Doctor XAI

An ontology-based approach to black-box sequential data classification explanations

Cecilia Panigutti
Scuola Normale Superiore
cecilia.panigutti@sns.it

Alan Perotti
ISI foundation
alan.perotti@isi.it

Dino Pedreschi
University of Pisa
dino.pedreschi@unipi.it

ABSTRACT

Several recent advancements in Machine Learning involve black-box models: algorithms that do not provide human-understandable explanations in support of their decisions. This limitation hampers the fairness, accountability and transparency of these models; the field of eXplainable Artificial Intelligence (XAI) tries to solve this problem providing human-understandable explanations for black-box models. However, healthcare datasets (and the related learning tasks) often present peculiar features, such as sequential data, multi-label predictions, and links to structured background knowledge. In this paper, we introduce *Doctor XAI*, a model-agnostic explainability technique able to deal with multi-labeled, sequential, ontology-linked data. We focus on explaining *Doctor AI*, a multi-label classifier which takes as input the clinical history of a patient in order to predict the next visit. Furthermore, we show how exploiting the temporal dimension in the data and the domain knowledge encoded in the medical ontology improves the quality of the mined explanations.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**; **Machine learning**; • **Applied computing** → **Health care information systems**.

KEYWORDS

explainable artificial intelligence, machine learning, healthcare data

ACM Reference Format:
Cecilia Panigutti, Alan Perotti, and Dino Pedreschi. 2020. Doctor XAI: An ontology-based approach to black-box sequential data classification explanations . In *Conference on Fairness, Accountability, and Transparency (FAT* '20), January 27–30, 2020, Barcelona, Spain.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3351095.3372855

1 INTRODUCTION

The recent availability of large amounts of electronic health records (EHRs) provides an opportunity for low-cost access to rich longitudinal clinical data. EHRs are usually noisy, sparse, fragmented, have high dimensionality and nonlinear relationships among variables [48]. The ability of Deep Learning techniques [26] to model such highly nonlinear relationships enables them to have good predictive performance without the need for feature engineering. This has led to many successful applications of such technologies to clinical tasks based on EHR data [42, 49]. Deep Learning techniques have been proven useful for information extraction from clinical notes [21], patients and medical concept representation learning [35], outcome prediction [9, 12, 29, 38, 42] and new phenotype discovery [8, 25]. Some of these works focus on developing predictive models able to forecast any future diagnosis. Most of these models take as input the clinical history of the patient and output the set of future diagnoses [9, 31]. This kind of versatile models, able to tackle mixed scenarios, can be extremely useful in day-to-day clinical practice. However, the complexity of Deep Learning models hinders the straightforward understanding of the rationale behind their decisions [20]. This lack of interpretability prevents the deployment of such models in real-world healthcare scenarios. For instance, it was proven that biases in the data [7] and adversarial examples [15] can easily mislead such black-boxes. Furthermore, being able to understand the reasoning behind the model's predictions would increase the healthcare professionals' trust in such a technology and would increase its acceptance and use [14]. Recently, being able to explain the reasoning behind machine learning decisions also became a legal requirement prescribed by Art. 22 of the GDPR (General Data Protection Regulation) [34]. Indeed, GDPR requires the data processor to provide to the data subject *meaningful information about the logic involved, as well as the significance and the envisaged consequences of such processing for the data subject* in case of a *decision based solely on automated processing* which might produce *legal effects concerning him or her*[1].

In this paper, we introduce **Doctor XAI**, a novel explainability technique able to deal with multi-labeled, sequential, ontology-linked data. *Doctor XAI* is a post-hoc interpretability method that focuses on *local* explanations, i.e., it explains the rationale behind the classification of a single data point. It is also *model-agnostic*, as it produces explanations whose computation is not based on the black-box inner parameters or structure. In this regard, *Doctor XAI* is similar to other black-box-agnostic techniques [17, 36, 39, 40]. However, to the best of our knowledge, ours is the first agnostic XAI technique applicable to sequential and ontology-linked data classification. The mining of sequential data is of pivotal importance in healthcare since this format is how typically the clinical history of the patient is represented. Furthermore, the presence of an ontology associated with the data is widespread in the medical and biological fields [5, 43]. Given a patient whose clinical history classification needs an explanation, *Doctor XAI* first generates a

