

Automatic and Explainable Labeling of Medical Event Logs with Autoencoding

Hugo de Oliveira, Vincent Augusto, Baptiste Jouaneton, Ludovic Lamarsalle, Martin Prodel, Xiaolan Xie

▶ **To cite this version:**

Hugo de Oliveira, Vincent Augusto, Baptiste Jouaneton, Ludovic Lamarsalle, Martin Prodel, et al.. Automatic and Explainable Labeling of Medical Event Logs with Autoencoding. IEEE Journal of Biomedical and Health Informatics, 2020. emse-03128429

HAL Id: emse-03128429

https://hal-emse.ccsd.cnrs.fr/emse-03128429

Submitted on 2 Feb 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Automatic and Explainable Labeling of Medical Event Logs with Autoencoding

Hugo De Oliveira, Vincent Augusto, Baptiste Jouaneton, Ludovic Lamarsalle, Martin Prodel and Xiaolan Xie

***Abstract*—Process mining is a suitable method for knowledge extraction from patient pathways. Structured in event logs, medical events are complex, often described using various medical codes. An efficient labeling of these events before applying process mining analysis is challenging. This paper presents an innovative methodology to handle the complexity of events in medical event logs. Based on autoencoding, accurate labels are created by clustering similar events in latent space. Moreover, the explanation of created labels is provided by the decoding of its corresponding events. Tested on synthetic events, the method is able to find hidden clusters on sparse binary data, as well as accurately explain created labels. A case study on real healthcare data is performed. Results confirm the suitability of the method to extract knowledge from complex event logs representing patient pathways.**

***Index Terms*—process mining; event log; healthcare data; patient pathways; autoencoding;**

I. Introduction

Data analytic regroups an extensive number of methods to investigate data produced in various systems such as industry, software engineering and healthcare. Knowledge extraction from such data is a lever to improve performances, to predict or simply to describe the reality of facts. Among different types of data, event logs are challenging to analyze because of the presence of time, the high variability of events, and the complex relations between events. Thus, the use of widespread data mining algorithm may not be fully straightforward for some applications. A wise preprocessing step to capture meaningful information may be necessary. Describing processes, these data are present in the manufacture industry, in software engineering and in healthcare [1]. To analyze event logs, a data-driven approach named *process mining* has been proposed [2]. Between data mining and process modeling, event logs are impartially used to extra [3].

The French national health insurance database (SNIIRAM) is a non-clinical claim database. Containing healthcare reimbursements of almost all French citizens, the amount of data is colossal. 66 million inhabitants were part of this database in 2015 [4]. Among all reimbursement information contained in the SNIIRAM, patients' hospitalizations are provided. However, no precise medical information such as test results, imaging reports, or vital signs are available. Nevertheless, such a database is useful to map patient pathways [5]–[8], perform medical data clustering [9] and prediction tasks [10], [11]. Regarding healthcare processes, the complexity is multifold. Illustrations are, but not limited to, the presence of free text, the granularity of events analyzed, and the occurrence of multiple event simultaneously, leading to multiple codes describing a given event. These codes, representing medical activities of different types, could be numerous and often inherit of hierarchical structures [6]. Even if the hierarchy could be useful to simplify codes and reduce the overall cardinality, the choice of the accurate level in order to produce meaningful events is not obvious, depends on the pathology or the health process studied and often requires a clinical expertise. This aspect of complexity is one of the main challenge regarding non-clinical claim database, such as the SNIIRAM.

Therefore, the main contribution of the present paper is a new methodology to analyze the complexity of events and produce meaningful labels. Using autoencoding and clustering, the proposed method creates artificial labels from initial data. These labels are assigned to events, transforming the raw event log by reducing the overall variability of events. The method provides transparency for practitioners by giving an interpretation for each created artificial label. In practice, the contribution consists of a preprocessing methodology to treat this particular complexity of events. As a result, available process mining tools[1] can be used starting from event logs obtained via the proposed methodology.

This paper is organized as follows. An overview of related works is given in Section II. Preliminary notations are presented in Section III. Section IV introduces the problem addressed in this paper. The proposed methodology is described in Section V. To validate the method, a design of experiments is presented in Section VI, followed by a case study based on real-life healthcare data in Section VII. Finally, conclusions and perspectives are given in Section VIII.

II. Literature review

Healthcare data analysis constitutes a large field to test and apply a wide spectrum of analytic methods. Among them, machine learning and more recently deep learning methods have been largely deployed. Electronic Health Records (EHR) have permitted the development of new models and methods, boosting the field publication activity [12], [13]. Among the tasks addressed by deep learning, supervised learning and concept embedding emerge for a majority of studies in

H. De Oliveira, V. Augusto and X. Xie are with Mines Saint-Étienne, Univ Clermont Auvergne, CNRS, UMR 6158 LIMOS, Centre CIS, F - 42023 Saint-Étienne France (e-mails: hdeoliveira@hevaweb.com; augusto@emse.fr; xie@emse.fr).

H. De Oliveira, B. Jouaneton, L. Lamarsalle and M. Prodel are with HEVA, 186 avenue Thiers, F-69465, Lyon, France (e-mails: bjouaneton@hevaweb.com; llamarsalle@hevaweb.com; mprodel@hevaweb.com).

X. Xie is also with the Antai College of Economics and Management, Shanghai Jiao Tong University, China.

[1]Such as `ProM`, `Disco`, `PM4Py` or `bupaR`.

healthcare [14]. In spite of the need to develop high performing predictive models, explainability has been highlighted as a key issue for future model developments [14], [15]. Valuable for practitioners and experts of the medical field, the explanation of predictive results has already been addressed by deep learning studies [16], [17].

In addition to supervised learning and concept embedding, process mining is also a promising analytic method. By performing process discovery on event log data, process mining algorithms produce graphical and interpretative representations of occurring processes. The field is active and recent publications are numerous [1], particularly in healthcare [18]. Initiatives of the research community to improve practices and knowledge sharing illustrate the field activity.[2] Recent topics addressed are, but not limited to, privacy [19], clinical guideline [20], and data cleaning [21]. Regarding process discovery, an optimal procedure to construct process models from raw data bases has been proposed by Prodel et. al. in 2018 [6]. Applied to healthcare data event logs, the proposed method has been adapted to take into account temporal information during the optimization process [8].

Patients' pathways analysis based on real-life healthcare data is valuable to represent and understand how patients' care occurs in real-life. Even if deep representation has been applied on clinical pathways [22], process mining use graphical process models as a support for representation. Thus, it makes the method suitable to discover patients' pathways from raw data when the focus lies on interpretation. However, medical data can be complex to analyze, due to the variety of different medical codes used in claims databases (e.g. diagnoses, procedures and drugs). As a result, the labeling of events is a challenging step in data processing. A commune practice is the definition of labels by hand, based on expert knowledge [8]. The detection of hidden healthcare sub-processes has been proposed, using Hidden Markov Models (HMM) [23]. The presented method allows a reduction of complexity by the enrichment of the log with HMM-derived states, reducing complexity and saving experts time. A practical solution proposed by Prodel et al. in 2018 [6] is the creation of labels during the optimization, using the hierarchy of events from one type of codes, such as main diagnosis. However, when multiple codes from multiple coding systems characterize the events of a process, the selection of the right aggregation level and the combination of codes is not, to the best of our knowledge, a treated problem.

