Information Sciences 549 (2021) 53–67

Contents lists available at ScienceDirect

Information Sciences

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

Detecting anomalies in business process event logs using statistical leverage

Jonghyeon Ko, Marco Comuzzi *

Department of Industrial Engineering, Ulsan National Institute of Science and Technology, Republic of Korea

ARTICLE INFO

Article history:
Received 29 May 2020
Received in revised form 20 October 2020
Accepted 12 November 2020
Available online 28 November 2020

Keywords:
Business process event log
Anomaly score
Case anomaly detection
Statistical leverage
Information-theoretic measure

ABSTRACT

The presence of anomalous information in a business process event log, such as missing, duplicated or swapped events, hampers the possibility of extracting useful insights from event log analysis. A number of approaches exist in the literature to detect anomalous cases in event logs based on different paradigms, such as probabilistic, distance-based or reconstruction-based anomaly detection. This paper proposes a novel method for anomaly detection in event logs based on the information-theoretic paradigm, which has not been considered before in event log anomaly detection. In particular, we propose an anomaly score for cases of a process based on statistical leverage and three different methods to set the anomaly detection threshold. The proposed approach does not require large data sets to train machine learning models, which are necessary for instance in reconstruction-based approaches. The proposed approach shows remarkable anomaly detection capability in experiments conducted using publicly available event logs in respect of existing methods in the literature. One of the proposed anomaly detection thresholds also shows to handle variable case anomaly ratios more effectively than other methods in the literature.

1. Introduction

Process mining [1,2] or, more generally, evidence-based business process management [3], relies on the availability of so-called *event logs*, which store events captured during the execution of business processes to support process analysis and improvement. Events in an event log are usually characterised by a timestamp, an identifier of the activity that was executed, 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 [4], which can stem from a variety of root causes, such as system malfunctioning or human mistakes. These result in different types of errors, such as abnormal activity labels, or missing, duplicated, and swapped events [5]. Errors in event logs hamper the possibility of extracting useful insights from event log analysis. Hence, a new stream of research has emerged in recent years that deals with event log *cleaning*. Strictly speaking, the problem of event cleaning is the one of detecting anomalous information in an event log, i.e., anomalous process cases or events. This can be distinguished from event log *reconstruction*, which deals with imputing correct values to be used in place of the ones missing or detected as anomalous [4,6,7]. This paper focuses on the former problem, which can be also referred to as event log

* Corresponding author.
E-mail address: mcomuzzi@unist.ac.kr (M. Comuzzi).

https://doi.org/10.1016/j.ins.2020.11.017
0020-0255/

anomaly detection [8–10]. Specifically, we focus on case-level anomaly detection, that is, detecting anomalous process cases in an event logs, e.g., cases in which events are missing or in which events have been swapped.

Anomaly detection can be model-aware or model-agnostic. Model-aware approaches rely on the availability of a model of positive or negative behaviour. Such models can be given or extrapolated from clean examples of behaviour. For instance, signature-based network anomaly detection [11] is an example of a model-aware approach that uses negative models where suspicious Internet traffic patterns (signatures) are used to identify anomalous behaviour by attackers. In process mining, conformance checking [12] can be seen as a collection of model-aware approaches to anomaly detection that use process models, given or discovered from clean traces, as positive models of process behaviour.

This paper focuses on model-agnostic approaches, which do not rely on the existence of a model of positive or negative behaviour, such as a process model. This type of approaches are particularly relevant in real word situations, where a correct process model or clean traces from which it can be discovered are often unavailable. Existing model-agnostic approaches belong to different paradigms: probabilistic [8,13–16], which create an intermediate process representation, distance-based [6,9], which adopt a notion of distance among traces to evaluate their (a) normality, and reconstruction-based [4,6], which adopt reconstructive machine learning techniques. A fourth type of approach to anomaly detection, which has not been considered extensively with event log data, is the information-theoretic one. This involves the definition of an information-theoretic measure of process cases, which is used to consider cases whose information measures differs excessively in respect of most others as anomalous. Since an information measure is an inherent measure associated with a data set, e.g., an event log, it does not require a large number of observations to be computed. Moreover, normally it requires a lower number of parameters to be set and, if developed correctly, it is more likely to be consistently effective across different data sets of the same type, i.e., different event logs produced by different business processes.

This paper proposes a novel information-theoretic approach for detecting anomalous cases in an event log based on statistical *leverage*. Statistical leverage, introduced by Hoaglin & Welsch [17], has been for a long time a representative support measure to explain how far away one observation is distributed from others in a data set. Many popular measures for anomaly detection in the field of statistics, such as Cook's distance and Welsch-Kuh distance, have been developed based on it.

Owing to the event log complex temporal and sequential relations, traditional distances based on leverage cannot be used effectively as given for event log anomaly detection. For instance, in the context of attribute-level anomaly detection in event logs, Nguyen et al. [4] have shown that Cook's distance, which they use as a baseline for detecting anomalous timestamps, performed very poorly. In order to measure the difference between cases for anomaly detection, this paper proposes a novel measure based on leverage, which takes into account that process cases may have different length and, therefore, the number of features in a pre-processed event log may differ among cases.

A further problem that we deal with in this paper is how to set the value of the anomaly threshold above which a case is considered anomalous. We propose and compare three different ways of setting this threshold that differ in the amount of a priori information that they require.

The proposed anomaly detection framework, i.e., the novel leverage-based measure combined with the anomaly threshold, is evaluated on artificial and real life event logs against state of the art case-level anomaly detection techniques. The results obtained are remarkable, particularly on real life logs.

The paper is organised as follows. Section 2 reviews the related work about anomaly detection with business process event logs. Section 3 formally introduces the problem of event log anomaly detection, while Section 4 presents the framework. Experimental results are presented in Section 5, while conclusions are finally drawn in Section 6.

2. Related work

We restrict the analysis to model-agnostic approaches. Therefore, this section does not consider conformance checking approaches, in which a process model is given or obtainable from clean traces.

Following the nomenclature provided by Pimentel et al. [18], model-agnostic approaches to anomaly detection are divided into five categories, namely probabilistic, distance-based, reconstruction-based, information-theoretic, and domain-specific. In this paper, we are interested in techniques that are general and, as such, can be applied to different event logs produced by different types of business processes. Therefore, we do not consider domain-specific approaches.

Probabilistic approaches create a model of process behaviour from event logs using likelihood- or frequency-based algorithms. This model is then used to assess whether the behaviour described by the events in the log is anomalous as it does not conform to it. Approaches in this category are similar to model-aware approaches. However, a model of the process behaviour in this case is not given or cannot be obtained from clean traces.

Bezerra & Wainer [8] introduce three probabilistic anomaly detection algorithms (Threshold, Iterative, and Sampling) that eliminate anomalous traces one by one utilizing frequency-based process discovery algorithms, such as the heuristic miner, with fixed configuration parameters. Leemans et al. [13] develop an extended process discovery algorithm based on the inductive miner [19] that filters locally infrequent behaviours through log splitting. Genga et al. [15] identify anomalous patterns from a group of anomalous sub-graphs discovered using a conformance checking algorithm that adopts as input the non-anomalous subgraphs discovered by a frequent sub-graph mining (FSM) algorithm. Ghionna et al. [14] develop a clustering algorithm that utilises sub-patterns extracted using the Markov clustering algorithm and that filters cases matching a number of sub-patterns above a certain predefined cut-off threshold. Lu et al. [16] deal with the problem of event-level

anomaly detection, by mapping representative execution graphs and detecting deviating nodes under a predefined frequency threshold. Nolle et al. [6] first have used Hidden Markov models (HMMs) and t-STIDE [20] for anomaly detection in business process event logs. They propose 6 probabilistic models, i.e., Likelihood [21], Naïve, Sampling [8], HMMs, t-STIDE, and t-STIDE+ to detect anomalous event attributes. These approaches too require to fix the value of one or more given parameters, such as thresholds of state transitions and outputs for HMMs, or a threshold α on anomaly scores for the Likelihood method.

To sum up, the probabilistic models have approached anomaly detection discovering graph-based models of the behaviour described by event logs and filtering anomalous traces using a fitness score. This implies that the anomaly detection performance is controlled by the parameters set for discovering the models. Moreover, these approaches need cut-off thresholds set a priori, such as an estimate of the number of anomalous cases that exist in an event log, which can be challenging to identify.

Distance-based methods exploit a notion of distance between observations to measure their similarity and group them. Observations that do not fit the groups identified are then considered anomalous. This type of techniques have been adopted rarely with event log data. Nevertheless, a few approaches in the literature have adopted standard clustering algorithms, such as k-Nearest Neighbour (kNN) and One-Class Support Vector Machine (OC-SVM).

Sureka [9] transforms a trace into a sequence of strings using activity labels to calculate a distance between each pair of sequences using the normalized longest common subsequence (nLCS) algorithm. Then, using an idea inspired by the kNN algorithm, the author proposes an algorithm to find the k-th nearest points to a a given trace. The anomaly score of a trace is defined as the inverse of the nLCS measure between a trace and its k-th nearest trace. Nolle et al. [6] use the OC-SVM method to divide observations into a positive region, with high density of observations, and a negative one, with low density of observations. This is done by transforming an event log into an integer-encoded matrix using one-hot encoding and zero-padding. Both approaches also need to set the values of parameters a priori, such as the number of neighbors k or the threshold value nu for OC-SVM, which implicitly determines the number of anomalous traces that will be detected.