* AP acknowledges partial support from Research Project "Casa Nel Parco" (POR FESR 14/20 - CANP - Cod. 320 - 16 - Piattaforma Tecnologica "Salute e Benessere") funded by Regione Piemonte in the context of the Regional Platform on Health and Wellbeing and from Intesa Sanpaolo Innovation Center. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

FAT '20, January 27–30, 2020, Barcelona, Spain*

ACM ISBN 978-1-4503-6936-7/20/02.
https://doi.org/10.1145/3351095.3372855

[1] Art. 13 paragraph 2f, Art. 14 paragraph 2g, Art. 15 paragraph 1h

local synthetic neighborhood around the selected patient exploiting the semantic information encoded in the ontology and uses the black-box model to label it. Then it transforms the clinical history of such synthetic patients into a format suitable to train a decision tree. This transformation allows taking the sequential nature of the data into account. Finally, *Doctor XAI* trains a decision tree on the labeled synthetic neighborhood, and it extracts an explanation in the form of a decision rule. We applied *Doctor XAI* to explain the decisions of *Doctor AI* [9], a recurrent neural network which takes as input patients' sequential EHR data and predicts the next visit set of diagnoses. We compared the quality of the explanations provided by *Doctor XAI* against those of the same technique without the ontological information. We show how exploiting the semantic information encoded in the ontology increases the performance of the explainability technique across all the evaluated metrics. We want to highlight that, even if our system deals by design with sequential, multi-labeled, ontology-based data, none of these features is strictly necessary: *Doctor XAI* can be used with datasets displaying any combination of the three aforementioned features, by exploiting only the corresponding specific modules. The contribution of this paper is twofold:

- We propose an agnostic explainability technique able to tackle the sequential data classification explanation problem.
- We show how exploiting the semantic information present in the medical ontology increases the quality of explanation.

The paper is structured as follows: Section 2 reviews related work on the topics of explainable artificial intelligence, machine learning for sequential healthcare data and ontology use in machine learning; Section 3 introduces the algorithmic building blocks of our XAI technique as well as the pipeline as a whole; Section 4 presents experimental setups and results; Section 5 ends the paper with the conclusions and directions for future work.

2 RELATED WORK

2.1 Doctor AI

Doctor AI [9] is a Recurrent Neural Network (RNN) with Gated Recurrent Units (GRU) that predicts the patient's next visit time, diagnoses and medications order. We focus here only on the diagnosis prediction task of the model. The authors trained their model on 260.000 patients of the EHRs database of Sutter Health Palo Alto Medical Foundation. The multi-hot input vector representing the diagnoses at each time-step of patient clinical history is first projected in a lower-dimensional space and then received as input by a stack of RNN layers implemented using GRUs. Finally, a Softmax layer is used to predict the diagnosis codes of the next time-stamp. The predictive performance of *Doctor AI* are evaluated using recall@n with $n = 10, 20, 30$ achieving 0.79 recall@30.

2.2 Explainable AI

In response to the increasing demand for interpretability, a vast literature on this matter has been produced [18]. Many interpretability approaches related to sequential data modeling focus on adding an *attention mechanism* [2, 44] to a sequential model and use the attention weights as a form of explanation [3, 11, 33, 47], however recent works have highlighted how this kind of explanation might lack consistency [6, 22, 41] and that attention should not be used as an explanation. Other interpretability approaches related to sequential data modeling focus on understanding the internal behavior of the black-box under study [45]; conversely, our approach is agnostic to the black-box whose outcome we want to explain. The agnostic approach to explanations was first introduced in LIME [39]. The intuition behind LIME is that even if the decision boundary learned by the black-box in the feature space can be arbitrarily complex, it can always be locally approximated by a simpler, more interpretable model. In the LIME approach, the explanation is the set of weights of a sparse linear model. Other examples of agnostic approaches that focus on explaining the black-box behavior around a specific instance are SHAP [32] and ANCHORS [40]. The SHAP approach evaluates local features importance using a game theory approach, while ANCHORS formulate the problem using a multi-armed bandit approach. Our work shares some of the features of these approaches: for example, we mine our explanations using a perturbation-based strategy. However, unlike any other method, we also exploit the domain knowledge encoded in the ontology to generate relevant perturbations for the specific problem under study. By doing so, we increase the quality of the generated synthetic instances, which is of pivotal importance for the quality of the explanation provided. Similarly to LIME, we train a local surrogate model able to mimic the black-box behavior around the data point, and similarly to ANCHORS, our approach produces rule-based explanations. However, we mine our explanations from a multi-label decision tree [36], which allows us to deal with a complex output space, the multi-label one, in a simple, straightforward manner. Furthermore, none of the aforementioned approaches can be directly applied to a sequential input. In our work, we introduce a temporal encoding/decoding scheme that allows the user to visualize which events are the most relevant for the instance classification directly on the temporal sequence.

