activities for a case. This problem has been approached initially in the literature relying on a model of *correct* behaviour, captured for instance by a process model or labelled instances from which a model of positive/negative behaviour can be extracted. More recently, however, machine learning-based approaches have tackled this problem more generally as an *unsupervised* problem, without any reliance on an existing process model or labelled data.

In this context, this paper gives a brief overview of our recent and ongoing work on trace-level anomaly detection using statistical leverage. We propose a novel information-theoretic measure of anomaly score of a trace, based on the definition of statistical leverage. Besides supporting the anomaly detection task – through the definition of an appropriate anomaly threshold – an anomaly score is also able to represent the severity of anomalies, which is an aspect that cannot be handled naturally by classification-based approaches. The approach has been customised to both offline and online settings, i.e., treating an event log as a *batch* of events accumulated over a relatively long period, and as a *stream* of events that become available as soon as they captured, respectively.

The content of this paper is compiled from our previous publications on offline [2] and online [3] anomaly detection. This paper is organised as follows. Section 2 introduces the definition of the proposed anomaly score, the anomaly detection method and the experimental results in the offline settings, whereas Section 3 discusses the same in the online settings. Conclusions are drawn in Section 4.

2 Offline settings: anomaly score and anomaly detection

2.1 Anomaly score

In statistics, the *leverage* is a measure capturing how far away one observation is from other observations in a dataset [1]. It has been used as a key support coefficient to develop statistical measures of anomaly scores for tabular numeric data such as the Cook's distance or the Welsch-Kuh distance.

The main idea underpinning our approach is to define an anomaly score for each trace in a log based on the notion of statistical leverage. To calculate the *leverage* of traces in an event log E, this has to be appropriately pre-processed to obtain a numeric matrix of observations $X(E)$. Similarly to other approaches in the literature [5, 6], we consider one-hot sequence encoding of activity labels and zero-padding for trace-level aggregation of events (see Figure 1). Given an event log E and its activity labels $A_E = \{a_1, \ldots, a_K\}$, K dummy attributes $d_{i,j,k}$ are created for each event $e_{i,j}$ in a trace $\sigma_j \subseteq E$. In this way, using one-hot encoding, each trace σ_j of length N_j is encoded into $N_j \times K$ attributes. Then, for trace-level aggregation, cases are aggregated in a $J \times (N^{max} \times K)$ matrix

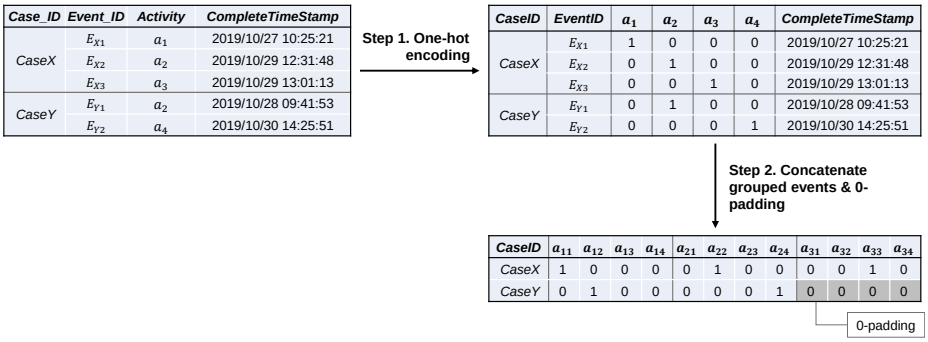

Fig. 1. One-hot encoding and 0-padding

$X(E)$ by zero-padding, where $N^{max} = max_{\sigma_j \in E}(N_j)$ is the length of the longest trace(s) in E.

Then, the *leverage* $\hat{l}(\sigma_j)$ of a trace σ_j is calculated using the matrix $X(E)$. The leverage of the j-th trace in a log is equal to j-th diagonal element of the projection matrix $H(E) = X(E) \cdot (X(E)^T \cdot X(E))^{-1} \cdot X(E)^T$:

$$\hat{l}(\sigma_j) = h_{j,j} \in H(E) \tag{1}$$

However, the leverage $\hat{l}(\sigma_j)$ defined above suffers from a bias due to zero-padding, which increases the leverage of longer traces and decrease the one of shorter traces (which are more likely to be considered similar because they contain many zero-padded values). In order to counter this issue affecting $\hat{l}(\sigma_j)$, the anomaly score that we propose is a weighted version of the leverage $\hat{l}_w(\sigma_j)$, which considers a trace-length weighting factor w_j that can optimally be adjusted using a robustly-fitted model on different real-life event logs. The detailed procedure of defining the weight w_j is described in our published paper [2].