Reconstruction-based approaches use reconstructive machine learning techniques to detect anomalous process behaviour. Reconstructive techniques, such as autoencoders, are machine learning algorithms that reconstruct their own input, i.e., which try to learn an identity function. Data points characterised by higher reconstruction error are then identified as anomalies in the input data set. Nolle et al. [6] have applied autoencoders to detect artificially-generated trace anomalies in business process event logs. Nguyen et al. [4] compare different autoencoder architectures in detecting attribute-level anomalies in event logs. Anomalies in this work may also concern timestamps. Nolle et al. [7] have developed BINet, which is a recurrent neural network architecture to detect trace- and attribute-level anomalies. BINet identifies anomalies by predicting next events in a trace and comparing these to the actual events registered in an event log.

Similarly to probabilistic approaches, the distance-based and reconstruction-based approaches strongly rely on how well the underlying learning model's performance fluctuates according to the values chosen for several hyperparameters. Moreover, particularly in the case of reconstruction-based approaches, a large amount of data is normally required to train a model.

Finally, information-theoretic approaches to anomaly detection have not been applied extensively to business process event log data. The entropy of event logs, as defined in [22], can be seen as an information-theoretic measure of the content of an event log and, in particular, process traces. However, the entropy is an absolute measure of the information content of a trace and it does not serve well the use case of anomaly detection, which requires comparative measures, i.e., measures of the information content of individual traces that can be compared among each other to discriminate between anomalous and non-anomalous trace behaviour.

3. Problem definition

An event log E contains distinct events. An event is a tuple $e_{i,j} = (c, t, a)$, where c is the id of the process case to which e belongs, t is the timestamp at which e has been recorded and a is the event type, e.g., the activity that was executed. Note that an event may contain additional attributes, which however we do not consider in this work. An event belongs to one particular trace, i.e., the sequence of events executed in a particular process case. The indexes i and j identify the position in the trace and the trace to which an event belongs, respectively.

We use the notation $\#_x(e)$, with $x \in \{c, t, a\}$, to refer to the value assumed by a particular attribute x in an event e. For example, for an event e, we can have $\#_c(e)$ = A1000 and $\#_t(e)$ = *2019–03-09 12:00:00 AM* to signify that event e belongs to case number A1000 and has been logged at midnight of March 9th, 2019. We refer to $\mathscr{E}$ as the universe of events, $\mathscr{A}$ as the universe of event types, and to $A_E \subseteq \mathscr{A}$ as the set of activity labels in an event log E, that is, $A_E = \{l \in \mathscr{A} : \exists e \in E \text{ s.t. } \#_a(e) = l\}$.

A trace σ_j is the sequence of N_j events executed in a particular process case in an event log E, i.e., $\sigma_j = \left[e_{1,j}, \ldots, e_{n,j}, \ldots, e_{N_j,j}\right]$, with $\#_t(e_{i+1,j}) > \#_t(e_{i,j}), \forall i \in [1, N_j - 1]$ and $\#_c(e_{i,j}) = \#_c(e_{k,j}), \forall i, k \in [1, N_j]$. We refer to $\mathscr{S}$ as the universe of event sequences. We use J to denote the number of traces in an event log E.

To test a model-agnostic approach to anomaly detection in event logs, researchers normally inject existing event logs, assumed to be *clean*, with different types of anomalies, using one or more anomaly *patterns* [4,6–8,21]. In this work, we consider 6 case-level anomaly patterns, i.e., *Replace*, *Skip*, *Rework*, *Early*, *Late*, and *Insert* that have already been considered by

previous research [4,7,8,21]. These are non-trivial anomalies that are likely to be introduced by real world root causes and that cannot be easily identified by frequency-based filters typically available in commercial process mining products. The 6 patterns are exemplified in Fig. 1 and defined in detail in Table 1.

To define the problem of case anomaly detection, we introduce a binary label associated with traces in an event log, which evaluates to 1 if a case is anomalous and to 0 otherwise. Given an event log, the problem of case anomaly detection is the one of learning or estimating the function $l : \mathscr{S} \longrightarrow \{0, 1\}$:

$$l(\sigma_j) = \begin{cases} 1 & \text{if } \sigma_j \text{ is anomalous} \\ 0 & \text{otherwise} \end{cases}$$

4. Framework

This section describes the proposed leverage-based case anomaly score (Section 4.1) and and the methods that we propose to set the anomaly detection thresholds (Section 4.2).

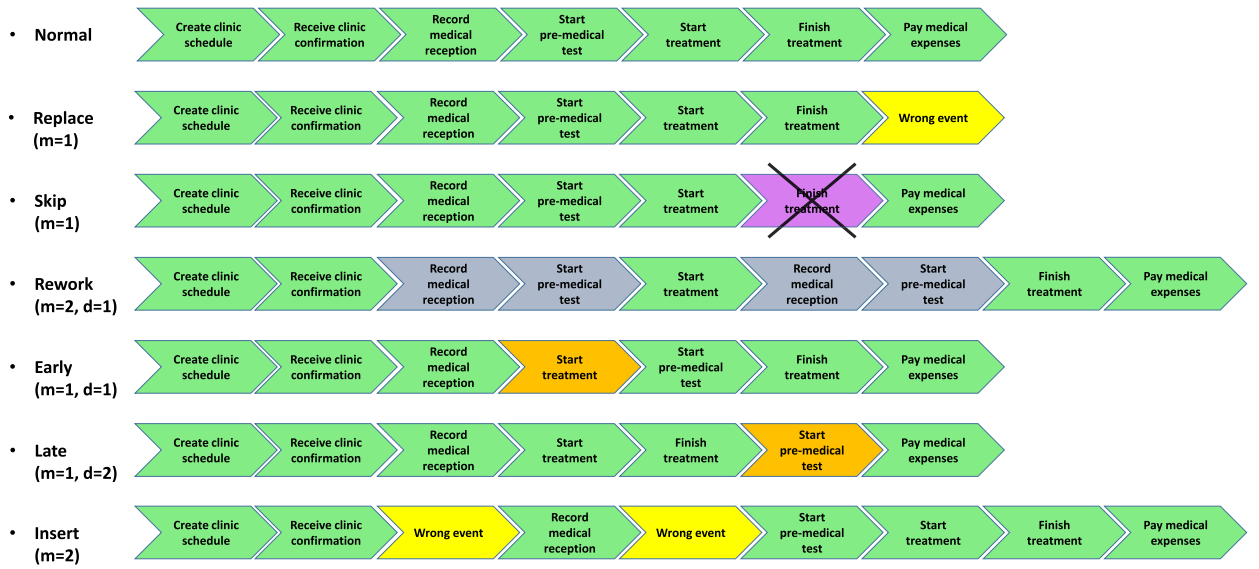


Fig. 1. Case-level anomaly patterns: example.

Table 1
Definitions of the 6 types of case-level anomaly patterns.

Pattern	Definition
replace(σ, M)	Given a trace σ_j of length N events and a positive integer M, one event $e_{i,j} \in \sigma_j$ is randomly chosen. Then, an integer value m is randomly chosen in $[1, min(N-i, M)+1]$. Finally, $\#_a(e_{k,j})$, with $k = i, \ldots, i+m-1$, are changed to a different activity label in A_E.
skip(σ, M)	Given a trace σ_j of length N events and a positive integer M, one event $e_{i,j} \in \sigma_j$ is first randomly chosen. Then, an integer value m is randomly chosen in $[1, min(N-i, M)+1]$. Finally, the m events $e_{k,j} \in \sigma_j$, with $k = i, \ldots, i+m-1$, are deleted from E.
rework(σ, M, D)	Given a trace σ of length N events and positive integers M and D, one event $e_{i,j} \in \sigma_j$ is first randomly chosen. Then, integer values m and d are randomly chosen in $[2, min(N-i, M)+1]$ and $[0, min(N-i-m, D)+1]$, respectively. Finally, the m events $e_{k,j} \in \sigma_j$, with $k = i, \ldots, i+m-1$, are repeated after the event $e_{i+m+d-1,j}$.
early(σ, M, D)	Given a trace σ of length N events, and positive integers M and D, one event $e_{i,j} \in \sigma$ is first randomly chosen. Then, integer values m and d are randomly chosen in $[1, min(N-i, M)+1]$ and $[1, min(i-1, D)]$, respectively. Finally, the timestamps $\#_t(e_{k,j})$ of the m events e_k, with $k = i, \ldots, i+m-1$, are randomly modified so that these events are moved back of $i-d$-th positions in σ_j.
late(σ, M, D)	Given a trace σ_j of length N events and positive integers M and D, one event $e_{i,j} \in \sigma$ is first randomly chosen. Then, integer values m and d are randomly chosen in $[1, min(N-i, M)+1]$ and $[1, min(N-i, D)]$, respectively. Finally, the timestamps $\#_t(e_{k,j})$ of the m events e_k, with $k = i, \ldots, i+m-1$, are randomly modified so that these events are moved forward of $i+d$ positions in σ_j.
insert(σ, M)	Given a trace σ_j of length N events and a positive integer M, an integer value m is first randomly chosen in $[1, M]$. Then, m events $e_{k,j} \in \sigma_j$, with $k \in [1, M]$, are randomly chosen as places to insert an anomalous event. Finally, the m events are generated using random activity labels in A_E and timestamps such that each of the events is inserted after the m respective events $e_{k,j}$ with $k \in [1, M]$.