2.3 Ontology use in machine learning

In our work, we exploit the medical ontology of ICD-9 (*International Classification of Diseases*, Ninth Revision[2]) diagnosis codes to increase the fidelity performance of the interpretable model to the black-box. The increase in predictive performance, thanks to the infusion of knowledge in the learning procedure, was adopted in several other works. For example, in [10], the authors use an attention mechanism that leverages the medical ontology of ICD-9 to learn a code representation that combines the embeddings of its ontology ancestors. They then train this attention mechanism together with an RNN with GRU units to improve the classification performance of prediction of the predictive model. They show that the performance is increased by 10% with respect to a basic model that does not exploit the medical ontology. Furthermore, they show that the learned representation of medical codes aligns with medical knowledge. Moreover, the authors of [37] show how disease classification performance can improve using features based on the ICD-9 codes semantic similarity. To compute the ontological similarity among sets of ICD-9 codes, i.e., a visit, they first calculate the semantic similarity of each pair of terms in the sequences as the *importance* of their lowest common ancestor in the hierarchy

[2] http://icd9.chrisendres.com/

and then take the maximum of these similarities as the similarity of the two sequences. This approach over-estimates the similarity of the two sequences since it is sufficient to have one ICD-9 code in common to have similarity equal to one. The *importance* of the lowest common ancestor is related to the level of the term in the hierarchy; according to the authors this feature is related to the rarity of the disease, but it just captures how well specified is the disease. However, even with this basic approach to encoding medical knowledge into the learning process, the performance of the algorithms is increased. We use a more sophisticated approach to compute patients similarity as detailed in Section 3.2.

3 METHODS

In this section, we introduce the components of *Doctor XAI* and how they form the full explanation pipeline. Our technique is based on the idea presented in [39] of learning an interpretable classifier able to mimic the decision boundary of the black-box that is relevant to the decision taken for a particular instance. More formally:

Given an instance x and its black-box outcome $b(x) = y$, an explanation is extracted for this individual decision from an inherently interpretable model c trained to mimic the local behavior of b.

For our approach, we follow the general pipeline of generating a set of synthetic instances (called *neighborhood*) surrounding the instance x we want to explain, labeling them utilizing the black-box b, training an interpretable model c on the labeled neighborhood, and finally extracting an explanation in the form of a symbolic rule. However, we have developed specific modules in order to deal with the temporal dimension in the data and exploit linked structural knowledge representation: Figure 1 illustrates our explanation pipeline.

3.1 The explanation pipeline

The starting point is the data point whose black-box prediction we are interested in explaining. As the first step, we select the data points that are closest to the instance to be explained in the available dataset: these points are called the *real neighbors* of the instance. We can either select the closest data points according to a standard distance metric, such as the Jaccard one or exploit ontology-base similarities. We describe the latter in Subsection 3.2. In both cases, we obtain a set of real neighbors, each of which is represented as a sequence. We then generate the synthetic neighborhood perturbing the first real neighbors to ensure the locality of the augmented neighborhood. The synthetic neighbors' sampling is crucial to the purpose of auditing black-box models. Ideally, the synthetic instances should be drawn from the true underlying local distribution. Unfortunately, this distribution is generally unknown, and how to generate meaningful synthetic patients is still an open question. While most state-of-the-art agnostic explainers employ random perturbations, we use the domain knowledge encoded in the ICD-9 ontology to generate more meaningful synthetic instances, as explained in Subsection 3.3. It could be argued that the interpretable model could be trained directly on the closest real neighbors. However, the rationale behind the generation of synthetic neighbors is that we want to build a dense training set for

