International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Mining with Improved Deep Auto-Encoder for Medical Data Record Analysis Using Feature Representation

R. Ramprasad[1] , C. Jayakumari[2]

Submitted: 22/02/2023 **Revised:** 17/04/2023 **Accepted:** 10/05/2023

Abstract*:* Data mining is the finest technique for extracting knowledge from patient routes. Medical occurrences are intricately organized in event logs and frequently documented using several medical codes. Before applying process mining analysis, labeling these occurrences properly is challenging. This study presents a new method for managing complex events in medical records. Improved deep auto-encoding (IDAE) generates precise labels by grouping similar events in latent space. Also, an explanation is given by decoding the instances that correspond to the generated labels. When tested on simulated events, the method successfully uncovered hidden clusters in sparse binary data and provided precise justification for created labels. Real medical data are used in a case study. The outcomes support the theory's effectiveness in knowledge extraction from complicated event logs depicting patient pathways.

Keywords: *data analysis, learning model, medical data representation, clustering, mining data*

1. Introduction

The study of data generated in several systems, including those used in business, software development, and healthcare, combines many techniques. The ability to forecast outcomes or merely characterize the truth of facts can be achieved by extracting knowledge from such data [1]. Event logs present unique challenges for data analysts since they contain time, have high variability, and have intricate relationships between events. As a result, common data mining algorithms may need to be more obscure for some applications. To gather relevant data, a careful pre-processing step can be required [2]. These data describing processes can be found in the manufacturing, software engineering, and healthcare sectors. A data-driven strategy called process mining has been suggested to evaluate event logs. Event logs are unbiasedly utilized in addition to process modeling and data mining [3].

A claim database that is not clinical is French National Health Insurance Database (SNIIRAM). The volume of information is enormous and includes practically all French residents' healthcare reimbursements [4]. This database included 66 million people. Patients' hospitalizations are included in the SNIIRAM's list of payment details. However, the availability of specific medical data like vital signs, imaging reports, and test results could be improved [5]. The mapping of patient pathways, the clustering of medical data, and prediction tasks are all relevant tasks that may be performed with a database of this type. The intricacy of healthcare procedures is multifaceted [6]. Examples include free text, the level of detail in the events studied, and the simultaneous occurrence of many events that resulted in multiple codes defining a single event [7]. There may be a large number of these codes, which represent various medical tasks, and they frequently inherit hierarchical patterns. Since that relies on the pathology or health activity being investigated and is not always obvious, it is frequently important to employ medical competence to choose the optimum level to lead to productive actions [8]. The hierarchy still exists even though it might help simplify codes and lower total cardinality. One of the biggest problems with the non-clinical claim database is its complexity [9] – [10].

To examine the complexity of events and create meaningful labels, a novel methodology is presented in this work as its key contribution. The suggested method generates artificial labels from basic data using auto-encoding and clustering. The raw event log changes once those labels are applied to events because the overall variance of the events is decreased. The strategy offers practitioners transparency by explaining each artificially manufactured label. To handle these issues, the proposed IDAE is anticipated as a method for feature representation. It consists of various improved auto-encoders. Features are extracted from prior lower-level features. Like traditional encoding, every IDAE's output must reconstruct the lower-level features; however, the input data must reconstruct every layer. The layer-wise higher-level features are learned to represent prior features and input, which determines the data patterns superior to conventional AE.

[1] *Research Scholar, Bharathiar University.
Coimbatore, Tamil Nadu 641046.
India. Email: trramprasadphd@gmail.com*

[2] *Middle East College, oman,
Email: jayakumari@mec.edu.om*

To enhance the performance, the proposed model gives better flexibility.

This essay has the following format. A list of related works is given in Chapter 2. Chapter 3 describes the suggested methodology. The design of experiments is explained in Chapter 4 to validate the strategy, and a research report relating to current medical sources is presented. Conclusions and points of view are presented in Chapter 5.

2. Related Works

The only goals of the complicated healthcare system are the prevention, identification, and treatment of illnesses or disabilities for human well-being [11]. Medical personnel (doctors or other health workers), medical centers (such as hospitals and clinics for supplying pharmaceuticals and other technology for treatment or diagnosis), and a financial institution supporting the first two are the three primary elements of the health industry [12] – [13]. The healthcare sector comprises many professions, including medicine, dentistry, psychology, nursing, midwifery, and physiotherapy. Several degrees of medical intervention are required based on the severity of the condition [14] – [15]. Professionals provide connectivity for primary care, highly expertise emergency care, diagnostic treatment and surgical/diagnostically treatments (quaternary care) [16]. Medical professionals are in charge of keeping an eye on the patient's health records, which include details on diagnoses and medications, clinical and medical data (such as information from radiology and lab examinations), and other confidential or personal health records [17]. Such patient medical records were traditionally kept as either physical copies or computerized records. A paper-based file system was used to keep track of all the results of medical exams [18].

The "All of Us" initiative, recently announced by the National Institutes of Health (NIH) and accessible at https://allofus.nih. gov, is noteworthy [19]. Over the coming years, it plans to gather information from one million patients or more, including medical imaging, EHR, and environmental and socio-behavioral data [20]. Handling recent healthcare-related data has become simpler thanks to EHRs. Below is a concise list of a few of the special benefits of using EHRs. The fundamental benefit of EHRs is that they give medical staff greater accessibility to a person's full medical history [21] – [23]. The data contains medical diagnoses, demographics, clinical narratives, prescriptions, laboratory test results, and information on known allergies [24] – [25]. Diagnosing and treating medical problems are now more time effective due to a lower delay for early test reports. With time, there has been a considerable reduction in extra and pointless exams, misplaced instructions, and discrepancies brought on by inconsistent handwriting [26] – [28]. There is a huge enhancement in treatment coordination among various medical professionals. The prevalence of drug allergies has decreased due to resolving such logistical errors in prescription dosage and frequency.