Therefore, the main contribution of this paper is the proposal of a general methodology to treat complex events such as multiple medical activities, in order to apply process mining. Following the definition of Lenz and Reichert [24], the proposed method identifies the activities of the *medical treatment process* by analyzing the coded events of the *organizational processes*. The core of this methodology is based on recent work in representation learning. A widely used method in representation learning is *autoencoding* [25]. The general idea behind autoencoding is the learning of a structure which can encode and decode information while minimizing the loss of information. By compressing the data, a transformation of the input representation is performed.

Thus, before formally introducing the problem and the proposed methodology, preliminary notations are presented in the following.

III. Preliminaries

A formal description of the data involved is provided in the following, including events, traces and event logs.

Definition 1: Event. Each event denoted e is a couple (c, t, a) where:

- c is the related *case ID* of the event, with the id function $id(e) = c$ returning the case ID of event e;
- $t \in T$ with $T = \mathbb{N}$ or $\mathbb{R}$ corresponds to the event time also called *time-stamp*, with the event time function $time(e) = t$ returning the time-stamp of event e;
- a is a nonempty set called *activity set*, each element $a_i \in a$ being an *activity*.

Definition 2: Trace. A trace is a sequence of events denoted as $\sigma = e_1, e_2, \ldots, e_m$ with $m \in \mathbb{N}^*$ such that $time(e_k) < time(e_{k+1})$ and $\forall e, e' \in \sigma$, $id(e) = id(e')$. The *size* of the trace $|\sigma|$ is defined as the number of events in σ.

Definition 3: Event log. An event log is a set of traces denoted as $L = \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$ with $n \in \mathbb{N}^*$. The *size* of the event log $|L|$ is defined as the number of traces in L. Its *length* $len(L)$, defined as $len(L) = \sum_{\sigma \in L} |\sigma|$ gives the total number of events composing L.

According to the previous definitions, an event log is a group of traces, each trace being a succession of events characterized by activities occurring at a given time-stamp, composing an activity set.

Definition 4: Log activity set. Let L be an event log. The *log activity set* of L is the nonempty set $\mathcal{A}$ defined as:

- $\mathcal{A} = \{a_i | \exists \sigma \in L, \exists e = (c, t, a) \in \sigma, a_i \in a\}$
- $\forall \sigma \in L, \forall e = (c, t, a) \in \sigma, \forall a_i \in a, \exists! \mathcal{A}_j \subset \mathcal{A} | a_i \in \mathcal{A}_j$.

Thus, the log activity set is composed of activities encountered in L, with every single activity of event log L belonging to a unique subset $\mathcal{A}_j$ of $\mathcal{A}$.

Proposition 1:

$$\mathcal{A} = \bigcup_j \mathcal{A}_j \text{ and } \bigcap_j \mathcal{A}_j = \emptyset \tag{1}$$

Proposition 2:

$$card(\mathcal{A}) = \sum_j card(\mathcal{A}_j) \tag{2}$$

Definition 5: Activity vector. Let $e = (c, t, a)$ be an event, the *activity vector* x of e is defined such that $x \in E_X = \{0, 1\}^d$, E_X being the *activity vector space*, with $d = card(\mathcal{A})$. A mapping function $vect()$ is also introduced, defined as:

$$vect : \mathcal{A} \to E_X$$
$$a \mapsto vect(a) = x$$

with its inverse $vect^{-1} : E_X \to \mathcal{A}$.

The previously defined functions allow a mapping between an activity set and its corresponding activity vector. The

[2] http://pods4h.com/

case id	time-stamp	medical activities
0	0	{Z511; ZZNL053}
0	10	{Z511}
0	20	{Z5101; 9261771}
1	0	{Z511; 9261110}
1	5	{Z511; 9261110}
1	20	{Z511; 9261110}
1	50	{Z511; 9261771; ZZNL053}

(a) Activity set representation.

case id	time-stamp	Z511	Z5101	ZZNL053	9261771	9261110
0	0	1	0	1	0	0
0	10	1	0	0	0	0
0	20	0	1	0	1	0
1	0	1	0	0	0	1
1	5	1	0	0	0	1
1	20	1	0	0	0	1
1	50	1	0	1	1	0

(b) Activity vector representation.

TABLE I: Example of an event log of patient pathways.

activity vector x of an event is the equivalent representation of an activity set using 1-of-k coding from all possible activities from $\mathcal{A}$ to the activity vector space E_X. Therefore, a given event e could be defined as $e = (c, t, a)$ or $e = (c, t, x)$ knowing $vect$ and $vect^{-1}$ without any meaning loss regarding the log activity set.

Definition 6: Activity matrix. The *activity matrix* of an event log L is defined as $M_X = (x_i)_{i \in [\![1, len(L)]\!]}$ with $dim(M_X) = len(L) \times d$.

The activity matrix M_X of an event log L gives a binary representation of activity sets, which is a common representation in machine learning.

Example 1: Let us define an event log $L = \{\sigma_1\}$ with one trace $\sigma_1 = e_1, e_2$ having two events such that $e_1 = (c_1, t_1, a_1)$ and $e_2 = (c_2, t_2, a_2)$ with:

- $\mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2$ with $\mathcal{A}_1 = \{A_1, B_1\}$ and $\mathcal{A}_2 = \{A_2, B_2, C_2\}$;
- case ids $c_1 = c_2 = c$;
- time-stamps $t_1 = 0$, $t_2 = 10$;
- activities $a_1 = \{A_1, B_2\}$, $a_2 = \{A_1, B_1, A_2, C_2\}$.

According to the previous definitions:

- $d = card(\mathcal{A}_1) + card(\mathcal{A}_2) = 5$;
- $x_1 = (1, 0, 0, 1, 0)$ and $x_2 = (1, 1, 1, 0, 1)$.

Example 2:

Table I presents a short event log related to patient pathways analysis using data as found in claims database. Each row is a hospitalization event. Events with the same case ID are ordered by increasing time stamp and represent a trace, which is a patient's hospitalization history. Medical activities are of three different categories: diagnosis, medical procedures and drugs, coded using standard notations ICD-10 (International Classification of Diseases 10^{th} revision), CCAM (*Classification Commune des Actes Médicaux*) and UCD (*Unité Commune de Dispensation*), respectively. According to previous definitions, $\mathcal{A} = \mathcal{A}_{diag} \cup \mathcal{A}_{med.proc.} \cup \mathcal{A}_{drugs}$, with $\mathcal{A}_{diag} = \{\text{Z511}; \text{Z5101}\}$, $\mathcal{A}_{med.proc.} = \{\text{ZZNL053}\}$ and $\mathcal{A}_{drugs} = \{261771; 9261110\}$. Moreover, $d = card(\mathcal{A}_{diag}) + card(\mathcal{A}_{med.proc.}) + card(\mathcal{A}_{drugs}) = 5$. Representation using activity set and its equivalent using activity vector are presented in Table Ia and Table Ib, respectively.