4.1. Anomaly score

In statistics, *leverage* is a measure capturing how far away one observation is from other observations in a data set [23]. It has been used to detect outliers in models that adopt different techniques, such as linear regression [17,24], ANOVA [17], principal component analysis [25], and sampling methods [26–28]. It is also used as a support measure to develop influence functions such as the Cook's distance, the Welsch-Kuh Distance, and the Welsch's Distance.

Given a matrix $X \in \mathbb{R}^{J\times I}$ with J observations and I attributes (or features), the leverage of the j-th observation in X is the j-th diagonal element $h_{j,j}$ of the projection matrix $H = X(X^TX)^{-1}X^T$. Note that H is symmetric and idempotent and it is a square matrix of size J. The diagonal elements $h_{j,j} \in H$ represent a standardised measure capturing how much the j-th observation lies from other observations in a data set. In particular, $0 \leqslant h_{j,j} \leqslant 1, \forall j$, and $\sum_j h_{j,j} = I$, that is, the diagonal elements of H sum to the number of columns I.

The first issue that we face when developing a leverage-based anomaly measure to process event logs is that, while the objective is to develop a measure for case anomaly, observations (rows) in an event log are events. Therefore, an event log E has to be appropriately pre-processed to obtain a matrix of observations $X(E)$ that can be used to calculate a projection matrix $H(E)$. In particular, E must be pre-processed in such a way that each observation in $X(E)$ is constituted by features obtained from an individual trace in E. Pre-processing involves two steps: (i) *encoding* of events in a trace and (ii) aggregation of encoded traces (see an example in Fig. 2).

The anomalies that we consider in this paper are related to the sequence of activities in a case. Therefore, as far as encoding of information in events is concerned, we only focus on the activity attribute of events $\#_a(e)$. Specifically, similarly to other approaches in the literature [4,7], we consider one-hot sequence encoding of activity labels in events. One-hot encoding, also referred to as integer or dummy encoding, transforms a categorical attribute into a set of dummy numerical attributes. 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 $\sigma_j \subseteq E$. Given $\#_a(e_{i,j}) = a_{\bar{k}} \in A_E$, the dummy attribute $d_{i,j,\bar{k}}$ is set to 1, whereas other dummy attributes are set to 0:

$$d_{i,j,k} = \begin{cases} 1 & \text{if } \#_a(e_{i,j}) = a_k \\ 0 & \text{otherwise} \end{cases}$$

In this way, each trace σ_j of length N_j is encoded into $N_j \times K$ features.

A second issue that we now face to use the encoded cases as input for calculating a projection matrix is that each trace is encoded in a vector of variable length $N_j \times K$. In order to aggregate encoded traces into a matrix of observations $X(E)$, zero-padding is applied to bring all encoded cases to the same length. Specifically, zero-padding is applied to those encoded traces shorter than the longest trace(s) in E. As a result, cases are aggregated in a $J \times (N^{max} \times K)$ matrix $X(E)$, where $N^{max} = max_{\sigma_j \in E}(N_j)$ is the length of the longest trace(s) in E.

We can now define a first leverage-based case anomaly score $\hat{l}(\sigma_j)$ simply as j-the diagonal element h_j of the projection matrix $H(E) = X(E) \cdot (X(E)^T \cdot X(E))^{-1} \cdot X(E)^T$:

$$\hat{l}(\sigma_j) = h_{j,j}$$

Zero-padding introduces an additional problem affecting this first anomaly score $\hat{l}(\sigma_j)$. Because the leverage of observations sum to the number of attributes I, as long as the length of traces in E is not all same (see Fig. 3(a)), the zeros introduced by zero-padding create a *seesaw* effect that increases the leverage of longer traces and decrease the one of shorter traces.

In order to counter this issue affecting $\hat{l}(\sigma_j)$, we propose a new measure $\hat{l}_w(\sigma_j)$ that considers a trace-length weighting factor. Since the cases with shorter trace length have lower leverage than normal because of 0-padding, a weight should

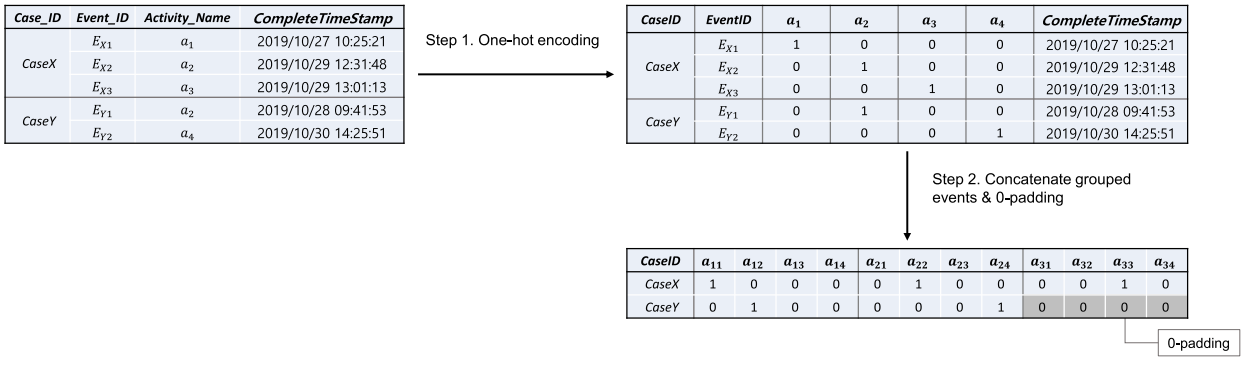
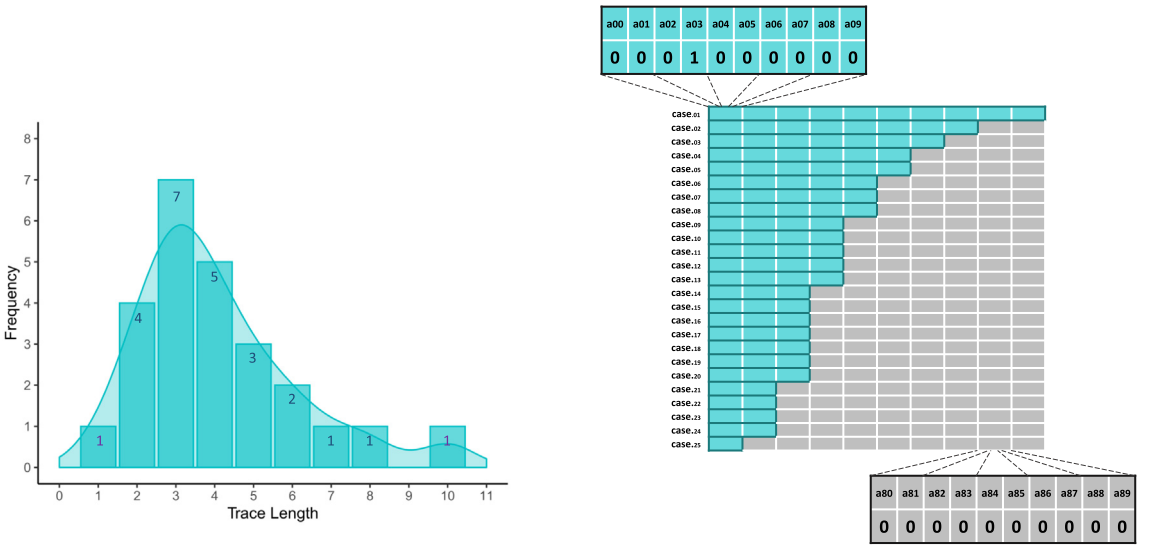


Fig. 2. Encoding and aggregation of events in pre-processing.



(a) Histogram for trace length (b) Preprocessed matrix of log

Fig. 3. Seesaw effect of of zero-padding.

increase the leverage of these shorter traces in respect of the longer ones. We design a weighting factor w_j for adjusting the leverage of a trace σ_j using a combination of trace length scaling and a fitted power coefficient.

Trace length scaling aims at generalising the distribution of trace length for different event logs. In order to define a weighting factor in the range $[0, 1]$, we scale the trace length applying a sigmoid normalisation function, which first normalises the length of traces N_j to their average $mean[N_j]$, i.e., a Z-transformation, and then scales the obtained results in the range $[0, 1]$, using a sigmoid function:

$$Z_j = \frac{\{N_j - mean[N]\}}{stdev[N]} \tag{1}$$

$$sig(Z_j) = \frac{1}{1 + e^{-Z_j}} \tag{2}$$

This normalisation technique generally is used to improve the fit accuracy while limiting the computational complexity [29]. It also provides a more uniform scaling of large values compared, for instance, to MinMax normalisation, which would flatten all values above a certain threshold to 1 [29,30].

The weighting factor w_j is then defined as:

$$w_j = \left[1 - sig(Z_j)\right]^{c(N^{max})} \tag{3}$$

Eq. (3) introduces a power coefficient $c(N^{max})$, which varies with the maximum trace length N^{max} in an event log. This coefficient is introduced to scale the weighting factor according to the maximum trace length in E. Without this coefficient, the normalised weighting factor cannot differentiate the adjustment among event logs with different maximum trace length, because the Z transformation unifies distributions of trace length imposing $mean = 0$ and $stdev = 1$ for all event logs.