Medical practitioners now have access to electronic and web-based platforms, considerably increasing their ability to practice medicine. These platforms provide automated alerts and cues for vaccinations, cancer screenings, abnormal test results, and routine examinations. Encouraging interaction among medical personnel and patients would result in better continuity of treatment and quick intervention [29]. Due to minimal paperwork, they are connected to digital authorization and quick insurance clearances. Fast data retrieval, improved public health surveillance through timely disease outbreak reporting, and improved key healthcare quality indicator reporting to organizations are all made possible by electronic health records (EHRs). EHRs can help manage the rising costs of medical insurance and can offer useful information regarding care provided to samples in workplace healthcare insurance. EHR reduces or eliminates delays and misunderstandings in claims management and billing. Millions of bits of important patient-life information are made accessible thanks to the internet and EHRs [30].

3. Methodology

This section concentrates on modeling a novel deep auto-encoder (see Fig 1) for data record analysis. A traditional stacking encoder may extract hierarchical characteristics based on the data input using a layer-wise pre-training method. Therefore, the high-level characteristics can be recovered by reducing the reconstruction error for the high-level characteristics' preceding low-level features. Since the reconstruction is inaccurate, the lowest to highest concealed layers have successive information loss. Hence, just a portion of the information from the raw data may be retained by the top-layer concealed features. Furthermore, it needs to be clarified how well each concealed layer's learned features specify underlying data. This issue is addressed by developing stacked auto-encoding. To help find the hidden features, the raw data input is transferred from each improved auto-encoder to the output layer that reconstructs the input layer. With this strategy, features have to reconstruct prior low-level features and the initial data. Hierarchical features could be methodically learned in a way that captures the observed raw data input from low to high layers by stacking several improved auto-encoders to model deeper network. Additionally, because every upgraded auto-encoder must independently rebuild the raw input data, information loss only builds up over time.

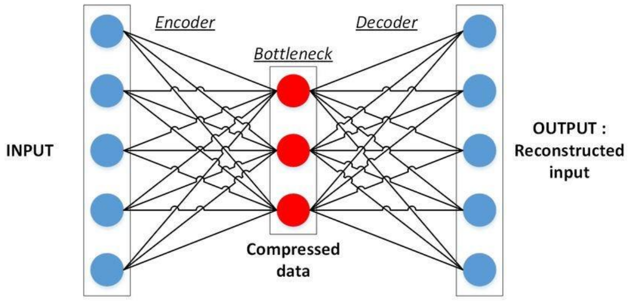


Fig 1 Conventional encoder model

3.1. Auto-encoding

An improved deep auto-encoder abbreviated as IDAE includes the hidden, input, & output layers. IDAE intends to reconstruct concurrently using feature and data in input layer. Network diagram for the IDAE is shown in Fig 2. The hidden and input layers' variable vectors are called h and z, corresponding to traditional auto-encoder. Here, x represents the raw input variable vector. In contrast to AE, IDAE's output layer attempts to collect data from the reconstructed input layer $\tilde{z}$ and input x. The terms "encoder parameters" and "decoder parameters" continue to be described as $z \in R^d$, $h \in R_1^d$, and $\tilde{z} \in R^d$, respectively. Hence, their activation functions are f and $\tilde{f}$. As a result, it is simple to determine that:

$$\begin{bmatrix} \tilde{z} \\ \tilde{x} \end{bmatrix} = g_{\theta}(z) = \tilde{f}(f(z)) \qquad (1)$$

At the input layer, using training samples of the raw observed data $\{x_1, x_2, \ldots, x_N\}$ and the corresponding feature data $z_i \in \{z_1, z_2, \ldots, z_N\}$, the reconstruction function is minimized to obtain the hidden layer data and the model parameters as follows:

$$J(W, \widetilde{W}, b, \widetilde{b}) = \frac{1}{2N} \sum_{i=1}^{N} \left(\left|\left|\tilde{x}_i - x_i\right|\right|^2 + \left|\left|\tilde{z}_i - z_i\right|\right|^2 \right) \qquad (2)$$

It makes it obvious that an improved encoder concentrates on restructuring its data from the input layer and aims to recreate the data. The input data structure for deep networks is crucial.