Thus, event log notations as well as activity vector and matrix formalization have been introduced. The following definitions introduce the problem setting.

IV. Problem definition

Definition 7: Label function and set. Given an event log L, a *label function* λ is a function such that:

$$\lambda : \mathcal{A} \to \mathcal{L} \tag{3}$$

with $\mathcal{L}$ being the *label set*.

The label function maps activities of an event log L to $\mathcal{L}$, a set of possible labels for each activity vector.

Definition 8: Explaining function. Given an event log L and its label set $\mathcal{L}$, an *explaining function* η on L is a function:

$$\eta : \mathcal{L} \to \mathcal{A} \tag{4}$$

The explaining function allows a mapping of a label l to interpretative elements a from the log activity set : $a \subset \mathcal{A}$.

Definition 9: Activity clustering problem. Lets L be an event log with its log activity set $\mathcal{A}$ and its activity matrix M_X, with $dim(M_X) = len(L) \times d$. The *activity clustering problem* on L is defined as the search of the triple $(\mathcal{L}, \lambda, \eta)$ such that:

- $\mathcal{L}$ is a label set, $\mathcal{L} \neq \emptyset$;
- $K_{min} \leq card(\mathcal{L}) \leq K_{max}$, with $K_{min}, K_{max} \in \mathbb{N}^*$;
- $K_{max} \ll d$;
- λ is a label function;
- η is an explaining function.

The main objective here is to find an accurate triple $(\mathcal{L}, \lambda, \eta)$ for the considered event log L. This problem can be seen as a clustering problem, with $\mathcal{L}$ being the set of cluster labels and λ the clustering function. The particularity here is for the input data M_X to be sparse and of high dimensionality d in terms of features. The label set $\mathcal{L}$ should be finite, its cardinality (number of elements) being reasonable. A high cardinality of $\mathcal{L}$ will induce difficulties in process mining readability, also encouraging overfitting regarding $\mathcal{A}$. This consideration leads to the proposed upper bound condition $K_{max} \ll d$. But a small cardinality for the label set will lead to a lack of information and medical meaning, motivating the lower bound K_{min}. Moreover, as explainability is an essential constraint when dealing with medical pathways analysis, elements of $\mathcal{L}$ should keep a medical meaning, justifying the search of the explain function η. Given these considerations, it is assumed in this paper that there exist some relevant clusters hidden in medical activities of event log L. This assumption suggests that some combinations of elements of $\mathcal{A}$ which could characterize well a sufficient number of events e could be found.

In the following, a methodology based on autoencoding is proposed to find a relevant label set and a label function from a given event log.

V. Proposed methodology

A. Overview

To solve the activity clustering problem for an event log L and find an accurate triple $(\mathcal{L}, \lambda, \eta)$, an autoencoding method is proposed. The idea is to transform space data

from E_X to a *latent space* E_Z of reduced dimensionality, where similar elements are close to each other. This transformation is done using an *autoencoding* architecture, with an *encoder* $f : E_X \rightarrow E_Z$ and a decoder $g : E_Z \rightarrow E_X$. In the latent space, because the dimensionality is reduced, applying clustering methods is meaningful. *Latent clustering* is proposed to assign for each element z a label through a function $h : E_Z \rightarrow \mathcal{L}$. Thus, cluster labels defined through clustering in latent space will constitute the label set $\mathcal{L}$ defined in Definition 7:

$$\lambda : \mathcal{A} \xrightarrow{vect(a)} E_X \xrightarrow{f(x)} E_Z \xrightarrow{h(z)} \mathcal{L} \tag{5}$$

To construct the explaining function η, the function $h^{-1} : \mathcal{L} \rightarrow E_Z$ first returns for each cluster label l, the set of corresponding vectors in latent space Z_l. Then, each activity $z \in Z_l$ can be decoded from E_Z to E_X using the decoding function g learned during the autoencoder training phase. This results in a set of activity vectors X_l, from which an average vector $\bar{X}_l$ is computed. Finally, $\bar{X}_l$ is interpreted as a set of activities using $vect^{-1}$ function:

$$\eta : \mathcal{L} \xrightarrow{h^{-1}(l)} E_Z \xrightarrow{g(Z_l)} E_X \xrightarrow{vect^{-1}(\bar{X}_l)} \mathcal{A} \tag{6}$$

Consequently, the proposed method is composed of three steps: (1) autoencoder training, (2) latent space clustering, and (3) clusters' related activities decoding. These steps are presented in more detail in the following sections.

B. Autoencoder training

To perform clustering on sparse, binary, high-dimensional activity vectors, the data is transformed into a new space where variables are continuous and the dimensionality is lower. This transformation is performed using an autoencoder, trained on an activity matrix. In this paper, three methods for autoencoder training are investigated.

1) Autoencoder: An autoencoder (AE) is composed of two functions f and g, named the *encoding* and *decoding* functions, respectively. The encoder transforms a vector x from the input space into a new vector z from the *latent space*, its dimensionality being drastically reduced. The decoder takes the vector z from the latent space and decode it back to the input space, resulting in a new vector x'. The training of a classic autoencoder is done by minimizing the reconstruction error, usually the binary cross entropy loss function. The dimensionality reduction allows a concentration of information while keeping the useful information in latent space for the reconstruction of input data. In this paper, encoder and decoder are constructed using symmetric feed-forward, fully connected neural networks.

2) Denoising autoencoder: A denoising autoencoder (DAE) is constructed using the same architecture as for an AE. A noisy vector $\tilde{x}$ is created from x, which is encoded and then decoded. The loss function remains unchanged, while the goal of training is to be robust against artificially added noise and to keep useful information to decode data without noise.

3) Variational autoencoder: A variational autoencoder (VAE) is a particular autoencoder where the learned variables are parameters of a distribution. The encoder f is an *inference network* $q(x|z)$ and the decoder g a *generative network* $p(z|x)$. The *reparameterization trick* makes the training of the network possible using gradient descent optimization. The loss function here is the inverse of the expected lower bound $ELBO$ defined as $ELBO = \mathbb{E}_{q(z|x)}\left[\log \frac{p(x,z)}{q(z|x)}\right]$. In practice, the single sample estimate $\log p(x|z) + \log p(z) - \log q(z|x)$ with z sampled from the inference network is optimized [26].

C. Latent space clustering

Once an autoencoder is trained using activity matrix, a representation of every activity vector in latent space E_Z can be obtained. In E_Z, observations are characterized by a reasonable number of continuous features. As a result, applying clustering in this space is meaningful. K-means algorithm is used to learn the function $h : E_Z \rightarrow \mathcal{L}$ and create K clusters of similar observations in latent space. The parameter K corresponds to the final number of labels in $\mathcal{L}$: $K = card(\mathcal{L})$, respecting $K_{min} \leq K \leq K_{max}$ (Definition 9). To find such an accurate value of K, one possible criterion could be to maximize the *mean silhouette score* [27], defined as:

$$\mathcal{S} = \frac{1}{len(L)} \sum_{z \in M_Z} \frac{b_z - a_z}{max(a_z, b_z)} \tag{7}$$

with

- a_z : the mean distance between z and all other points of the same cluster;
- b_z : the mean distance between z and all other points of the nearest next cluster.