Ideally, an optimal value of $c(N^{max})$ should be estimated for each event log, using for instance additional event log data from the same process or similar ones. This is impractical and often impossible in real world situations, since such additional data may not be available. Moreover, since N^{max} does not have an upper limit, we cannot formulate a finite state optimisation that could hold for all event logs, but we only can estimate the optimal value of $c(N^{max})$ that maximises the anomaly detection performance. Since we have no means to assume a linear relation between N^{max} and $c(N^{max})$, we have considered a simple non-linear regression equation $f(x) = a + x^b$, controlled by two parameters a and b. A lower number of parameters to estimate increases the generalisability of the model, while possibly reducing its fitness. Considering the F1-score (calculated as shown in Section 5.3) as anomaly detection performance measure, we have estimated the value of $c(N^{max})$ that maximises this performance measure using as samples 6 real event logs[1] publicly available and commonly used in process mining research. The results of this non-linear regression are reported in Eq. (4). The estimates of a and b in Eq. (4) are significant under significance level 0.01.

[1] These event logs belong to the ones made available by the Business Process Intelligence Challenge in 2012, 2013 and 2017.

$$c(N^{max}) = \begin{cases} -2.2822 + (N^{max})^{0.3422} & \text{if } N^{max} > \frac{2.2822}{0.3422} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Because its parameters are estimated using different types of event logs, we argue that the values in Eq. (4) can be used generally with any event log other than the ones considered in the evaluation of this paper. In some cases, estimating a value for $c(N^{max})$ considering event logs more *similar*, for instance in terms of case level variability and trace length frequency, to the ones on which anomaly detection is applied, may lead to a better estimate.

To conclude, we can now define a weighted leverage measure using the weight w_j defined above, that is:

$$\hat{l}_w(\sigma_j) = w_j \cdot \hat{l}(\sigma_j) \tag{5}$$

4.2. Determining anomalous cases

After having calculated an anomaly score for all traces in an event log E, the final step to complete an anomaly detection framework is to determine, based on the values of the score, which cases should be considered anomalous. In some situations, anomaly detection can be a subjective task, in which different users may identify different patterns as anomalous to a different extent [31]. In this work, however, we focus on automatic classification of anomalous cases and, therefore, we provide next three different ways to detect anomalous cases only based on their anomaly score.

4.2.1. Method 1: anomaly score threshold

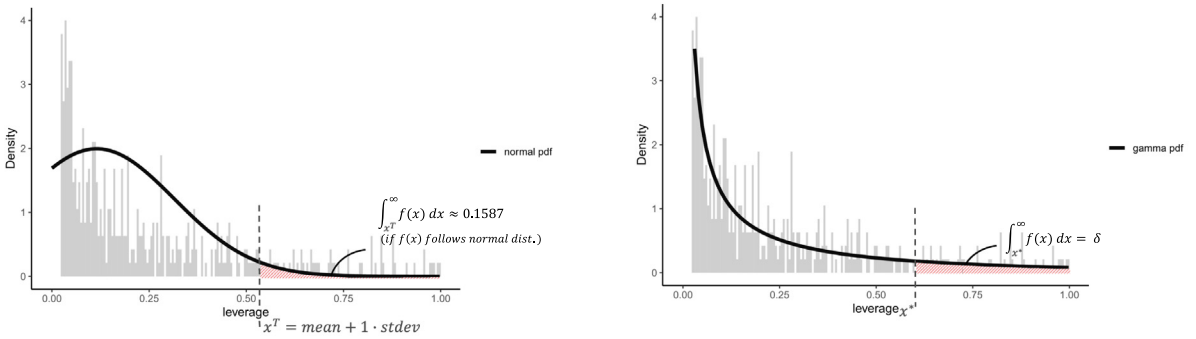
The first method ($M1$) sets a numerical threshold T; if the leverage of a trace is higher than T, then the corresponding case is considered anomalous. Similarly to the threshold used in timestamp anomaly detection in [4], we consider $T = mean_{\sigma_j \subseteq E}[\hat{l}_w(\sigma_j)] + \alpha \cdot stdev_{\sigma_j \subseteq E}[\hat{l}_w(\sigma_j)]$, with $\alpha = 1$. That is, a case is considered anomalous if it exceeds the expected value of the anomaly score of one standard deviation or more. The value of the parameter α can be used to adjust the value of T. Note that here we assume that the values of the leverage-based anomaly scores are normally distributed and that a leverage value signals an anomalous case when falling out of the quantile $Q(\alpha)$ of the normal probability density function (see Fig. 4(a); note that $Q(\alpha = 1) = 0.8413 = 1 - 0.1587$).

4.2.2. Method 2: fitting a gamma distribution

For the second method ($M2$), we consider a traditional approach of statistics that identifies outliers by approximating an underlying known univariate distribution of data [32]. This is done by fitting the anomaly scores obtained on a parametric gamma distribution. The gamma distribution is particularly suitable in situations where outliers belong to a *long tail*, i.e., they are characterised by high values of a given anomaly score [33]. It is described by two parameters (shape and rate) and it is often a better choice than the exponential distribution, which also fits with long tail data, but it is defined only by one parameter (λ). As such, the gamma distribution is able to fit more extreme distributions of anomaly scores compared to the exponential distribution. Given the two parameters shape α and rate β, the gamma probability density function is defined in Eq. (6).

$$f(x; \alpha, \beta) = \frac{\beta^{\alpha} \cdot x^{\alpha-1} \cdot e^{-\beta \cdot x}}{(\alpha - 1)!} \text{for } x \geqslant 0, \tag{6}$$

The gamma distribution results in the log-likelihood:



(a) Probability density function of normal distribution and corresponding T value of $\alpha = 1$

(b) Probability density function of gamma distribution

Fig. 4. Application of statistical probability density function.

$$\begin{aligned}\ell(x;\alpha,\beta) &= \ln\prod_{i=1}^{n} f(x_i;\alpha,\beta)\\ &= n\alpha\ln\beta - n\ln(\alpha-1)! + (\alpha-1)\sum_{i=1}^{n}\ln x_i - \beta\sum_{i=1}^{n}x_i\end{aligned} \tag{7}$$

The Maximum Likelihood Estimation (MLE) of α and β is obtained solving the non-linear equations $\frac{\partial\ell}{\partial\beta} = 0$and $\frac{\partial\ell}{\partial\alpha} = 0$, respectively. These cannot be solved in closed-form, but they can be solved numerically using the gradient descent rule, obtaining the MLE estimates $\hat{\alpha}$ and $\hat{\beta}$. Using these two estimates, the cumulative probability function is defined as follows:

$$F(x) = \int_0^x f(t)\mathrm{d}t = \frac{\gamma(\hat{\alpha},\hat{\beta}x)}{(\hat{\alpha}-1)!},$$

where $\gamma(s,x) = \int_0^x t^{s-1}e^{-1}\mathrm{d}t$is the lower incomplete gamma function.

Given an anomaly threshold δ, a trace is considered anomalous if it falls outside of the quantile defined by δ of the estimated probability density defined by Eq. (8) (see Fig. 4(b)). That is, a trace σ_j is anomalous if:

$$F(\hat{l}_w(\sigma_j)) > \{1-\delta\}$$

In experiments, we consider $\delta = 0.1$.

4.2.3. Method 3: Using the empirical cumulative density function

The methods $M1$ and $M2$ are not flexible in respect of different ratios of anomalous cases in an event log, because they involve a fixed parameter (α in $M1$ and δ in $M2$). To overcome this limitation, the third method ($M3$) uses the empirical cumulative density function (ecdf) of the leverage $\hat{l}_w(\sigma_j)$ and, in particular, its slope, to identify a proper anomaly detection threshold that does not depend on a fixed parameter. This method is inspired by [7], where the anomaly threshold values are selected by considering significant changes in the derivative of the function relating the thershold value to the predicted anomaly ratio.