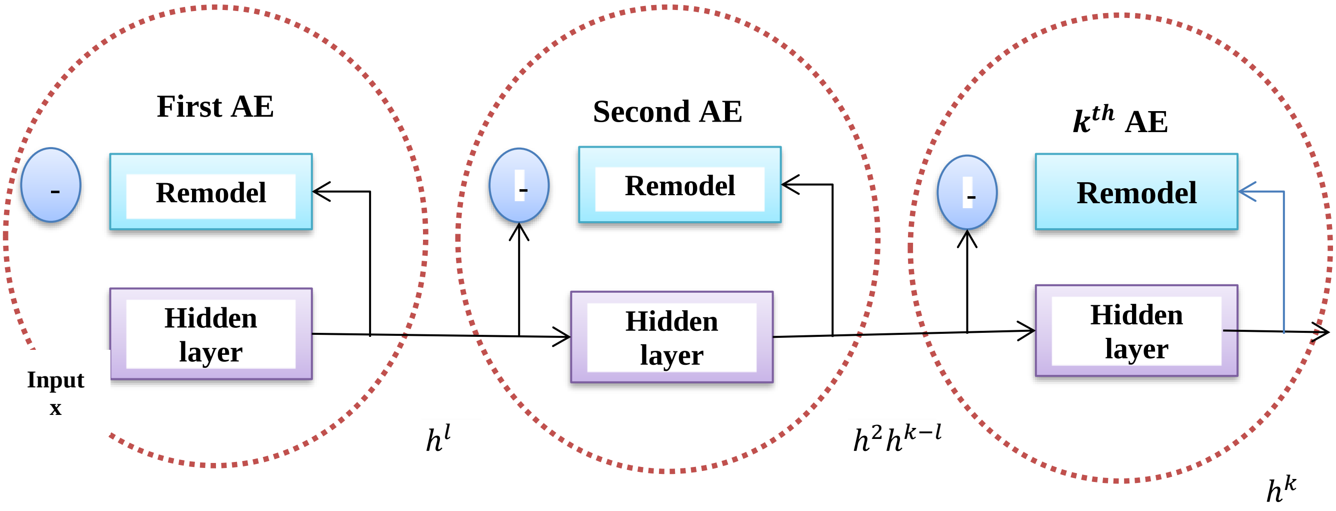


Fig 2 Improved auto-encoding

3.2. Stacking encoder

A deep network can be built by a hierarchically stacking encoder. Fig 2 depicts the proposed encoder model for organizational structure. The input x is first transmitted to the auto-encoder input to construct initial layer feature vector h^1. Input is reconstructed as x at encoders' output.

Initial auto-encoder pre-processing is done by utilizing the BP technique by lowering the reconstruction error among initial data x and data $\tilde{x}$ is reconstructed. As this input layer only contains raw data, it should be highlighted that the encoder is a conventional auto-encoder. The second-layer feature h^2 is created by connecting initial feature vector h^1 to encoders' input layer. In output layer, first-layer feature h^1 and input data x have all been synchronously recreated and are therefore referred to as $\tilde{h}^1$ and $\tilde{x}^1$, respectively. The second-layer feature data and associated network parameters can be obtained by pre-training the encoder using the BP approach. As a result, until the K^{th} value is achieved, all Ks can be built and trained independently.

The deep encoder structure may be used to learn hierarchical aspects of observed data with encoder 1 and moving up to the k^{th} auto-encoder. The unique pre-processing procedure for the auto-encoder is shown in Fig 3. The first supply of the raw samples of observable data $\{x_1, x_2, .., x_N\}$ is made to input layer of complete encoder. Initial feature data $\{h_1^1, h_2^1, .., h_N^1\}$ are produced by the first hidden layer employing weight and bias parameters of $\{W_1, b_1\}$ and non-linear activation function f. The feature learning technique cannot be applied because $\{W_1, b_1\}$ and $\{h_1^1, h_2^1, \dots, h_N^1\}$ is unknown. Therefore, encoder 1 is initially intended for pretraining. The encoder's first-level feature, h^1, uses the parameter $\{\widetilde{W}_1, \tilde{b}_1\}$ and activation function f to recreate data as x at output layer. The encoders' (1) objective function is reconstruction error over training data.

$$\boldsymbol{J_1(W_1, b_1, \widetilde{W}_1, \tilde{b}_1) = \frac{1}{2N}\sum_{i=1}^{N}\left||\tilde{x}_i - x_i\right||^2} \qquad (3)$$

Because the deep network retains the encoder component of the encoder after being pre-trained, the second-level features can be trained using $\{h_1^1, h_2^1, \dots, h_N^1\}$. The remaining hidden-level features $h^2, h^3, \dots, h^k$, then acquired layer by layer. Assuming that in this scenario, the k^{th} encoder $(k = 1, 2, \dots, K - 1)$ has been trained and developed. Then, forward propagation may determine the k^{th} level feature data $\{h_1^k, h_2^k, \dots, h_N^k\}$. The $(k + 1)^{th}$ level features $\{h_1^{k+1}, h_2^{k+1}, \dots, h_N^{k+1}\}$ are then derived using activation function f with parameter $\{W_{k+1}, b_{k+1}\}$. For this, it is necessary to construct $(k + 1)$, where h^k and h^{k+1} are input and hidden variables. The intrinsic raw data is preserved by simultaneously reconstructing the raw input data at output layer and feature data at input layer of $(k + 1)$, which are designated as $\tilde{x}^k$ and $\tilde{h}^k$, respectively. The encoder is pre-trained in the following manner to reduce the reconstruction error for k^{th} feature and input data:

$$\boldsymbol{J_{k+1}(W_{k+1}, b_{k+1}, \widetilde{W}_{k+1}, \tilde{b}_{k+1}) = \frac{1}{2N}\sum_{i=1}^{N}\left(\left||\tilde{x}_i^k - x_i\right||^2 + ||\tilde{h}_i^k - h_i^k||^2\right)} \qquad (4)$$

In some situations, the deep encoder has an advantage over the conventional encoder because it accumulates hierarchical data features from lower to higher level. Some added constraint is applied to output layer of every auto-encoder to regain initial data. Each feature is subsequently taught to aptly explain the underlying data and low-level features. Since every encoder attempts to recreate the original input data separately, this approach prevents information loss from lower to higher layers. Some learned features capture the intrinsic original data.

3.3. Procedure

It is beneficial and effectual to employ deep features for particular tasks. So, deep auto-encoder is utilized in this research for process data modeling, where it predicts the challenging quality output variable. Let the training data for the quality variable be $Y = \{y_1, y_2, \dots, y_N\}$ which corresponds to the observed input data that has not been processed, $X = \{x_1, x_2, \dots, x_N\}$. The training and testing phases make up the modeling process for encoder modeling. The K^{th} stacked encoders are pre-trained network during training step. The parameters are then modified using the BP approach by adding an output layer for quality variable to the network hidden layer. During training, the deep encoder model can forecast the quality variable for testing $X_t = \{x_1, x_2, \dots, x_{N_{test}}\}$. The following are the precise processes for encoder modeling:

1) As the supervised network fine-tuning and training datasets for pre-training, collect the quality output data $X = \{x_1, x_2, \dots, x_N\}$ and the raw input data $Y = \{y_1, y_2, \dots, y_N\}$. Then, the network structure is determined.

2) Reduce the reconstruction error between input data x that has been reconstructed and the raw data input x, to create and pre-train the first encoder. Hence, the parameter set $\{W_1, b_1\}$ and first-level feature data $H^1 = \{h_1^1, h_2^1, \dots, h_N^1\}$ are both attainable.