2.2 Thresholding methods

The anomaly score defined above ranges between 0 and 1 and the higher its value the more likely a trace to be anomalous. Therefore, to support anomaly detection, a threshold has to be chosen that discriminates between anomalous (above threshold) and normal (below threshold) traces.

We propose 3 different methods to define such a threshold: (i) a deviation-based threshold ($M1$), (ii) a statistical distribution-based threshold ($M2$), and (iii) a distributional gap-based threshold ($M3$).

In $M1$, the threshold T is defined by the mean and standard deviation of the anomaly scores of all traces in a log. In particular, a trace is considered anomalous if its score is greater than the sum of the average and the standard deviation of the anomaly scores of all traces in a log, i.e., $T = mean_{\sigma_j \subseteq E}[\hat{l}_w(\sigma_j)] + stdev_{\sigma_j \subseteq E}(\hat{l}_w(\sigma_j))$. In $M2$, the threshold is defined as the 0.9 quantile of the probability density function of a gamma distribution fitted using the anomaly

scores of all traces in a log. Finally, in $M3$ the threshold is chosen empirically as the first stationary point of the expected cumulative denisty function of the anomaly score values. The latter method $M3$, in particular, is robust because it does not require any prior knowledge about the ratio of anomalies in an event log, and therefore it is easy to use due to its non-parametric design.

2.3 Evaluation

The proposed anomaly detection framework, i.e., the anomaly score and the anomaly threshold to support anomaly detection, has been evaluated on artificial and real life event logs against state of the art anomaly detection techniques (heuristic methods, classification-based and deep learning-based methods). In a nutshell, the evaluation has found that:

- For real logs, the proposed framework shows the highest average F1-score, compared with the state of the art baselines, while, for artificial logs, the performance is comparable with the one of best baselines;
- The proposed framework is better than the state of the art baselines at maintaining a good performance particularly with low anomaly ratios in event logs; it also shows robust performance regardless of the anomaly ratio, while the performance of the baselines tends to degrade for lower anomaly ratios;
- The proposed framework shows a balanced performance in respect of recall and precision, with $recall/1.5 \leq precision \leq 1.5 \cdot recall$ in most experiments, while other baselines often show skewed performance towards one of the two performance measures.

More details about the evaluation are reported in our original publication [2].

3 Adapting the framework to online settings

3.1 Online anomaly detection

Anomaly detection in online settings can be crucial for discovering anomalies in process execution as soon as they occur and, consequently, allowing to promptly take early corrective actions. However, the online settings introduce additional challenges such as requirements of adaptability to concept drift, and the finite memory usage.

To cope with these challenges, we have adapted the framework described in the previous section to online settings by considering the two basic concepts of grace period and (trace-based) sliding window. The parameter Grace Period (GP) specifies a minimum number of traces to be completed, i.e., for which all the events have to be received, before the framework can be evaluated, which prevents running the anomaly detection model at early stages with an insufficient number of received events. The Sliding Window (SW) aims at keeping the finite