As an example, Fig. 5 shows the ecdf and the respective derivative for one of the artificial event logs considered in the experimental evaluation. Note that the y-axis of the ecdf represents the percentage (rate) of cases in an event log that have an anomaly score lower than a given x value. Assuming that there is a distributional difference in the anomaly scores between anomalous and non-anomalous cases, this figure highlights a general pattern in the ecdf of $\hat{l}_w(\sigma_j)$: there is normally a large gap between the highest anomaly score associated with mostly non-anomalous cases and the lowest anomaly score value associated with mostly anomalous cases. In Fig. 5, even though the ecdf is always increasing, the derivative of the ecdf assumes negative values since the stationary points for derivative are small and evenly spaced but the points of anomaly score are unevenly spaced. A reasonable way to set the value of the anomaly threshold is to search for the first negative valley in the derivative of ecdf (0.023 in the figure). Therefore, in $M3$ we consider as anomaly threshold the first relative minimum of the derivative of ecdf. This is calculated numerically. Note that, owing to the fact the ecdf is a discrete function, the

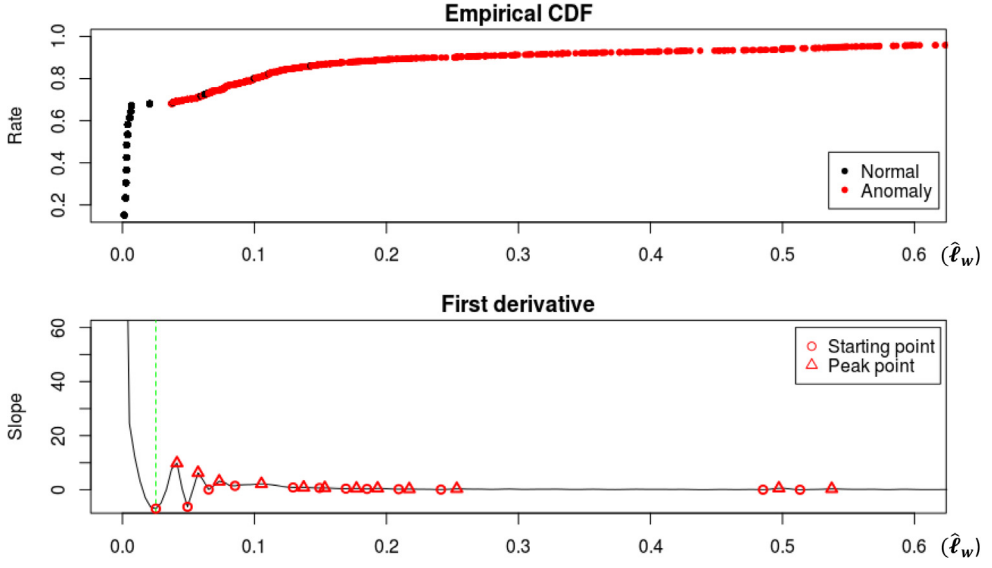


Fig. 5. Example of ecdf, related slope, and identification of anomaly detection threshold (data used: 9th replication of Wide process from [6]).

identification of stationary points depends on a threshold determining a tolerance level on the gaps between points in the ecdf.

5. Evaluation

This section first presents the data sets that we have considered for evaluating the proposed anomaly detection framework (Section 5.1), and the performance metrics and baseline models that we consider in our evaluation (Section 5.2). Then, it presents the experimental results. In particular, we have tested the capability of the proposed leverage-based scores to serve as an information-theoretic measure to identify anomalies (Section 5.3), and, finally, we have compared the performance as an anomaly detection method of our framework in respect of existing proposals in the literature (Section 5.4).

All the data sets and code to reproduce the experiments discussed in this section are publicly available at https://github.com/jonghyeonk/Leverage_Eventlog.

5.1. Data sets

An anomaly detection framework must be generalisable to different types of event logs. As common practice in this research field [4,7,8,10,21], we consider different artificial and real life event logs in our evaluation (see Table 2; note that this table is compiled considering the event logs after they have been injected with anomalies).

As far as artificial logs are concerned, we consider two groups of event logs, i.e., the ones used by Nguyen et al. [4] and Nolle et al. [7] in the evaluation of their work, which we label ART-LOG-1 and ART-LOG-2, respectively. Group ART-LOG-1 provides 20 artificial logs obtained by simulating 2 different business process models (small and large) using the PLG2 tool[2] [34]. Group ART-LOG-2 provides more realistic event logs generated by the simulation of 5 different business process models (Small, Medium, Large, Huge, and Wide) using the PLG2 tool. For each process models, 10 event logs are available.

Regarding real life logs, we consider the event logs made available by the Business Process Intelligence Challenge (BPIC) in 2012, 2013, and 2017. *BPIC 2012* is a log of a loan application process from a Dutch financial institute and *BPIC 2017* is an updated event log of the company using a new workflow system. *BPIC 2013* is a log of an incident and problem management process from Volvo IT. In the remainder, we refer to this group of event logs using the label REAL-LOG.

The anomaly patterns defined in Section 3 are injected into the original event logs on cases randomly selected until a certain ratio of anomalous cases is reached. We consider the ratios 5%, 10%, and 30% in the experiments. Similarly to [4], the ART-LOG-1 logs are injected using only the *Replace* pattern with $M = 1$, so that activity names are randomly replaced by a different one. Similarly to [7], for the logs of ART-LOG-2, one of the 5 patterns *Skip*, *Rework*, *Early*, *Late*, and *Insert* is picked randomly to perturb one case and this procedure is repeated until the target anomaly ratio is reached. The parameters used for each pattern are the following: $M = 2$ for the *Skip* pattern; $M = 3$ and $D = 5$ for the *Rework* pattern; $M = 2$ and $D = 5$ for the *Early* and *Late* patterns; $M = 2$ for the *Insert*.

For logs in REAL-LOG, all 6 anomalous patterns are injected randomly until a given ratio of anomalous cases is obtained, using the same parameter values considered for ART-LOG-1 and ART-LOG-2. Note that real life event logs might already have inherent anomalies before anomaly injection. Nevertheless, we assume that all cases in these logs are normal. While we cannot guarantee that real life logs did not contain anomalies *before* being injected to run our experiments, it has been considered acceptable by previous research, e.g., [4,7,10,21,35], to inject anomalies into the original real life event logs and test the proposed anomaly detection methods on the labelled logs obtained.

5.2. Evaluation metrics and baseline models

As baselines, we compare the performance of the proposed anomaly detection framework against 5 existing methods in the literature. As probabilistic approaches, we consider the Sampling and Naïve method from [8]. These have been re-implemented using the same parameter values of the published paper: the Naïve method uses $cut\,off = 2\%$; the Sampling method uses $cut\,off = 2\%$ and $sampling\,factor = 70\%$. As distance-based approach, we consider the One-Class Support Vector Machine (OC-SVM), which we implemented using the python module 'scikit-learn' with RBF kernel with $gamma = scale$, as in [7]. As reconstruction-based approach, we consider the distributional autoencoder (DAE) and the BINet1 from [6,7], using the same parameters adopted in the original paper: DAE uses parameters $epochs = 50$ and $batch\_size = 500$; BINet1 uses parameters $epochs = 20$ and $batch\_size = 500$. As anomaly detection thresholds, we consider the ones defined for each method in the published papers ($elbow_{\uparrow}$ for DAE and LP_{mean} for BINet1).

Regarding evaluation metrics, a model-agnostic anomaly detection framework can be evaluated similarly to a classifier that aims at predicting correctly a known label for each case (anomalous v. non-anomalous). Note that, in the evaluation phase, the value of this label is known because anomalies are injected before running the experiments. As such, performance is evaluated using the traditional metrics of precision, recall, and F1-score.

[2] http://plg.processmining.it/.

Table 2
Descriptive statistics of event logs (number of activities, cases, events, and variants counted after injecting anomalies).

Group	Event logs	# of logs	# of activities	# of cases	# of events	# of variants
ATR-LOG-1	Small_log	10	14	2 K	28 K	96–506
	Large_log	10	10	15 K	120 K	209–222
ART-LOG-2	Small	10	38–39	5 K	39 K–49 K	172–817
	Medium	10	58–63	5 K	24 K–34 K	174–824
	Large	10	75–83	5 K	52 K–57 K	229–1,269
	Huge	10	86–107	5 K	36 K–46 K	229–1,224
	Wide	10	59–67	5 K	26 K–32 K	152–847
REAL-LOG	BPIC_12	1	71	13 K	262 K-263 K	4,691–5,736
	BPIC_13	3	7–25	0.8 K-7.5 K	2 K-66 K	191–3,018
	BPIC_17	2	15–51	31 K-42 K	194 K-565 K	463–13,281

5.3. Experimental results: suitability of leverage-based score as anomaly score

The objective of the first part of our evaluation is to show the suitability of the proposed leverage-based scores to serve as anomaly scores. Before comparing the performance of the proposed framework against baselines, in fact, we first aim at ensuring that, in ideal conditions, the proposed leverage-based scores are *good* anomaly scores, i.e., they discriminate effectively between anomalous and non-anomalous cases. This part of the evaluation also helps to show that the weighted leverage measure $\hat{l}_w(\sigma)$ normally outscores the non-weighted version $\hat{l}(\sigma)$ as an anomaly score.

To achieve our objective, we run experiments in which, for a given event log, we use different values of the anomaly detection threshold (from 0 to 1 at increasing steps of 0.0001). We then determine the threshold value that achieves the maximum performance (F1-score). Fig. 6 shows the results averaged across event log groups and for different ratios of anomalous case injection (5%, 10% and 30%). Note that, by design, these results can be seen as an upper bound that could be obtained by an ideal anomaly detection method that always identifies an optimal anomaly detection threshold.

For ART-LOG-1, both the weighted and non-weighted leverage scores achieve perfect maximum performance for all anomaly ratios. For ART-LOG-2, the maximum performance is close to perfect (always above 0.93). For real logs the maximum performance is still high particularly in the case of the weighted leverage for high anomaly ratio (30%), for which an average F1-score of 0.74 is achieved.