3). Then, using the input data $H^1 = \{h_1^1, h_2^2, \dots, h_N^1\}$, the second encoder model is created. For $k = 1$ in Eq. 4, encoder 2 can be trained by reducing reconstructed error function for initial feature h_l and unprocessed input x. Then employing encoder 2, next feature data $H^2 = \{h_1^2, \dots, h_N^2\}$ can be calculated along with $\{W_2, b_2\}$.

4). The entire network is a layer-trained layer-wise to acquire bias and weight parameter sets $\{W_k, b_k\}_{k=1,2,\dots,K.}$

5. After pre-training is complete, the top layer of the encoder is expanded for quality variable. The bias and weight are initially set up using pre-trained parameters $\{W_k, b_k\}_{k=1,2,\dots,K}$. Using back-propagation, the weight parameters can be changed to reduce the identified output error on training data.

6. After training, it is possible to integrate the tested input data $X_t = \{x_1, x_2, .., x_{N_{test}}\}$ using the learned network. Testing samples quality can be predicted using forward propagation from input layer to output layer as follows: $\hat{Y}_t = \{\hat{y}_1, \hat{y}_2, \dots, \hat{y}_{N_{test}}\}$.

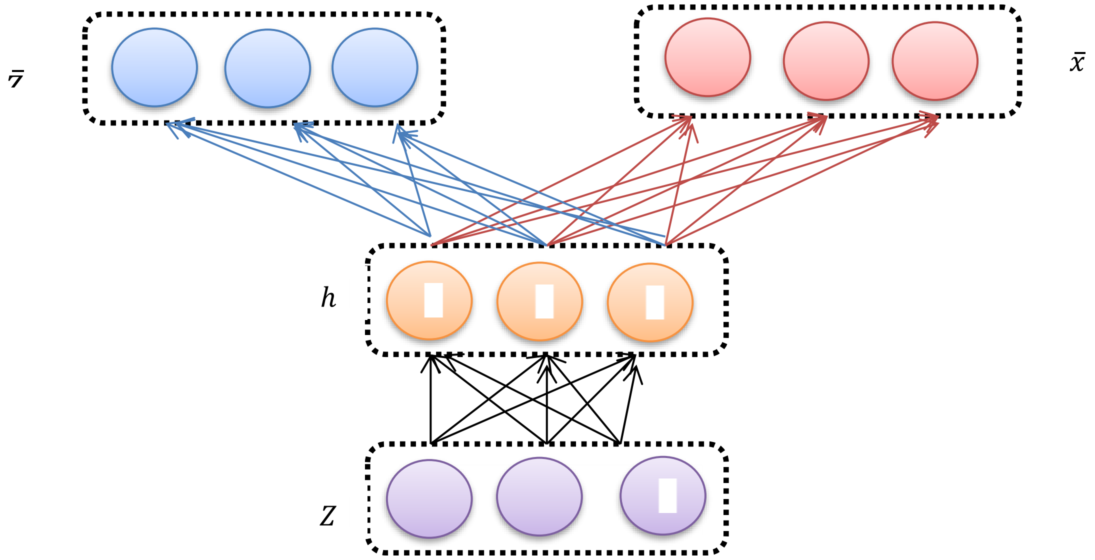


Fig 3 Improved deep auto-encoder

4. Numerical Results and Discussion

This section presents an experimental design using artificial data. This experiment has several goals: a) to validate the precision of the methodology in event logs to identify hidden patterns (clusters); b) to illustrate the cluster labels' interpretation by the decoder's accuracy; c) to evaluate the performance of direct clustering versus auto-encoding on sparse data; and d) to compare the performance of various auto-encoding techniques.

Event logs, made up of events whose precise labels have been verified by the issue mentioned, serve as input data. Hence, hidden labels were reflected in the data by utilizing groups of actions from which vectors would be constructed. Codes from the same "set" are frequently used to define similar hospitalizations; these codes can be estimated using clinicians' clinical knowledge. These data specialists' discoveries about non-clinical claims data served as the inspiration for this structure. For instance, a specific operation is entered into the database using operation's code and certain codes connected to some diagnoses relating patient's health. There are correlations between two stays for the same procedure, even though they are rarely identical, and some codes are derived from the same "set."

Upon consideration of these comments, synthetic data were created to represent an activity matrix with every row representing an activity vector and column representing activities on a scale of $1 - of - k$. The term "number of clusters to find" refers to the variety of hidden labels in the data. N_k vectors were created with the following: $N_k = N(\mu_N = \frac{2500}{k})$, for every label k, were created for each label $M_k^C \in N^*$ vector. The overall number of defining activities for every label k is denoted by the notation $M_k^C = N^*$, structured in a manner that $M_k^C = N\left(\mu_C, \frac{\mu_C}{5}\right)$, where $\mu_C = \alpha \times \beta \times \mu_N$. Here, M is the sum of all the distinct actions engaged. A number $M^a = \alpha \times \mu_N \in N^*$ of activities of a particular label k were randomly selected to generate each activity vector. Here, M^a is the amount of randomly selected activities from a list of common activities. The value of the corresponding attribute was set to 1 for these activities while remaining at 0 for all other activities. An overlapping ratio, additionally introduced, indicates the number of activities shared between a label and the label that is closest to it. Besides that, $N_{noisy} =$ 250 noisy occurrences were generated. In the generated data set, the actions that composed these noisy occurrences were dispersed among all feasible activities, and no specific pattern was linked to hidden label k. Here, N represents total number of rows and defined as $N = N_{noisy} + \prod_{k=1}^{k} N_k$. The parameters used in the experiment design result in an approximation of sparsity $\tilde{S}$ ranging from 0.9 to 0.99.