D. Clusters' activities decoding

Once the latent space clustering has constructed a function $h : E_Z \rightarrow \mathcal{L}$, the label function $\lambda : \mathcal{A} \rightarrow \mathcal{L}$ is fully defined. A label is assigned to each activity set of event log L, which can be used in the final process model as a node label. To construct the explaining function $\eta : \mathcal{L} \rightarrow \mathcal{A}$, a methodology based on the decoding of each cluster's activities is proposed. The hypothesis formulated here is that for each cluster defined in latent space E_Z, averaging the output of its activities decoding gives an overview of the cluster's meaning. In practice, the $vect^{-1}$ function needs a threshold to convert an activity vector into decoded activities. The analysis of activity vectors can lead to a judicious choice for the threshold, as presented in Remark 1, Section VI.

To summarize this section, the presented methodology allows for the analysis of activity sets from event logs and defines a set of labels that can be used for process mining. Each event of the event log has a label defined through the label function, λ, and every label is interpreted by the explaining function η. A summary of preliminary notations is proposed in Table II.

name	notation	name	notation
activity	a_i	log activity set	$\mathcal{A}$
activity set	a	activity vector	x
time-stamp	t	activity matrix	M_X
event	e	activity vector space	E_X
trace	σ	encoder	f
event log	L	decoder	g
label set	$\mathcal{L}$	latent vector	z
label function	λ	latent matrix	M_Z
explaining function	η	latent space	E_Z

TABLE II: Notations summary.

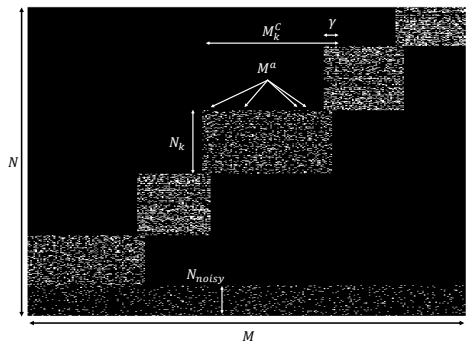


Fig. 1: Example of generated data. Here, $\kappa = 5$, $\alpha = 0.05$, $\beta = 2$ and $\gamma = 0.1$, leading to $\tilde{S} = 0.9$ with each white pixel being a value of 1 and black pixels being a value of 0.

VI. Design of experiments

In this section, a design of experiment on synthetic data is presented. The objectives of such an experiment are multiple: (1) to verify the accuracy of the methodology in identifying hidden patterns (clusters) in event logs; (2) to demonstrate the veracity of the decoder in explaining clusters' labels; (3) to compare performances of autoencoding methods with direct clustering on sparse data; and (4) to benchmark autoencoding methods with one another.

A. Data description

The input data used in this paper are event logs, composed of events for which accurate labels are searched, according to the problem defined in Definition 9. Thus, labels hidden in the data were represented using groups of activities, among which vectors will be generated. Such a construction was motivated by data experts observations regarding non-clinical claims data: similar hospitalizations are often described by codes from a same "set", which can be approximated using clinicians knowledge. As an example, a stay for a given operation can be fill in the database using one code representing the operation, another for medical imaging procedures, some codes related to particular diagnosis related to complications of the operation or to the medical condition of the patient. Thus, there exist similarities between two stays for the same operation, because some codes are issued from a same "set", but they are rarely identical.

Considering these remarks, synthetic data were generated representing an activity matrix as defined in Definition 6, where each row represented an activity vector and each column a 1-of-k representation of activities. The number of different labels hidden in the data (i.e. the number of clusters to find) is referred to as κ. For each label $k \in [\![1, \kappa]\!]$, N_k vectors were generated such that $N_k = \mathcal{N}(\mu_N, \frac{\mu_N}{5})$, $\mu_N = \frac{2500}{\kappa}$. The number of *characteristic activities* for each label k is referred to as $M_k^C \in \mathbb{N}^*$, constructed such that $M_k^C = \mathcal{N}(\mu_C, \frac{\mu_C}{5})$ with $\mu_C = \alpha \times \beta \times \mu_N$. The number of all different activities involved is referred to as M. For each activity vector of a given label k, a number $M^a = \alpha \times \mu_N \in \mathbb{N}^*$ of activities was randomly chosen to construct it. The number M^a corresponds to the number of activities randomly chosen among characteristic activities. For these activities, the corresponding attribute value was set to 1, keeping 0 otherwise. An overlapping ratio γ is also introduced, representing the quantity of activities of a label shared with the nearest one. Moreover, a number $N_{noisy} = 250$ of noisy events was generated. These noisy events were composed of activities among all possible activities in the constructed data set, with no particular pattern related to a given hidden label k. As a result, the total number of traces (i.e. the number of rows) is $N = N_{noisy} + \sum_{k=1}^{\kappa} N_k$. The parameters that have been chosen in the design of experiments lead to an approximated sparsity $\tilde{S}$ between 0.9 and 0.99. An example of synthetic data generated is presented in Figure 1.

B. Experiment description

The autoencoder methods AE, DAE, and VAE, presented in Section V-B, were implemented and compared in terms of performances. Neural networks used are feed-forward networks, composed of four fully connected layers of size $10 \times d_{latent}$, $5 \times d_{latent}$ and d_{latent}, the latter being the dimensionality of the latent space defined as $d_{latent} = 8$. For DAE, noise was defined as randomly selecting elements in vectors and changing their values (from 0 to 1 or 0 to 1). 1% of noise is added in every vector. For VAE, d_{latent} couple of parameters of Gaussian distributions were learned as latent variables. The inverse single sample Monte Carlo estimate of the *ELBO* was used as the loss function to minimize during training. For each parameter combination of the data, autoencoder training was done using a symmetric architecture between the encoder and the decoder. Dropout and L-2 regularization were used for each layer to prevent the training from overfitting. The chosen optimizer for training was *Adam*, with mini-batch of size 32. The total number of epochs was set to 1000. Of the overall data, 80% was used for training, while 20% was kept to evaluate validation error for early stopping (with a patience of 25 iterations). After autoencoder training, K-mean clustering in latent space was applied using all training and validation data, K being fixed by maximizing mean silhouette score for $K \in [\![K_{min}, K_{max}]\!]$ with $K_{min} = 2$ and $K_{max} = 15$.