Interestingly, the results of Fig. 6 on the REAL-LOG group could be explained by the presence of *natural* anomalies in the real logs, as also implied by Nolle et al. [6]. Specifically, the average F1-score for the REAL-LOG group increases with the ratio of anomalies injected, which may appear counter-intuitive as one would expect that more injected anomalies would be harder to identify. Such a performance, however, can be explained by assuming that there exist a sufficiently large amount of hidden natural anomalies in the original event logs, i.e., labelled incorrectly in our evaluation. For example, let us assume that there are hidden n (e.g., 20%) natural anomalies in an event log and that a certain ratio r of anomalies are then injected in the same event log. Then, the n natural anomalies will disturb, i.e., decrease, the performance of the method because of the combined effect of two factors: (i) the framework is designed to assign higher leverage score to the n natural anomalous cases in the original log, but (ii) these natural anomalous cases are labelled as non-anomalous while assessing the performance of the framework (unless, obviously, they are affected by the anomaly injection process). The lower r, the larger this disturbance effect, resulting in lower performance of the framework in the case of real event logs injected with lower anomaly ratios.

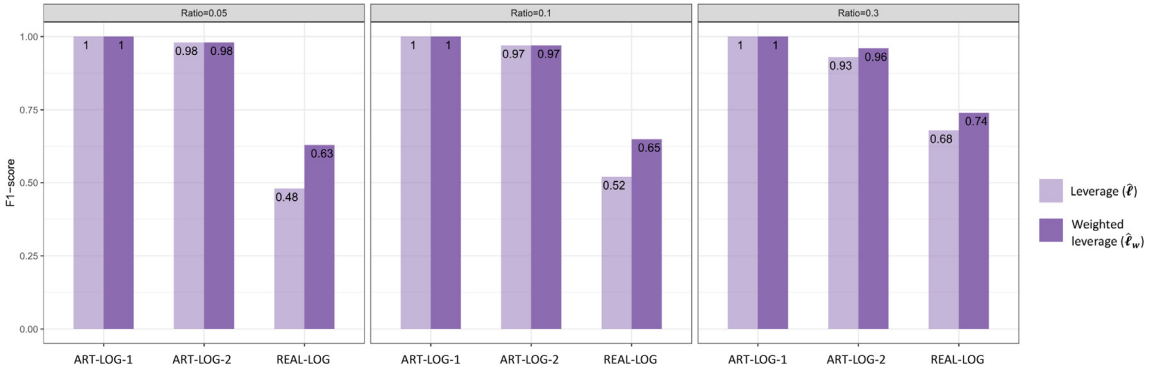


Fig. 6. Maximum performance (F1-score) obtained ex-post, for different event log groups and case anomaly ratios.

Table 3
Performance improvement (↑ F1) for different event logs (N^{max} is the maximum trace length).

Data	ratio	N^{max}	↑ F1	ratio	N^{max}	↑ F1	ratio	N^{max}	↑ F1
Smalllog	0.05	8	0	0.1	8	0	0.3	8	0
Largelog	0.05	14	0	0.1	14	0	0.3	14	0
Small	0.05	13	0	0.1	13	0	0.3	13	0∼+0.062
Medium	0.05	11	0	0.1	11	0	0.3	11	+0.006∼+0.061
Large	0.05	15	0	0.1	15	−0.003∼0	0.3	15	0∼+0.072
Huge	0.05	13∼14	0∼+0.017	0.1	13∼14	0∼+0.013	0.3	13∼14	+0.006∼+0.04
Wide	0.05	10	0	0.1	10	0	0.3	10	+0.004∼+0.172
BPIC12	0.05	175	+0.283	0.1	175	+0.219	0.3	175	+0.12
BPIC13	0.05	22∼123	+0.029∼+0.212	0.1	22∼123	+0.061∼+0.157	0.3	21∼124	−0.001∼+0.046
BPIC17	0.05	8∼61	0∼+0.125	0.1	8∼61	0∼+0.119	0.3	8∼61	0∼+0.07

Table 3 shows the improvement of the F1-score achieved by the weighted leverage score in respect of the non-weighted score, and the relative maximum trace length N^{max}. It can be noticed that the weighted leverage $\hat{l}_w(\sigma)$ achieves consistently[3] equal or better performance than the non-weighted one and that this performance improvement is larger for event logs with higher maximum trace length N^{max}. This demonstrates that the proposed weighting factor generally helps improving the performance of the framework, as intended by design.

5.4. Experimental results: anomaly detection

In this section, we first provide a birds-eye perspective on the experimental results obtained to highlight some general insights. Then, we discuss more specifically how specific design choices regarding the anomaly injection process and the parameterisation of baselines and proposed methods affect the experimental results. Note that the results discussed in this section for the proposed methods ($M1, M2$ and $M3$) consider the use of the weighted leverage score $\hat{l}_w(\sigma)$, which generally outscores the non-weighted version as demonstrated in the previous section.

Fig. 7 shows the average precision, recall and F1-score of the proposed methods and the baselines for different event log groups and different anomaly ratios. As far as artificial logs are concerned (ART-LOG-1 and ART-LOG-2), the proposed methods $M1, M2$ and $M3$ show good F1-score performance, mainly due to a higher precision compared to the baselines. The proposed methods are consistently outscored only by the Sampling method. The performance of Sampling, however, strongly depends on the parameter settings, as discussed later in this section. In the case of real event logs, the proposed methods show better F1-score than all the baselines, owing to a consistently higher precision.

Based on the results of Fig. 7 for the group REAL-LOG, we can divide the proposed methods and the baselines into three categories using the ratio between average precision and recall:

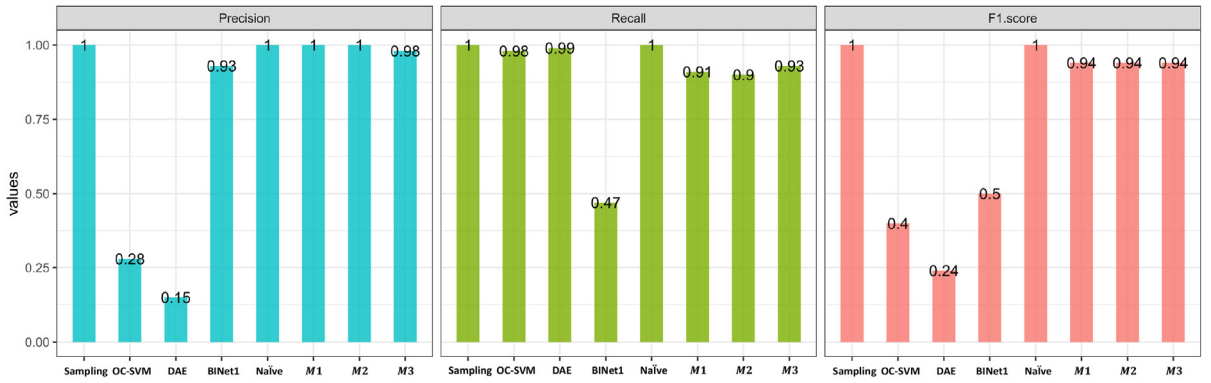
- Positively skewed methods ($M1$ and $M2$): these methods are characterised by $precision/recall > 1.5$; they tend to be skewed towards positive samples, i.e., they are good at recognising correctly the non-anomalous traces, but often fail to recognise anomalous traces;
- Negatively skewed methods (Sampling, OC-SVM, DAE, Naïve): these methods are characterised by $precision/recall < 1.5$ and tend to be skewed towards negative samples, i.e., they are good at recognising correctly the anomalous traces, but often mistake non-anomalous traces for anomalous ones, therefore creating false positives;
- Balanced methods (BINet1, $M3$): these methods are characterised by $recall/1.5 \leqslant precision \leqslant 1.5 \cdot recall$; they tend to show a more balanced performance across positive and negative samples.

Depending on the application scenario, the results presented thus far can help analysts to select a method in the most appropriate category. For instance, positively skewed methods can be chosen when the benefits of higher precision outscore the cost of handling false positives during the anomaly detection phase.

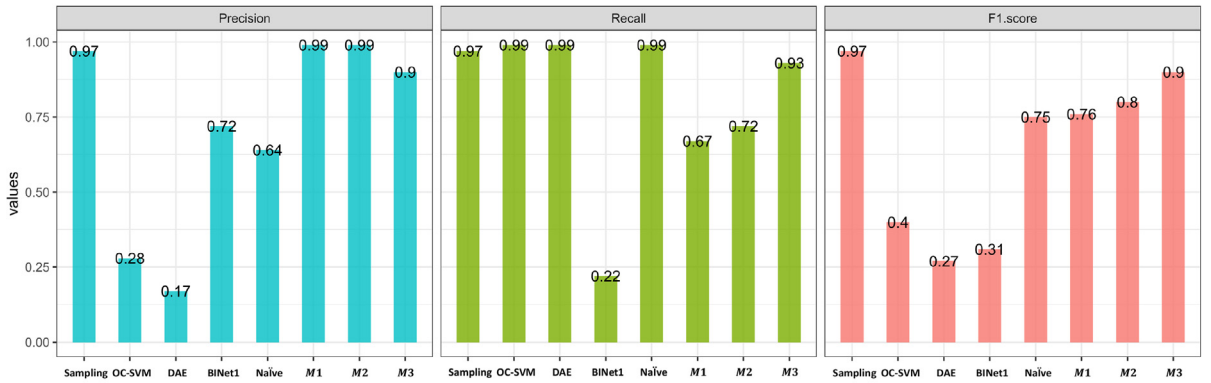
Fig. 8 shows the same results reorganised per event log group and anomaly ratio. This different visualisation highlights the higher stability of the proposed method $M3$ in respect of different anomaly ratios, particularly in the case of real event logs. The difference in the performance achieved for different anomaly ratios, in fact, is lower for $M3$ than for other methods and baselines. This can be explained by considering the fact that $M3$ tries empirically to find the best anomaly detection threshold, without fitting a known distribution type on the anomaly score values. Even in this case, the performance of Sampling appears to be remarkable in the case of artificial logs, but this is due to parameter settings as we explain next.