4.1. Discussion

Conventional auto-encoder methods AE, IDAE, and IDAE's performance were put into practice. Here, feed-forward networks with fully connected layers (four) totaling $10 \times d_{latent}, 5 \times d_{latent}, and\, d_{latent},$ where dlatent is the dimension of the latent space, equivalent to eight. For IDAE, the noise was defined as the arbitrary selection of components in vectors with (0-1). Each vector is made with 1% noise. Many Gaussian distributional parameters were acquired for IDAE (latent variables. The inverse estimate was the loss function to be minimized. An asymmetric design between the encoder and decoder was used for auto-encoder training for every data parameter combination. Dropout and $L2$ regularisation were utilized for each layer to minimize over-fitting during training. Adam was the selected optimizer for training, and a 32-piece mini-batch was used. There would be 1000 epochs altogether. 20% of the data is adopted to evaluate validation error for termination, while the remaining 80% was used for training (with 25 iterations of patience). Following auto-encoder training, all training and validation data were used to apply K-mean clustering. K was set by maximizing the mean for $K \in [K_{min}, K_{max}]\ with\ K_{min} = 2\ and\ K_{max} = 15.$

Performances were assessed regarding explainability and clustering, respectively. An automated method created confusion matrix between detected clusters and hidden labels to assess the effectiveness of clustering. To improve accuracy, just the detected clusters' columns were discretized, allowing the proposed clusters to be compared to corresponding potential ones in hidden ones (the sum of the diagonal). Accuracy, subsequently determined for the resulting confusion matrix, is defined as a method's capacity to assign the right label to each occurrence precisely. The explaining F-score F_η is a new metric that measures how well the strategy can account for discovered clusters. Let the function $c: k \to c(k) \in K^{pred}$ return the appropriate cluster label in line with the previously specified confusion matrix optimization. Let $k \in [1, \kappa] = K^{true}$ represent the label and hidden cluster, respectively. We calculate the cluster $c(k)$ average of the decoded elements and contrast the cluster's typical activities (pred c(k)) with those of the associated label (true k):

$$\boldsymbol{F_\eta = \frac{2 * R_\eta * P_\eta}{R_\eta + P_\eta}} \tag{5}$$

$$\boldsymbol{R_{\eta} = \frac{\sum_{k \in K^{true}} . a_k^{true} \cap a_{c(k)}^{pred}}{\sum_{k \in K^{true}} card(a_k^{true})}} \tag{6}$$

$$\boldsymbol{P_\eta = \frac{\sum_{k \in K^{true}} . a_k^{true} \cap a_{c(k)}^{pred}}{\sum_{k \in K^{pred}} card(a_{c(k)}^{true})}} \tag{7}$$

A high explaining recall demonstrates the capacity of the function to link hidden activities of the detected labels with matched activities. Besides, high explanatory precision for decoding that retains engaging track activities without generic. Every label that is found corresponds to hidden label. There might be a difference between the number of concealed and discovered clusters. The explaining recall will be impacted by the number of expected vs. concealed clusters and the explaining precision by the number of predicted versus true clusters.

The threshold will significantly affect the metrics that were previously specified. Here, an automatic technique was applied in the experiment design process. A list of each decoded value for a cluster named l was generated using the average decoding vector $\bar{X}_l$. This list's items were arranged in decreasing order. A distinct list of values was produced by differentiating this curve. An appropriate threshold for sustaining activities in the connected cluster's explanation set was generated instantly using the resulting curve's minimum value. Ten datasets were created for each set of parameters. Before the rows were shuffled, the columns (activities) were shuffled (events). The proposed method was evaluated compared to a baseline containing straight K-mean clustering without auto-encoding and the previously mentioned auto-encoders (AE, IDAE, and IDAE). Performances were examined using the clustering and elucidating metrics' mean and standard deviation.

Table 1 provides a summary of the results. A total of 24 trials with varying degrees of difficulty were run. The evaluation techniques' F_η score and accuracy are shown for every set of parameters. Findings demonstrate that in sparse high-dimensional space, auto-encoding approaches outperform direct clustering (exceptionally, experiments 15 and 21 show weak IDAE performances). Consequently, for the suggested methodology, auto-encoding is crucial in data transformation. The outcomes also show that IDAE consistently performs better than the alternative approaches regarding accuracy and F_η. Even if the standard deviation rises for challenging studies, IDAE exhibits less fluctuation than the other approaches. The findings support using IDAE as a component of the suggested approach to identify and explain precise clusters.

Table 1 Encoder parameters

Exponential				
S. No	k	α	β	γ
1	5	0.06	3	0.0
2	5	0.06	3	0.11
3	5	0.06	3	0.26
4	5	0.06	6	0.0

5	5	0.06	6	0.11
6	5	0.06	6	0.26
7	5	0.11	3	0.0
8	5	0.11	3	0.11
9	5	0.11	3	0.26
10	5	0.11	6	0.0
11	5	0.11	6	0.11
12	5	0.11	3	0.26
13	10	0.06	3	0.0
14	10	0.06	3	0.11
15	10	0.06	6	0.26
16	10	0.06	6	0.0
17	10	0.06	6	0.11
18	10	0.06	3	0.26
19	10	0.11	3	0.0
20	10	0.11	3	0.11
21	10	0.11	3	0.26
22	10	0.11	6	0.0
23	10	0.11	6	0.11
24	10	0.11	6	0.26