Performances were evaluated regarding clustering on the one hand, and explainability on the second hand. To evaluate clustering performances, an automatic procedure constructed a confusion matrix between hidden labels and found clusters, maximizing the accuracy (the sum of the diagonal) by permuting columns only (found clusters) to align proposed clusters with potential corresponding ones in hidden ones. The

accuracy of resulting confusion matrix was then computed, defined as the ability of a method to accurately assign the right label to each event. To evaluate the ability of the method to explain found clusters, the *explaining F-score* F_η is introduced. Let $k \in [\![1, \kappa]\!] = K^{true}$ be the label of the cluster hidden in data, and $c : k \mapsto c(k) \in K^{pred}$ being a function returning the corresponding cluster label according to previously described confusion matrix optimization. The average of decoded elements from the cluster $c(k)$ is computed, and its activity set $a^{pred}_{c(k)}$ is compared to characteristic activities of the corresponding label a^{true}_k. To quantitatively analyze the decoding performances, the *explaining F-score* is defined such as: $F_\eta = \frac{2 \times R_\eta \times P_\eta}{R_\eta + P_\eta}$ with $R_\eta = \frac{\sum_{k \in K^{true}} a^{true}_k \bigcap a^{pred}_{c(k)}}{\sum_{k \in K^{true}} card(a^{true}_k)}$ the *explaining recall* and $P_\eta = \frac{\sum_{k \in K^{pred}} a^{true}_k \bigcap a^{pred}_{c(k)}}{\sum_{k \in K^{pred}} card(a^{pred}_{c(k)})}$ the *explaining precision*. These expressions are analogous to classical binary classification metrics. A high explaining recall means a high ability of the explaining function to find corresponding activities with hidden activities of the identified labels. Furthermore, a high explaining precision corresponds to a decoding that keeps interesting activities without being too general. Ideally, each discovered label corresponds to a hidden one. This is not necessarily the case, as the number of discovered and hidden clusters could be different. Fewer predicted than hidden clusters will impact the explaining recall, and more predicted than true clusters will impact explaining precision.

Remark 1: As mentioned in Section V-D, the threshold will have significant impact in the previously defined metrics. In the design of experiments conducted here, an automatic approach was used. For a cluster l, a list of all decoded values from the average decoding vector $\bar{X}_l$ was constructed. All elements of this list were ordered in descending order. By differentiation of this curve another list of values was obtained. The minimum value of the resulting curve was used to automatically define a judicious threshold for keeping activities in the explaining set of the related cluster.

For every combination of parameters, 10 data sets were constructed. Columns (activities) were shuffled, right before the shuffling of rows (events). The proposed method was applied using the previously described autoencoders (AE, DAE and VAE) as well as a direct K-mean clustering without autoencoding step, tested as a baseline (referred to as BASIC). Performances were analyzed through mean and standard deviation of clustering and explaining metrics. All algorithms and experiments were conducted using Python 3.7 and Tensorflow 1.14. A schematic description of experiments is presented in Figure 2.

C. Results

Results[3] are summarized in Table III. A total of 24 experiments of increasing difficulty were conducted. For each combination of parameters, the accuracy and the F_η score of the tested methods are presented. Results show that the autoencoding methods outperform the direct clustering in sparse high-dimensional space (except regarding experiments 15 and 21 where DAE performances are inferior). Therefore, autoencoding plays an important role in data transformation for the proposed methodology. Furthermore, the results highlight that VAE always outperforms the other methods regarding both accuracy and F_η. Event though the standard deviation increases for difficult experiments, VAE shows a lower variation compared to the other methods. As a conclusion, results motivate the choice of VAE as part of the proposed method to obtain accurate clusters and explain them.

[3]Detailed results regarding the design of experiments of Section VI, as well as the case study described in Section VII can be found as supplementary materials on the following website: https://artemis-emse-laparo.hevaweb.com/

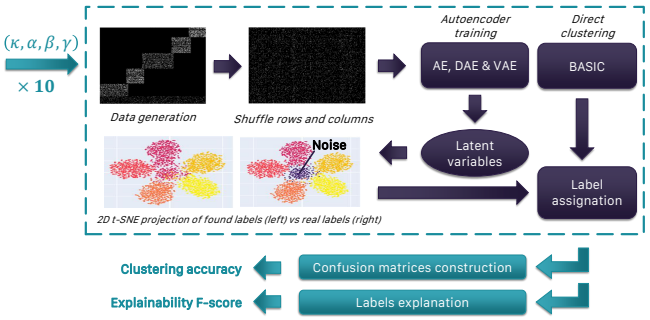


Fig. 2: Schematic representation of the design of experiments.

VII. Case Study

After proving the accuracy of the method on synthetic data, this section is dedicated to demonstrate the relevance of the method on real healthcare data. In that purpose, a medical case study is presented, were process mining was deployed to extract knowledge about patients' hospital pathways.

A. Overview

A laparotomy is an abdominal surgery consisting of a large incision of the abdomen, sometime necessary to investigate abdominal pain. Incisional hernia (IH) is one of the possible complications following laparotomy. These complication consists in a protrusion of the tissues of the abdomen through the abdominal muscle. The repair on an IH is a common surgery, which can lead to chronic pain and decreased quality of life. Colorectal surgeries, bariatric surgeries and abdominal aortic aneurysm are laparotomy surgeries that may lead to IH [7]. In this case study, we focused on patients developing an IH after a laparotomy operation. The objective was to apply the previously defined methodology to label raw medical event logs before applying process mining. A process mining study using manual labeling was also performed, to illustrate the relevance of the methodology to automatically define interesting labels.

B. Methods

The data[4] were extracted from the SNIIRAM database. All anonimized patients with a first laparotomy operation in 2010,

[4]Access to an extraction of the SNIIRAM database was provided by the French CNIL under the agreement number DR-2019-147.