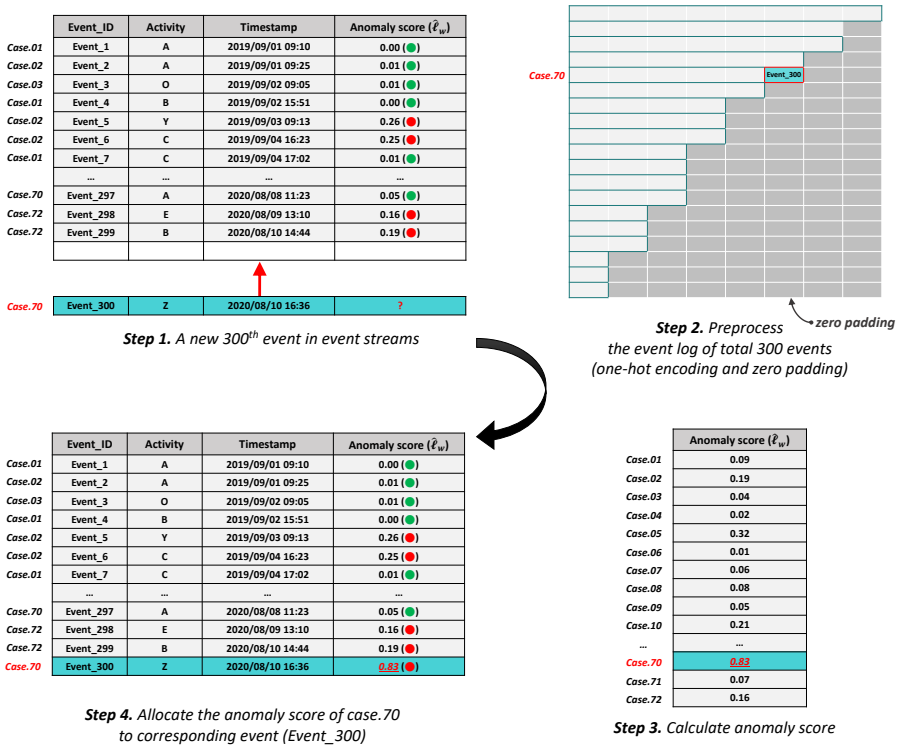

Fig. 2. Example of anomaly detection using a fixed anomaly threshold $T = 0.1$

memory usage by maintaining only the events of a finite number of recent traces for the analysis.

After having introduced the basic concepts of grace period and (trace-based) sliding window, we now introduce the main procedure for the online anomaly detection in Figure 1. Each time a new event is received, we get a sliding window data E_t satisfied with the two parameters GP and W. Then, using the anomaly scoe defined in the previous Section 2, after applying one-hot encoding & zero-padding to E_t, we calculate an anomaly score for all traces $\hat{l}_w(\sigma_j)$ in the sliding window.

For a quicker and fully-automated way of detecting anomalies in online settings, as far as thresholds are concerned, we consider three constant thresholds and one variable threshold. We consider the constant values $T_{c1} = 0.1$, $T_{c2} = 0.15$, and $T_{c3} = 0.2$ and, as variable threshold, the value $T_v = mean_{\sigma_j \subseteq E_t}[\hat{l}_w(\sigma_j)] + stdev_{\sigma_j \subseteq E_t}(\hat{l}_w(\sigma_j))$, which calculates the threshold based on the mean and standard deviation of the leverage scores of all traces in the sliding window. Based on a set threshold, a trace σ_j is labelled as anomalous if $\hat{l}_w(\sigma_j) > T$. This procedure is replicated each time a new event is received.

3.2 Evaluation

The proposed online framework for anomaly detection has been evaluated on artificial and a real life event log. The results obtained can be summarised as follows:

- Since event logs are structured by instance observations, not point observations, the performance of the proposed framework becomes high and stable when a sufficient number of events have been received for all traces in the window SW;
- The performance (accuracy) of the framework tends to improve as the size of the sliding window increases, although this also leads to increased run time;
- The proposed framework shows better recall than precision or F1-score, i.e., the framework is good at recognising correctly the anomalous traces, but often mistakes non-anomalous traces for anomalous ones, therefore creating false positives.

More details about this evaluation are available in our original paper [3].

4 Conclusions

This paper has presented a novel information-theoretic framework for anomaly detection of anomalous traces in business process event logs. The proposed framework has been applied to both online and offline event log settings. The results have shown proposed framework is robust on performance across different event logs and even with low anomaly ratios, as well as it can be easily used even by less experienced data analysts owing to its non-parametric design.

References

1. Everitt, B.: The cambridge dictionary of statistics cambridge university press. Cambridge, UK Google Scholar (1998)
2. Ko, J., Comuzzi, M.: Detecting anomalies in business process event logs using statistical leverage. Information Sciences **549**, 53–67 (2021)
3. Ko, J., Comuzzi, M.: Online anomaly detection using statistical leverage for streaming business process events. In: Process Mining Workshops: ICPM 2020 International Workshops, Padua, Italy, October 5–8, 2020, Revised Selected Papers. vol. 406, p. 193. Springer Nature (2021)
4. Ko, J., Lee, J., Comuzzi, M.: Air-bagel: An interactive root cause-based anomaly generator for event logs. In: Proceedings of International Conference on Process Mining (ICPM) Demo Track (2020)
5. Nguyen, H.T.C., Lee, S., Kim, J., Ko, J., Comuzzi, M.: Autoencoders for improving quality of process event logs. Expert Systems with Applications **131**, 132–147 (2019)
6. Nolle, T., Luettgen, S., Seeliger, A., Mühlhäuser, M.: Binet: Multi-perspective business process anomaly classification. Information Systems p. 101458 (2019)