In the remainder of this section we discuss more in detail some noticeable issues with each of the baselines and the proposed methods that may not appear straightforwardly from the numerical results presented thus far.

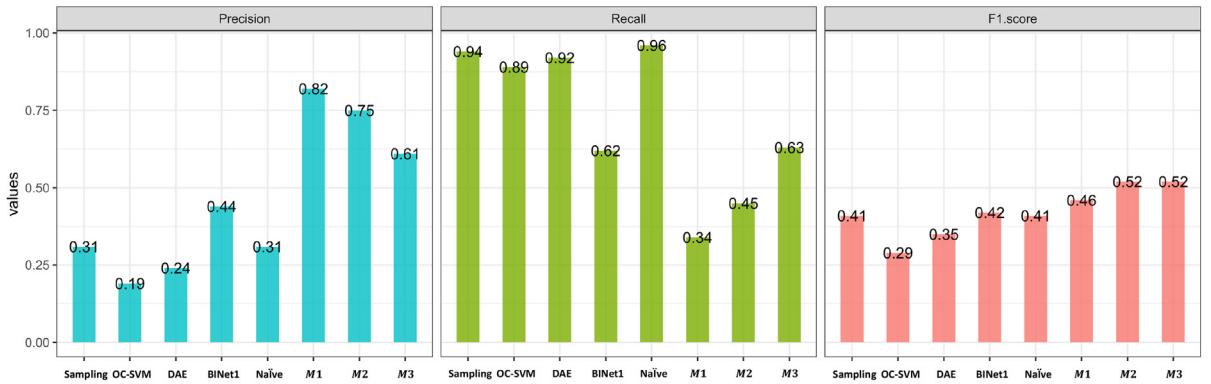
[3] With the exception of 2 logs: one of the *large* logs at 10% anomaly ratio and one of the *BPIC*2013 logs at 30% anomaly ratio, where the performance decreases of 0.003 and 0.001, respectively.



(a) Average precision, recall and F1 score on ART-LOG-1



(b) Average precision, recall and F1 score on ART-LOG-2

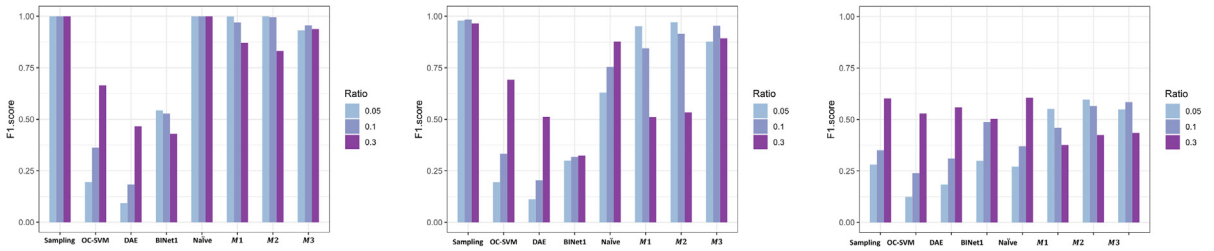


(c) Average precision, recall and F1 score on REAL-LOG

Fig. 7. Performance of each method by event log.

5.4.1. Sampling

As remarked earlier, the excellent performance of the Sampling method on artificial logs may be due to parameter settings in the experiment. Specifically, because of the way in which anomalies are injected, in the case of artificial logs there exist a few, very frequent non-anomalous trace variants and many anomalous trace variants. The ratio of cases (frequency) belonging to an individual anomalous trace variant is always below 2% even when 30% of the cases are injected with anomalies. Given that the sampling method identifies anomalous traces with a cut-off frequency $cut\,off = 2\%$ and filters it to process models with $sampling\,factor = 70\%$, the performance of Sampling is almost perfect for all artificial logs, in which anomalous trace variants are always less than 2% frequent and non-anomalous cases are always at least 70%. Conversely, if anomalous trace variants would be more than 2% frequent, the performance of the sampling method would have decreased sharply as in the case of the group REAL-LOG.



(a) Average F1 score according to anomaly ratio on ART-LOG-1

(b) Average F1 score according to anomaly ratio on ART-LOG-2

(c) Average F1 score according to anomaly ratio on REAL-LOG

Fig. 8. Stability of each method on anomaly ratio by event log.

In the case of real event logs, the performance of the Sampling method tends to be much lower for lower anomaly ratios (see Fig. 8) because this method has no inner means to automatically adjust to the magnitude of the anomaly ratio. In summary, the Sampling method requires that (i) the anomaly ratio and (ii) the frequency rate of anomalous trace variants are known a priori to effectively set the value of the parameters *cut off* and *sampling factor*.

5.4.2. OC-SVM

The OC-SVM method is generally more effective when there are only few anomalies in a data set [36]. The hyperparameter *nu* in OC-SVM controls the rate of positive and negative values, i.e., it serves as an approximate estimated anomaly ratio. Ideally, such a value should be known a priori. In other words, OC-SVM can be used most effectively if the anomaly ratio in a data set is known a priori. In practice, the default value $nu = 0.5$ has been used in our experiments to be consistent with the baseline research [7]. In the experimental results (see Fig. 8), the performance of OC-SVM decreases sharply for lower anomaly ratios because the default value $nu = 0.5$ is too high. Since anomalies are naturally a very small portion in a data set, we suggest to use lower values of *nu* in future research.

Finally, being an approach based on a machine learning algorithm (SVM), OC-SVM requires a large amount of data to be trained effectively and it may therefore underfit in the case of small event logs with only a limited number of traces.

5.4.3. DAE

Similarly to OC-SVM, the performance of DAE sharply decreases for lower anomaly ratios. As designed by Nolle et al. [6], we have used the threshold strategy $elbow_{\uparrow}$, which is applied to the error returned by DAE when comparing the reconstructed input and the original input. While this strategy should be able to react to the change of anomaly ratio, this does not appear in the experimental results. The reason may be that, because of one-hot encoding, the input data set has a high number of features, which limits the ability to train the autoencoder to generalise effectively. Note that this problem can be classified as underfitting [37].

5.4.4. BINet1

The BINet1 method shows good performance on artificial logs, but relatively low performance on the real logs. While this method is also based on deep neural networks like DAE, it shows a more balanced performance across different anomaly ratios. This may be due to the fact that BINet1 adopts a different philosophy than DAE, identifying anomalies by predicting individual events in a trace and comparing it with the original ones.

5.4.5. Naïve

The Naïve method shows excellent performance especially in terms of recall on ART-LOG-1. The performance, however, tends to decrease in ART-LOG-2 and REAL-LOG particularly for lower anomaly ratios. This can be explained by considering that this method has a limitation similar to the one of the Sampling method. The parameter $cut\,off = 2\%$ is used, in fact, to consider as anomalous all the traces belonging to a trace variant less frequent than 2%. The worse performance of Naïve compared to Sampling on ART-LOG-2 may be due to the lack of the second safety net represented by the parameter $sampling\,factor = 70\%$ in the Sampling method. When dealing with real logs, the performance of the Naïve method is comparable to the one of the Sampling method.

5.4.6. Proposed method M1

The performance of method *M*1 is overall satisfactory especially in terms of average precision for all event logs. As previously stated, however, the method *M*1 by design cannot react effectively to different anomaly ratios because of the predefined parameter α set to 1. Differently from other methods, as shown in Fig. 8, the average F1-score in this case is lower in the case of the highest anomaly ratio of 30%. This is due to the fact that $\alpha = 1$ corresponds to an estimated anomaly ratio of 16% when fitting a normal distribution on the leverage score values (see Fig. 8). Therefore, the ability of *M*1 to perform effectively decreases sharply once the anomaly ratio exceeds 16%.

5.4.7. Proposed method M2

The performance of method *M*2 is slightly better than method *M*1 particularly in the case of real logs (see Fig. 7). Similarly to *M*1, however, *M*2 also struggles to adjust to different anomaly ratios, resulting in worse performance for higher ratios. This is explained by the need to set a priori the parameter δ, which is set to the default value 0.1 in experiments. In this case, $\delta = 0.1$ when fitting a gamma distribution means to consider an expected anomaly ratio around 10% (see Fig. 8). Therefore, the performance decreases sharply above this ratio.

5.4.8. Proposed method M3

The method *M*3 has been introduced as an anomaly detection method that does not rely on a priori knowledge of the anomaly ratio in a data set. The experimental results show that *M*3 is not always the best performing method for anomaly detection, but the results of Fig. 8 show that the performance of *M*3 is not affected by the change of anomaly ratio, as intended by design.

With artificial logs ART-LOG-1 and ART-LOG-2, *M*3 shows the best performance in respect of other baselines if we except the Sampling and Naïve methods, which, as we discussed before, may accidentally benefit from their parameter settings. With real logs, the performance of this method is sometimes hampered by the numerical method adopted to find stationary points in the ecdf of the leverage scores. In some cases, in fact, we have found that the identification of stationary points is extremely sensitive to the parameter used in solving numerically the corresponding equation. The optimisation of this aspect falls beyond the scope of this paper. Nevertheless, we can claim that the performance values obtained for M3 in this paper are a lower bound and better performance may be obtained by calibrating the solution of the equation required to find the anomaly detection threshold.