	Exp.				ACCURACY								F_η							
					BASIC		AE		DAE		VAE		BASIC		AE		DAE		VAE	
	κ	α	β	γ	AVG	STD	AVG	STD	AVG	STD	AVG	STD	AVG	STD	AVG	STD	AVG	STD	AVG	STD
1	5	0.05	2	0.00	0.55	0.06	0.66	0.16	0.71	0.05	**0.90**	0.01	0.71	0.13	0.80	0.27	0.95	0.11	**1.00**	0.00
2	5	0.05	2	0.10	0.55	0.09	0.66	0.16	0.71	0.08	**0.89**	0.01	0.68	0.17	0.82	0.32	0.95	0.14	**1.00**	0.00
3	5	0.05	2	0.25	0.56	0.07	0.68	0.16	0.72	0.04	**0.89**	0.03	0.71	0.17	0.82	0.25	0.92	0.12	**0.97**	0.09
4	5	0.05	5	0.00	0.38	0.06	0.59	0.16	0.61	0.18	**0.85**	0.03	0.37	0.14	0.73	0.27	0.77	0.31	**1.00**	0.00
5	5	0.05	5	0.10	0.37	0.04	0.68	0.14	0.57	0.18	**0.86**	0.03	0.37	0.11	0.83	0.26	0.67	0.37	**1.00**	0.00
6	5	0.05	5	0.25	0.43	0.06	0.59	0.19	0.63	0.15	**0.85**	0.01	0.47	0.13	0.67	0.35	0.80	0.29	**1.00**	0.00
7	5	0.10	2	0.00	0.51	0.07	0.69	0.13	0.70	0.07	**0.88**	0.01	0.65	0.13	0.86	0.23	0.91	0.15	**1.00**	0.00
8	5	0.10	2	0.10	0.52	0.07	0.63	0.16	0.70	0.08	**0.89**	0.01	0.63	0.15	0.80	0.27	0.93	0.16	**1.00**	0.00
9	5	0.10	2	0.25	0.54	0.09	0.68	0.16	0.70	0.12	**0.88**	0.01	0.69	0.13	0.84	0.25	0.90	0.20	**0.94**	0.12
10	5	0.10	5	0.00	0.39	0.07	0.57	0.16	0.64	0.15	**0.82**	0.02	0.40	0.08	0.68	0.26	0.77	0.25	**1.00**	0.00
11	5	0.10	5	0.10	0.36	0.05	0.60	0.18	0.69	0.13	**0.83**	0.01	0.36	0.07	0.72	0.25	0.90	0.25	**0.97**	0.10
12	5	0.10	5	0.25	0.45	0.09	0.64	0.17	0.63	0.17	**0.84**	0.02	0.41	0.16	0.79	0.27	0.80	0.28	**1.00**	0.00
13	10	0.05	2	0.00	0.50	0.06	0.61	0.18	0.62	0.15	**0.87**	0.04	0.62	0.10	0.72	0.29	0.84	0.24	**0.98**	0.06
14	10	0.05	2	0.10	0.54	0.06	0.62	0.22	0.66	0.14	**0.88**	0.02	0.71	0.09	0.73	0.37	0.88	0.25	**0.97**	0.10
15	10	0.05	2	0.25	0.53	0.14	0.61	0.22	0.38	0.23	**0.86**	0.02	0.72	0.20	0.76	0.35	0.44	0.37	**0.94**	0.11
16	10	0.05	5	0.00	0.22	0.05	0.45	0.24	0.70	0.04	**0.82**	0.04	0.22	0.10	0.48	0.37	0.93	0.09	**0.95**	0.06
17	10	0.05	5	0.10	0.23	0.07	0.39	0.21	0.52	0.23	**0.83**	0.02	0.24	0.16	0.36	0.33	0.61	0.35	**0.96**	0.05
18	10	0.05	5	0.25	0.22	0.04	0.31	0.20	0.42	0.23	**0.83**	0.02	0.17	0.08	0.24	0.30	0.41	0.38	**0.93**	0.05
19	10	0.10	2	0.00	0.46	0.08	0.58	0.19	0.66	0.04	**0.86**	0.03	0.58	0.11	0.65	0.26	0.89	0.13	**1.00**	0.00
20	10	0.10	2	0.10	0.50	0.05	0.58	0.17	0.61	0.17	**0.86**	0.02	0.72	0.08	0.75	0.30	0.81	0.27	**0.97**	0.10
21	10	0.10	2	0.25	0.51	0.11	0.53	0.22	0.44	0.22	**0.86**	0.02	0.69	0.21	0.68	0.36	0.52	0.38	**0.94**	0.12
22	10	0.10	5	0.00	0.20	0.02	0.32	0.19	0.53	0.19	**0.82**	0.02	0.16	0.05	0.31	0.27	0.65	0.33	**0.95**	0.05
23	10	0.10	5	0.10	0.21	0.03	0.32	0.18	0.49	0.24	**0.82**	0.02	0.17	0.07	0.30	0.24	0.61	0.41	**0.92**	0.06
24	10	0.10	5	0.25	0.24	0.06	0.33	0.17	0.50	0.20	**0.80**	0.04	0.21	0.10	0.28	0.31	0.56	0.31	**0.89**	0.09

TABLE III: Clustering and explainability performance, measured by accuracy and F_η, respectively. For all parameters combinations, average and standard deviation over 10 replications are presented. Best values are highlighted in bold.

followed by an IH within 5 years after the operation were selected. This resulted in a total number of $7,906$ patients included in the study, for which all hospitalization information was extracted. Each patient's hospitalization was transformed into a trace of his ordered medical activities. Thus, the activity set was structured as follows:

$$\mathcal{A} = \mathcal{A}_{\text{MD}} \bigcup \mathcal{A}_{\text{AD}} \bigcup \mathcal{A}_{\text{MP}} \bigcup \mathcal{A}_{\text{D}} \bigcup \mathcal{A}_{\text{TAD}} \tag{8}$$

where:

- $\mathcal{A}_{\text{MD}}$ is the set of main diagnoses, the reasons of the hospitalization (MD, using ICD-10 coding system);
- $\mathcal{A}_{\text{AD}}$ is the set of additional diagnoses (AD, using ICD-10 coding system);
- $\mathcal{A}_{\text{MP}}$ is the set of medical procedures (MP, using French CCAM coding system);
- $\mathcal{A}_{\text{D}}$ is the set of delivered drugs (D, using French UCD coding system linked to ATC - Anatomical Therapeutic Chemical - classes);
- $\mathcal{A}_{\text{TAD}}$ is the set of drugs under temporary authorization for delivery in French hospitals (using French LPP - *Liste des Produits et Préstations* - coding system).

Moreover, for each activity code, hierarchical knowledge (codes of upper levels in the hierarchy) was added as part of the corresponding activity set. This procedure enables relations between activity codes of a same group during autoencoding. It also enriches the explainability of clusters, by providing hierarchical knowledge and setting the level of precision in coding depending on clusters, as shown in the following results.

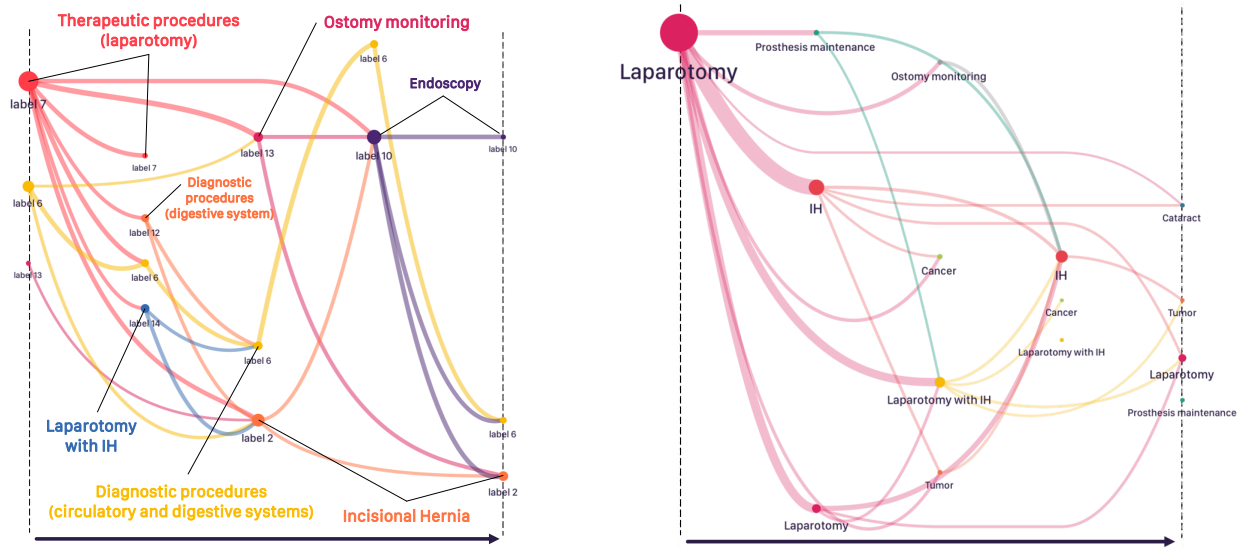
Stays related to dialyses or chemotherapy, which are known to appears very frequently, have been filtered. Codes appearing less than 50 times were also filtered, resulting in keeping 95.0% of codes while decreasing the size of the log activity set by 85.7%. The final event log constructed for the study contained, for 7906 traces (patients), 57533 events (stays) and 2228 unique activity codes. The previously defined methodology was conducted on the resulting activity matrix (of size 57533×2228), using VAE as autoencoder. The number of clusters used was $K = 15$, which qualitatively appears as a wise trade-off between explainability of clusters and final process model readability. The process mining framework used was the one proposed in [8], designed for application to medical event logs. The maximum number of nodes, edges and positions for process model optimization was fixed to 15, 25 and 5, respectively.