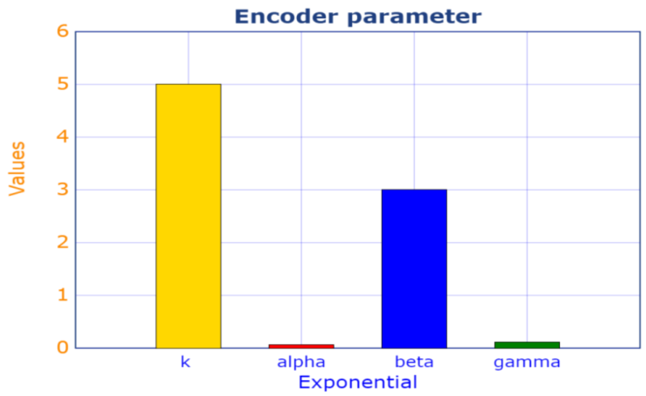


Fig 4 Encoder parameters

4.2. Dataset description

Here, the SNIIRAM database was mined for the data. All anonymous patients who had an IH within 5 years of the procedure and underwent their first laparotomy in 2010 were chosen. As a result, 7906 patients in total were included in the study, from who complete hospitalization data was retrieved. Every patient's stay in the hospital was converted into a record of prescribed medical procedures. As a result, the activity set was divided into the following sections:

$$\boldsymbol{A = A_{MD} \cup A_{AD} \cup A_{MP} \cup A_{D} \cup A_{TAD}} \qquad (8)$$

- In the (AD) ICD-10 classification system, AAD stands for a group of extra diagnoses;
- In the (AD) ICD-10 classification system, AAD stands for a group of extra diagnoses;
- According to the (MP) Fre0nch CCAM classification system, AMP refers to a group of medical operations.
- With coding system (D) related to the ATC (anatomical therapeutic chemical) classifications, AD gives collection of medications provided;
- In French, ATAD stands for a group of medications with temporary delivery permission; AMD stands for a group of primary diagnoses or the causes of hospitalization.

For each activity code, the associated activity set also contained hierarchical knowledge (codes for the hierarchy's higher levels). As a result, relationships between activity codes belonging to the same group might be enabled during auto-encoding. As evidenced by the data below, it improves cluster interpretability by provisioning hierarchical knowledge and adjusting the coding precision by clusters. The known, highly frequent stays associated with chemotherapy or dialyzes have been filtered. The log activity set's size was cut by 85.7%, while 95.0% of the codes were kept after filtering out codes that appeared fewer than 50 times. The study's final event record had 57533 events (stays), 2228 distinct activity codes, and 7906 traces (patients). The generated activity matrix, measuring 57533 by 2228, was subjected to the previously described approach utilizing IDAE as the auto-encoder. $K = 15$ clusters were used, which is a reasonable compromise between clusters and final process. It uses the design to medical event logs. For process model optimization, the maximum locations, edges, and nodes were set to 5, 25, and 15, respectively.

4.3. Results

The outcomes of automatic labeling were evaluated by comparing them to a process model starting from a manually labeled event log based on the authors' prior knowledge of pathophysiology. Each process model's edges and nodes are scaled by the total number of patients they represent. It is best to read the process models from left to right. Table 2 fully explains the clustering results achieved and displayed in Fig 4. There are commonalities in the process models, as seen in Fig 5. Laparotomy-related medical procedure codes most frequently fall under the hierarchy's "Therapeutic acts on the digestive system" code. The linked cluster (label 7) thus shows up at the start of the paths. The pathway's next section features label 2, label 14, and other labels, including IH-related codes (combining codes for IH and laparotomy). However, the process being considered is primarily filled by stays linked to diagnostic process (labels 6 and 12), particularly on canal (label 10).

Table 2 Accuracy comparison

Accuracy									
Basic AE		General AE		DAE		VAE		IDAE	
Average	SD	Average	SD	Average	SD	Average	SD	Average	SD
0.56	0.07	0.67	0.17	0.72	0.06	0.91	0.02	0.93	0.04
0.56	0.08	0.67	0.17	0.72	0.09	0.90	0.02	0.93	0.04
0.57	0.08	0.69	0.17	0.73	0.03	0.90	0.04	0.93	0.06
0.39	0.07	0.60	0.17	0.62	0.19	0.86	0.04	0.88	0.06
0.38	0.05	0.69	0.15	0.58	0.19	0.87	0.04	0.89	0.06
0.44	0.07	0.60	0.20	0.64	0.16	0.86	0.02	0.88	0.03
0.52	0.08	0.70	0.14	0.71	0.08	0.89	0.02	0.90	0.03
0.55	0.08	0.64	0.17	0.71	0.09	0.90	0.02	0.91	0.03
0.40	0.10	0.69	0.17	0.71	0.13	0.89	0.02	0.90	0.03
0.37	0.08	0.58	0.17	0.63	0.16	0.83	0.03	0.85	0.04
0.46	0.06	0.61	0.19	0.70	0.14	0.84	0.02	0.88	0.03
0.50	0.10	0.56	0.18	0.63	0.18	0.85	0.03	0.86	0.04
0.55	0.07	0.62	0.19	0.63	0.16	0.88	0.05	0.90	0.08
0.54	0.07	0.62	0.23	0.67	0.15	0.89	0.06	0.90	0.07
0.23	0.15	0.46	0.23	0.39	0.24	0.87	0.06	0.88	0.07
0.24	0.06	0.40	0.25	0.71	0.05	0.83	0.05	0.85	0.06
0.24	0.08	0.32	0.22	0.53	0.24	0.84	0.03	0.85	0.04
0.45	0.05	0.59	0.21	0.43	0.24	0.84	0.03	0.85	0.04
0.51	0.09	0.59	0.20	0.67	0.05	0.87	0.04	0.88	0.05
0.21	0.06	0.54	0.18	0.62	0.18	0.87	0.03	0.88	0.04
0.25	0.12	0.33	0.23	0.45	0.23	0.87	0.03	0.88	0.04
0.22	0.03	0.33	0.20	0.54	0.20	0.83	0.03	0.85	0.04
0.21	0.03	0.33	0.19	0.50	0.25	0.83	0.04	0.85	0.05
0.22	0.02	0.33	0.16	0.51	0.21	0.80	0.04	0.82	0.05
0.23	0.07	0.34	0.17	0.52	0.21	0.80	0.04	0.82	0.05