In summary, method *M*3 obtains a satisfactory performance while not facing the problems identified for other methods, such as (i) being unable to adjust to different anomaly ratios, (ii) requiring a priori knowledge about the anomaly ratio, and (iii) requiring enough data to train a machine learning model.

6. Conclusions

This paper has presented a novel information-theoretic framework for anomaly detection of anomalous cases in business process event logs. The results have shown that the performance of the framework is aligned with the one achieved by other baselines. The proposed framework, in particular method *M*3, however, does not impose the need to know a priori the ratio of anomalous cases or the need to collect extensive data to train a machine learning model. Moreover, the proposed framework is better than the baselines at maintaining a decent performance in respect of variations of the anomaly ratio of cases in an event log.

From a practical standpoint, the proposed framework may improve an organisation's capability of detecting anomalous behaviour in its processes, without the need to discover process models or to know a priori the expected ratio of anomalies. In a finance department, process anomalies may directly signal fraudulent behaviour of employees that may go undetected because it does not violate explicitly any corporate policies. In other contexts, identifying process anomalies may lead to process improvement. In a scenario analysed by the authors in the healthcare industry [38], for instance, process anomalies highlight situations in which patients refer to the wrong department to get treatment in a large outpatient department in Korea. If identified correctly, these cases can be avoided, leading to an improvement of performance and customer satisfaction.

Future work will evolve along different research lines. First, the calculation of leverage relies on inverting a large matrix, which can be a problem in case of mega-sized event logs. This problem can be overcome by developing more effective encoding of information in individual traces, requiring a lower number of features, by devising a divide and conquer approach in which leverage scores can be calculated hierarchically, or by comparing scores from clusters of cases. Second, we are investigating methods to reconstruct cases, that is, discovering the type of anomaly that affects a specific case identified as anomalous, by combining compliance checking and trace analysis methods. Finally, we are developing a benchmark to compare the performance of different anomaly detection techniques on pseudo-real anomalies, i.e., simulated using patterns of event anomalies derived from real world root causes of mistakes in event logs, such as the ones identified by [21].

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] A. Weijters, W. M. van der Aalst, Process mining: discovering workflow models from event-based data, in: Belgium-Netherlands Conf. on Artificial Intelligence, Citeseer, 2001..
[2] W. Van Der Aalst, Process mining: discovery, conformance and enhancement of business processes, Vol. 2, Springer, 2011.
[3] J. Recker, Evidence-based business process management: Using digital opportunities to drive organizational innovation, in: BPM-Driving Innovation in a Digital World, Springer, 2015, pp. 129–143..
[4] H.T.C. Nguyen, S. Lee, J. Kim, J. Ko, M. Comuzzi, Autoencoders for improving quality of process event logs, Expert Syst. Appl. 131 (2019) 132–147.
[5] R.S. Mans, W.M. van der Aalst, R.J. Vanwersch, A.J. Moleman, Process mining in healthcare: Data challenges when answering frequently posed questions, in: Process Support and Knowledge Representation in Health Care, Springer, 2012, pp. 140–153.
[6] T. Nolle, S. Luettgen, A. Seeliger, M. Mühlhäuser, Analyzing business process anomalies using autoencoders, Mach. Learn. 107 (11) (2018) 1875–1893.
[7] T. Nolle, S. Luettgen, A. Seeliger, M. Mühlhäuser, Binet: Multi-perspective business process anomaly classification, Inform. Syst. 101458 (2019).
[8] F. Bezerra, J. Wainer, Algorithms for anomaly detection of traces in logs of process aware information systems, Inform. Syst. 38 (1) (2013) 33–44.
[9] A. Sureka, Kernel based sequential data anomaly detection in business process event logs, CoRR abs/1507.01168 (2015) 1–4..
[10] K. Böhmer, S. Rinderle-Ma, Multi instance anomaly detection in business process executions, in: International Conference on Business Process Management, Springer, 2017, pp. 77–93..
[11] M. V. Mahoney, P. K. Chan, Learning rules for anomaly detection of hostile network traffic, in: Third IEEE International Conference on Data Mining, IEEE, 2003, pp. 601–604..
[12] S.J. Leemans, D. Fahland, W.M. Van der Aalst, Scalable process discovery and conformance checking, Softw. Syst. Modeling 17 (2) (2018) 599–631.
[13] S. J. Leemans, D. Fahland, W. M. van der Aalst, Discovering block-structured process models from event logs containing infrequent behaviour, in: International conference on business process management, Springer, 2013, pp. 66–78..
[14] L. Ghionna, G. Greco, A. Guzzo, L. Pontieri, Outlier detection techniques for process mining applications, in: International symposium on methodologies for intelligent systems, Springer, 2008, pp. 150–159.
[15] L. Genga, M. Alizadeh, D. Potena, C. Diamantini, N. Zannone, Discovering anomalous frequent patterns from partially ordered event logs, J. Intell. Inform. Syst. 51 (2) (2018) 257–300.
[16] X. Lu, D. Fahland, F. J. van den Biggelaar, W. M. van der Aalst, Detecting deviating behaviors without models, in: International Conference on Business Process Management, Springer, 2016, pp. 126–139..
[17] D.C. Hoaglin, R.E. Welsch, The hat matrix in regression and anova, Am. Stat. 32 (1) (1978) 17–22.
[18] M.A. Pimentel, D.A. Clifton, L. Clifton, L. Tarassenko, A review of novelty detection, Signal Processing 99 (2014) 215–249.
[19] S. J. Leemans, D. Fahland, W. M. van der Aalst, Discovering block-structured process models from event logs-a constructive approach, in: International conference on applications and theory of Petri nets and concurrency, Springer, 2013, pp. 311–329..
[20] C. Warrender, S. Forrest, B. Pearlmutter, Detecting intrusions using system calls: Alternative data models, in: Proceedings of the 1999 IEEE symposium on security and privacy (Cat. No. 99CB36344), IEEE, 1999, pp. 133–145..
[21] K. Böhmer, S. Rinderle-Ma, Multi-perspective anomaly detection in business process execution events, in: OTM Confederated International Conferences On the Move to Meaningful Internet Systems, Springer, 2016, pp. 80–98.
[22] C.O. Back, S. Debois, T. Slaats, Entropy as a measure of log variability, J. Data Semantics 8 (2) (2019) 129–156.
[23] B. Everitt, The cambridge dictionary of statistics cambridge University Press, UK Google Scholar, Cambridge, 1998.
[24] S. Chatterjee, A.S. Hadi, et al, Influential observations, high leverage points, and outliers in linear regression, Statistical science 1 (3) (1986) 379–393.
[25] S. Wold, K. Esbensen, P. Geladi, Principal component analysis, Chemometrics Intelligent Lab. Syst. 2 (1–3) (1987) 37–52.
[26] P. Drineas, M.W. Mahoney, S. Muthukrishnan, Relative-error cur matrix decompositions, SIAM J. Matrix Anal. Appl. 30 (2) (2008) 844–881.
[27] M.W. Mahoney, P. Drineas, Cur matrix decompositions for improved data analysis, Proc. Nat. Acad. Sci. 106 (3) (2009) 697–702.
[28] D. Papailiopoulos, A. Kyrillidis, C. Boutsidis, Provable deterministic leverage score sampling, in: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2014, pp. 997–1006..
[29] M. Klimstra, E.P. Zehr, A sigmoid function is the best fit for the ascending limb of the hoffmann reflex recruitment curve, Exp. Brain Res. 186 (1) (2008) 93–105.
[30] Z. Liu et al, A method of svm with normalization in intrusion detection, Procedia Environ. Sci. 11 (2011) 256–262.
[31] E. Esgin, P. Karagoz, Confidence-aware sequence alignment for process diagnostics, in: 2013 International Conference on Signal-Image Technology & Internet-Based Systems, IEEE, 2013, pp. 990–997..
[32] I. Ben-Gal, Outlier detection, in: Data mining and knowledge discovery handbook, Springer, 2005, pp. 131–146.
[33] N. Kumar, S. Lalitha, Testing for upper outliers in gamma sample, Commun. Stat.-Theory Methods 41 (5) (2012) 820–828.
[34] A. Burattin, Plg2: Multiperspective process randomization with online and offline simulations., in: BPM (Demos), 2016, pp. 1–6..
[35] L. Genga, M. Alizadeh, D. Potena, C. Diamantini, N. Zannone, Discovering anomalous frequent patterns from partially ordered event logs, J. Intell. Inform. Syst. 51 (2) (2018) 257–300.
[36] D. Devi, S.K. Biswas, B. Purkayastha, Learning in presence of class imbalance and class overlapping by using one-class svm and undersampling technique, Connection Sci. 31 (2) (2019) 105–142.
[37] J. Hua, Z. Xiong, J. Lowey, E. Suh, E.R. Dougherty, Optimal number of features as a function of sample size for various classification rules, Bioinformatics 21 (8) (2005) 1509–1515.
[38] M. Comuzzi, J. Ko, S. Lee, Predicting outpatient process flows to minimise the cost of handling returning patients: A case study, in: International Conference on Business Process Management, Springer, 2019, pp. 557–569..