Cluster label	Activity label	Set (level)
2	Postoperative venting of the anterior abdominal wall	MP (lvl. 4)
	Ventral hernia	MD (lvl. 2)
6	Diagnostic acts on the circulatory system	MP (lvl. 2)
	Diagnostic acts on the digestive system	MP (lvl. 2)
7	Therapeutic acts on the digestive system	MP (lvl. 2)
10	Endoscopy of the alimentary canal	MP (lvl. 3)
12	Radiography of the digestive system	MP (lvl. 3)
13	Encounter for attention to artificial openings	MD (lvl. 2)
	Therapeutic acts on the colon	MP (lvl. 3)
14	Therapeutic acts on digestive system	MP (lvl. 2)
	Therapeutic acts on the anus	MP (lvl. 3)
	Therapeutic acts on the abdominal wall	MP (lvl. 3)

TABLE IV: Explanation of clusters appearing on process model: relevant decoded activities with corresponding activity set and level in the hierarchy.

C. Results

Results obtained by automatic labeling (Figure 3a) were compared with a process model of the same dimensionality, constructed with the same process mining procedure but start-



(a) Automatic labeling.

(b) Manual labeling.

Fig. 3: Comparison of process models obtained using automatic labeling (left) vs manual labeling (right).

ing from an event log with manually defined labels according to authors' prior knowledge about the pathology (Figure 3b). The process models read from left to right, the size of nodes and edges being, for each process model, proportional to the number of patients represented. Explanation results regarding clusters obtained and visualized in Figure 3a are detailed in Table IV. According to Figure 3, similarities are observed between the process models. Most frequent medical procedure codes related to laparotomy are subcategories of the "Therapeutic acts on the digestive system" code in the hierarchy. Thus, the related cluster (label 7) appears at the beginning of the pathways. Label 2, which contains codes related to IH, appears in the following of the pathway, as well as label 14 (combining codes related to laparotomy and IH). Also, stays related to diagnostic procedures (labels 6 and 12) and more precisely to an endoscopy of the alimentary canal (label 10) occupy a significant place in the process analyzed. These stays were not considered during manual labeling but were pointed out by the automatic labeling procedure based on raw data. They may be related to patient's medical control or investigation regarding suspicion of complication after operation. This example illustrates that other interesting information can be extracted from raw data with minimal initial input from the user. However, by comparing the replayability score (which gives a quantitative fitness measure of the resulting process models), a gap was observed between automatic $(42,7\%)$ and manual (77.4%) labeling. The main explanation may be provided by the first laparotomy node (representing $3,802$ vs $7,906$ patients, respectively). In practice, 549 different medical procedure codes, from different chapters of the hierarchy, were selected by medical experts to identify laparotomy in the database. Even if most of the codes are gathered in label 7, remaining laparotomy stays were grouped in other less frequent clusters, which do not appear in the final process model because of the size constraint in optimization. Thus, even if a quantitative replayability gap remains between the two presented methods, the qualitative interpretation resulting from the pathway analysis remains similar, as most of the interesting events were pointed out. Moreover, the final process model and the explanation of clusters furnish an interesting base for discussion with medical experts.

VIII. Conclusion

In this paper, a methodology to handle the complexity of event logs regarding activities was presented. Based on autoencoding, artificial labels to characterize these events are created, which can be used to apply process mining. Explainability of each label is possible through decoding, which allows the practical application of this method in fields like healthcare where transparency is essential. A design of experiments was presented, designed to mimic non-clinical claims databases regarding authors' knowledge. The ability of the method to both create relevant clusters and explain them accurately was demonstrated. In particular, the Variational Auto-Encoder shows better performances than others tested autoencoders, motivating the use of such learning methods for further applications. Finally, a case study has been presented, illustrating the potential of the methodology when applied on real healthcare data. The presented method sounds promising as a preprocessing solution for process mining, to handle the complexity of medical activities in non-clinical claims databases and of other similar databases.

Further work will focus on the deployment of the method in new case study, to experience the method and generalize its utilization. Particularly, a focus on providing interactive tools to explore the results and facilitate discussions with clinical experts will be made. As the complexity of medical event logs depends on the database, other complexities could be considered in further studies. As an example, the integration of free text with structured medical information to create pertinent medical labels could be interesting to consider in future

works. As noted by Helm et al [28], a lack of sufficient coding in existing case studies remains. Therefore, addressing such complexities in the method will be interesting for practical uses on case studies with insufficiently coded data. Also, the present methodology is a preprocessing step, applied before deploying a process discovery algorithm (in our example, based on optimization). An interesting subject could be the fusion of both steps, by integrating the labeling step directly during the optimization procedure of creating the final process model. Also, as the K-mean algorithm was used to perform clustering in the latent space, future work will focus on testing other clustering algorithms. On a larger scope, future works will focus on the use of the proposed methodology to perform supervised learning on complex event logs. The proposal of a transparent classification algorithm is of interest, particularly for patients pathways data. Furthermore, the bridging of process mining and deep learning is an interesting research track. The use of recent advancements in deep learning for process analysis and prediction seems promising, in particular if process mining is used as an interface between model learning and human understanding.

REFERENCES

[1] A. R. C. Maita, L. C. Martins, C. R. L. Paz, L. Rafferty, P. C. K. Hung, S. M. Peres, and M. Fantinato, "A systematic mapping study of process mining," *Enterprise Information Systems*, vol. 12, pp. 505–549, May 2018.

[2] W. van der Aalst, T. Weijters, and L. Maruster, "Workflow mining: discovering process models from event logs," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, pp. 1128–1142, Sept. 2004.

[3] W. v. d. Aalst, *Process Mining: Data Science in Action*. Berlin Heidelberg: Springer-Verlag, 2 ed., 2016.