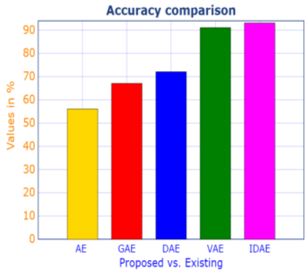


Fig 5a Accuracy comparison

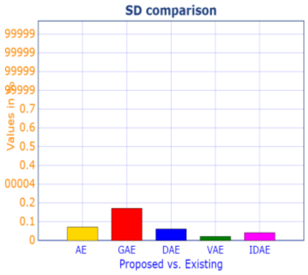
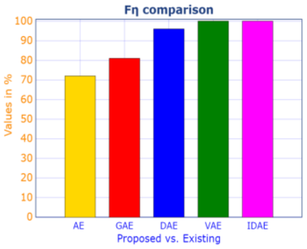


Fig 5b SD comparison

Table 3 F_η comparison

F_η									
Basic AE		General AE		DAE		VAE		IDEA	
Average	SD	Average	SD	Average	SD	Average	SD	Average	SD
0.72	0.14	0.81	0.28	0.96	0.12	1.00	0.00	1.00	0.00
0.69	0.18	0.82	0.33	0.96	0.15	1.00	0.00	1.00	0.00
0.72	0.18	0.83	0.26	0.93	0.13	0.98	0.08	0.99	0.09
0.38	0.15	0.74	0.28	0.78	0.32	1.00	0.00	1.00	1.00
0.38	0.1	0.84	0.2	0.68	0.3	1.00	0.0	1.00	1.0

	2		7		8		0		0
0.38	0.14	0.69	0.36	0.88	0.30	1.00	0.00	1.00	1.00
0.48	0.14	0.73	0.24	0.92	0.16	1.00	0.00	1.00	1.00
0.66	0.16	0.80	0.28	0.94	0.17	1.00	0.00	1.00	1.00
0.64	0.14	0.73	0.26	0.91	0.21	0.95	0.12	0.96	0.11
0.70	0.09	0.74	0.27	0.78	0.26	1.00	0.00	1.00	0.00
0.41	0.08	0.77	0.26	0.91	0.26	0.95	0.10	0.96	0.11
0.37	0.15	0.49	0.28	0.81	0.29	1.00	0.00	1.00	0.00
0.42	0.11	0.37	0.30	0.85	0.25	0.98	0.06	0.99	0.05
0.63	0.10	0.25	0.28	0.89	0.26	0.97	0.10	0.98	0.11
0.72	0.21	0.66	0.30	0.45	0.38	0.95	0.12	0.96	0.11
0.73	0.11	0.74	0.38	0.94	0.10	0.96	0.07	0.97	0.08
0.23	0.17	0.69	0.34	0.62	0.36	0.97	0.11	0.98	0.10
0.25	0.09	0.32	0.31	0.42	0.39	0.94	0.12	0.95	0.11
0.18	0.12	0.31	0.27	0.90	0.14	1.00	0.07	1.00	0.06
0.60	0.09	0.29	0.31	0.82	0.28	0.98	0.06	0.99	0.05
0.73	0.22	0.30	0.37	0.53	0.39	0.95	0.06	0.96	0.05
0.70	0.06	0.31	0.28	0.66	0.34	0.96	0.00	0.97	0.00
0.15	0.08	0.33	0.25	0.62	0.42	0.93	0.11	0.94	0.10
0.18	0.11	0.29	0.31	0.57	0.32	0.90	0.13	0.91	0.12
0.22	0.11	0.33	0.33	0.58	0.31	0.91	0.06	0.91	0.05



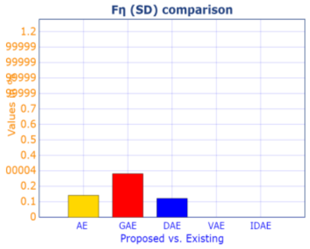


Fig 6a Accuracy comparison **Fig 6b** SD comparison

The automatic labeling method based on raw data brought these issues to light even though the stays were not considered when manually applied labels. These might be connected to the patient's medical monitoring or examination, a potential post-operative consequence. With just a little initial user input, this example demonstrates how raw data can be used to generate other sensitive data. Replayability score provides quantitative fitness assessments; it is easy to see the difference between automatic and manual labeling. The initial laparotomy node might offer the primary justification. Laparotomy was identified in the database by medical professionals employing 549 distinct medical procedure codes from various hierarchy chapters. Although label 7 applies to the bulk of the codes, the other laparotomy stays are sorted into other because of the size limitations of the optimization process, do not exist in the final process. While the majority of intriguing events were found, qualitative interpretation offered by analysis is comparable, even though there is still a difference in the quantitative replayability between the two techniques. Also, the summary of clusters and the final process model provides a stimulating starting point for discussions with medical experts.

5. Conclusion

This research offered an approach to deal with the complexity of activity-related event logs. Process mining can be used to characterize these occurrences using artificial labels that are constructed based on auto-encoding. Decoding permits the explanation of each label, enabling the use of this technology in real-world settings like the healthcare sector, in which openness is essential. Considering the authors' knowledge, an experimental design was given to simulate non-clinical claims databases. It was shown that the approach could both produce pertinent clusters and provide a precise explanation for them. Improved Deep Auto-encoder, in particular, outperforms the other studied auto-encoders, encouraging the usage of such learning techniques in other applications. The proposed methodology shows potential as a pre-processing strategy for process mining to deal with the complex nature of clinical procedures in non-clinical datasets and related databases. Future research will apply the suggested methods to supervise learning on complicated event logs on a bigger scale. Particularly for data on patient paths, the suggestion of a transparent classification method is intriguing. A fascinating study area is the integration of deep learning and process mining. Recent developments in deep learning have promise for prediction, particularly if process mining is used to integrate model.

References

[1] Mauro AD, Greco M, Grimaldi M. A formal definition of big data based on its essential features. Libr Rev. 2016;65(3):122–35.

[2] Reisman M. EHRs: the challenge of making electronic data usable and interoperable. Pharm Ther. 2017;42(9):572–5.

[3] Yin Y, et al. The Internet of things in healthcare: an overview. J Ind Inf Integr. 2016;1:3–13

[4] Yuan, Z. Ge, B. Huang, and Z. Song, "A probabilistic just-in-time learning framework for soft sensor development with missing data," IEEE. T. Contr. Syst. T., vol. 25, no. 3, pp. 1124-1132, 2017.

[5] Yuan, Y. Wang, C. Yang, W. Gui, and L. Ye, "Probabilistic density-based regression model for soft sensing of nonlinear industrial processes," J. Process Contr., vol. 57, pp. 15-25, 2017.

[6] Yuan, C. Ou, Y. Wang, C. Yang, and W. Gui, "Deep quality-related feature extraction for soft sensing modeling: A deep learning approach with hybrid VW-SAE," Neurocomputing, DOI: 10.1016/j.neucom.2018.11.107, 2020.

[7] Wang, Z. Pan, X. Yuan, C. Yang, and W. Gui, "A novel deep learning based fault diagnosis approach for chemical process with extended deep belief network," ISA T., vol. 96, pp. 457-467, 2020.

[8] Yuan, L. Li, Y. Wang, C. Yang, and W. Gui, "Deep learning for quality prediction of a nonlinear dynamic process with variable attention-based long short-term memory network," Can. J. Chem. Eng., DOI: 10.1002/cjce.23665, 2019.

[9] Yuan, L. Li, Y. A. W. Shardt, Y. Wang, and C. Yang, "Deep learning with spatiotemporal attention-based LSTM for industrial soft sensor model development," IEEE. T. Ind. Electron., pp. to be published, 2020.

[10] Liu, C. Yang, Z. Gao, and Y. Yao, "Ensemble deep kernel learning with application to quality prediction in industrial polymerization processes," Chemometrics. Intell. Lab. Syst, vol. 174, pp. 15-21, 2018.

[11] Wang, B. Gopaluni, J. Chen, and Z. Song, "Deep Learning of Complex Batch Process Data and Its Application on Quality Prediction," IEEE. T. Ind. Inf., vol. 10.1109/TII.2018.2880968 2019.

[12] Yuan, C. Ou, Y. Wang, C. Yang, and W. Gui, "A novel semi-supervised pre-training strategy for deep networks and its application for quality variable prediction in industrial processes," Chem. Eng. Sci., vol. 217, pp. 115509, 2020.

[13] Shang, F. Yang, D. Huang, and W. Liu, "Data-driven soft sensor development based on deep learning technique," J. Process Contr., vol. 24, no. 3, pp. 223-233, 2014.

[14] Yuan, Y. Wang, C. Yang, and W. Gui, "Stacked isomorphic autoencoder based soft analyzer and its application to the sulfur recovery unit," Inform. Sci., vol. To be published, 2020.

[15] Wang, D. Wu, and X. Yuan, "A two-layer ensemble learning framework for the data-driven soft sensor of the diesel attributes in an industrial hydrocracking process," J. Chemometrics., vol. 33, no. 12, pp. e3185, 2019.

[16] 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.

[17] Rav`ı, 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.

[18] 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.

[19] Helm, A. M. Lin, D. Baumgartner, A. C. Lin, and J. Kung, "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

[20] Valikodath NG, et al. Agreement of ocular symptom reporting between patient-reported outcomes and medical records. JAMA Ophthalmol. 2017;135(3):225–31

[21] Echaiz JF, et al. Low correlation between self-report and medical record documentation of urinary tract infection symptoms. Am J Infect Control. 2015;43(9):983–6.

[22] Sherubha, "Graph-Based Event Measurement for

Analyzing Distributed Anomalies in Sensor Networks", Sådhanå(Springer), 45:212, https://doi.org/10.1007/s12046-020-01451-w

[23] Sherubha, "An Efficient Network Threat Detection and Classification Method using ANP-MVPS Algorithm in Wireless Sensor Networks", International Journal of Innovative Technology and Exploring Engineering (IJITEE), ISSN: 2278-3075, Volume-8 Issue-11, September 2019

[24] Sherubha, "An Efficient Intrusion Detection and Authentication Mechanism for Detecting Clone Attack in Wireless Sensor Networks", Journal of Advanced Research in Dynamical and Control Systems (JARDCS), Volume 11, issue 5, Pg No. 55-68

[25] Yuan, L. Li, and Y. Wang, "Nonlinear dynamic soft sensor modeling with supervised long short-term memory network," IEEE. T. Ind. Inf., vol. 16, no. 5, pp. 3168-3176, 2020.

[26] Liao, J. Y. Tang, and X. Zhao, Course drop-out prediction on MOOC platform via clustering and tensor completion, Tsinghua Science and Technology, vol. 24, no. 4, pp. 412–422, 2019.

[27] Luo, and S. S. Zhao, Context-aware social media user sentiment analysis, Tsinghua Science and Technology, vol. 25, no. 4, pp. 528–541, 2020.

[28] J. Chen, Z. Lv, and H. Song, "Design of personnel big data management system based on blockchain," Future Gener. Comput. Syst., vol. 101, pp. 1122–1129, 2019

[29] C. Zhao, L. Ren, Z. Zhang, and Z. Meng, "Master data management for manufacturing big data: A method of evaluation for data network," World Wide Web, vol. 23, pp. 1407–1421, 2019

[30] D. Wu, L. Zhu, Q. Lu, and S. Sakr, "HDM: A composable framework for big data processing," IEEE Trans. Big Data, vol. 4, no. 2, pp. 150–163, Jan. 2018