[4] P. Tuppin, J. Rudant, P. Constantinou, C. Gastaldi-Ménager, A. Rachas, L. de Roquefeuil, G. Maura, H. Caillol, A. Tajahmady, J. Coste, C. Gissot, A. Weill, and A. Fagot-Campagna, "Value of a national administrative database to guide public decisions: From the système national d'information interrégimes de l'Assurance Maladie (SNIIRAM) to the système national des données de santé (SNDS) in France," *Revue d'Épidémiologie et de Santé Publique*, vol. 65, pp. S149–S167, Oct. 2017.

[5] M. Prodel, V. Augusto, X. Xie, B. Jouaneton, and L. Lamarsalle, "Stochastic simulation of clinical pathways from raw health databases," in *2017 13th IEEE Conference on Automation Science and Engineering (CASE)*, pp. 580–585, Aug. 2017.

[6] M. Prodel, V. Augusto, B. Jouaneton, L. Lamarsalle, and X. Xie, "Optimal Process Mining for Large and Complex Event Logs," *IEEE Transactions on Automation Science and Engineering*, vol. 15, pp. 1309–1325, July 2018.

[7] R. Phan, V. Augusto, D. Martin, and M. Sarazin, "Clinical Pathway Analysis Using Process Mining and Discrete-Event Simulation: an Application to Incisional Hernia," in *2019 Winter Simulation Conference (WSC)*, (National Harbor, MD, USA), pp. 1172–1183, IEEE, Dec. 2019.

[8] H. De Oliveira, V. Augusto, B. Jouaneton, L. Lamarsalle, M. Prodel, and X. Xie, "Optimal process mining of timed event logs," *Information Sciences*, vol. 528, pp. 58–78, Aug. 2020.

[9] M. Vandromme, J. Jacques, J. Taillard, L. Jourdan, and C. Dhaenens, "A Scalable Biclustering Method for Heterogeneous Medical Data," in *Machine Learning, Optimization, and Big Data* (P. M. Pardalos, P. Conca, G. Giuffrida, and G. Nicosia, eds.), Lecture Notes in Computer Science, (Cham), pp. 70–81, Springer International Publishing, 2016.

[10] M. Vandromme, J. Jacques, J. Taillard, A. Hansske, L. Jourdan, and C. Dhaenens, "Extraction and optimization of classification rules for temporal sequences: Application to hospital data," *Knowledge-Based Systems*, vol. 122, pp. 148–158, Apr. 2017.

[11] H. De Oliveira, M. Prodel, and V. Augusto, "Binary Classification on French Hospital Data: Benchmark of 7 Machine Learning Algorithms," in *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1743–1748, Oct. 2018.

[12] D. Ravì, C. Wong, F. Deligianni, M. Berthelot, J. Andreu-Perez, B. Lo, and G.-Z. Yang, "Deep Learning for Health Informatics," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, pp. 4–21, Jan. 2017.

[13] B. Shickel, P. J. Tighe, A. Bihorac, and P. Rashidi, "Deep EHR: A Survey of Recent Advances in Deep Learning Techniques for Electronic Health Record (EHR) Analysis," *IEEE Journal of Biomedical and Health Informatics*, vol. 22, pp. 1589–1604, Sept. 2018.

[14] C. Xiao, E. Choi, and J. Sun, "Opportunities and challenges in developing deep learning models using electronic health records data: a systematic review," *Journal of the American Medical Informatics Association*, vol. 25, pp. 1419–1428, Oct. 2018.

[15] R. Miotto, F. Wang, S. Wang, X. Jiang, and J. T. Dudley, "Deep learning for healthcare: review, opportunities and challenges," *Briefings in Bioinformatics*, vol. 19, pp. 1236–1246, Nov. 2018.

[16] E. Choi, M. T. Bahadori, A. Schuetz, W. F. Stewart, and J. Sun, "Doctor AI: Predicting Clinical Events via Recurrent Neural Networks," in *Machine Learning for Healthcare Conference*, pp. 301–318, Dec. 2016.

[17] E. Choi, M. T. Bahadori, J. Sun, J. Kulas, A. Schuetz, and W. Stewart, "RETAIN: An Interpretable Predictive Model for Healthcare using Reverse Time Attention Mechanism," in *Advances in Neural Information Processing Systems 29* (D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, eds.), pp. 3504–3512, Curran Associates, Inc., 2016.

[18] T. G. Erdogan and A. Tarhan, "Systematic Mapping of Process Mining Studies in Healthcare," *IEEE Access*, vol. 6, pp. 24543–24567, 2018.

[19] A. Pika, M. T. Wynn, S. Budiono, A. H. M. ter Hofstede, W. M. P. van der Aalst, and H. A. Reijers, "Towards Privacy-Preserving Process Mining in Healthcare," in *Business Process Management Workshops* (C. Di Francescomarino, R. Dijkman, and U. Zdun, eds.), vol. 362, (Cham), pp. 483–495, Springer International Publishing, 2019.

[20] R. Gatta, M. Vallati, C. Fernandez-Llatas, A. Martinez-Millana, S. Orini, L. Sacchi, J. Lenkowicz, M. Marcos, J. Munoz-Gama, M. Cuendet, B. de Bari, L. Marco-Ruiz, A. Stefanini, and M. Castellano, "Clinical Guidelines: A Crossroad of Many Research Areas. Challenges and Opportunities in Process Mining for Healthcare," in *Business Process Management Workshops* (C. Di Francescomarino, R. Dijkman, and U. Zdun, eds.), Lecture Notes in Business Information Processing, (Cham), pp. 545–556, Springer International Publishing, 2019.

[21] N. Martin, A. Martinez-Millana, B. Valdivieso, and C. Fernández-Llatas, "Interactive Data Cleaning for Process Mining: A Case Study of an Outpatient Clinic's Appointment System," in *Business Process Management Workshops* (C. Di Francescomarino, R. Dijkman, and U. Zdun, eds.), vol. 362, (Cham), pp. 532–544, Springer International Publishing, 2019.

[22] X. Xu, Y. Wang, T. Jin, and J. Wang, "Learning the Representation of Medical Features for Clinical Pathway Analysis," in *Database Systems for Advanced Applications* (J. Pei, Y. Manolopoulos, S. Sadiq, and J. Li, eds.), Lecture Notes in Computer Science, (Cham), pp. 37–52, Springer International Publishing, 2018.

[23] A. Alharbi, A. Bulpitt, and O. A. Johnson, "Towards Unsupervised Detection of Process Models in Healthcare," *Studies in Health Technology and Informatics*, vol. 247, pp. 381–385, 2018.

[24] R. Lenz and M. Reichert, "IT support for healthcare processes – premises, challenges, perspectives," *Data & Knowledge Engineering*, vol. 61, pp. 39–58, Apr. 2007.

[25] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, Massachusetts: MIT Press, Nov. 2016.

[26] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," *arXiv:1312.6114 [cs, stat]*, May 2014.

[27] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, Nov. 1987.

[28] E. Helm, A. M. Lin, D. Baumgartner, A. C. Lin, and J. Küng, "Towards the Use of Standardized Terms in Clinical Case Studies for Process Mining in Healthcare," *International Journal of Environmental Research and Public Health*, vol. 17, Feb. 2020.