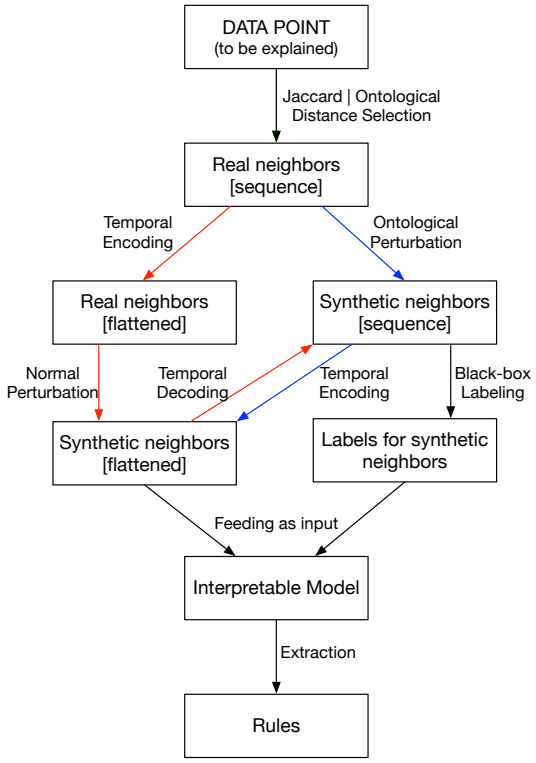


Figure 1: The explanation pipeline

the interpretable classifier c in order to increase its performance in mimicking the black-box. Unlike other explanation techniques, we do not perturb directly the features of the instance whose black-box decision we want to explain. By doing so, we prevent the case of generating a synthetic neighborhood containing only instances with the same black-box classification - a situation that would make the training of any interpretable model impossible. In other words, we ensure the *expressiveness* of the synthetic neighborhood, i.e., the black-box classifications are heterogeneous among the synthetic neighbors. For the perturbation steps in our pipeline, we can follow two alternative paths, represented by the red and blue arrows in Figure 1 (the two paths share the black arrows). The red path is based on the normal perturbation, which we describe in Subsection 3.4; the blue path involves the ontological perturbation, as described in Subsection 3.3. Both paths involve steps of temporal encoding/decoding (with the relative algorithms described in Subsection 3.5), since the black-box model requires a sequential input, whereas the interpretable one requires a tabular (flat) one. The red path is based on the normal perturbation: first, the real neighbors are encoded (flattened) into sparse vectors. Then the normal perturbation is applied in order to obtain a synthetic neighborhood - and this kind of data can be fed to an interpretable model. In order to obtain the labels for the synthetic data points, however, we have to decode them (back into sequences) so that we can feed them to our black-box model for labeling. Once we have both the synthetic neighborhood and the corresponding labels, we can train the interpretable model, and finally, extract symbolic rules. Similarly to [36],

we chose a multi-label decision tree as inherently interpretable classifier c. From such decision tree, we extract rule-based explanations in the form $p \rightarrow y$ where $y = c(x)$. The explanations are extracted by including in the rule premise p all the split conditions on the path from the root to the leaf node that is satisfied by the instance x. The blue path involves the ontological perturbation. In this case, we can apply the perturbation directly on sequential data, obtain a synthetic neighborhood as a set of sequences, and feed them to the black-box model for labeling. However, as it was for the red path, the interpretable model requires a tabular input, so we proceed to flatten (time-encode) the synthetic neighbors in a set of vectors. At this point, the blue path follows the same final steps as described above: training of the interpretable model and extraction of symbolic rules. We remark that, while we followed a general framework for our model-agnostic explanation pipeline, we have extended the framework with novel contributions in order to deal with structured data and sequential data respectively. We observe that these components can be independently plugged in an explanation pipeline according to the nature of the data point to be explained.

3.2 Ontological Distances

In this section, we define a new distance measure that allows us to select the first neighbors of the instance whose decision we want to explain. Each patient's clinical history is represented as a list of visits, which in turn are encoded as lists of ICD-9 codes. Every instance is therefore a list of lists of ICD-9 codes. More formally, if we define the set of ICD-9 codes as $C = \{c_1, c_2, \ldots, c_{|C|}\}$, each patient's clinical history is represented by a sequence of visits $V_1, \ldots, V_M$ such that $V_i \subseteq C$. A simple example of a patient clinical history representation is as follows:

$$[[433.10, 453.81], [453.81], [453.81, 788.5, 790.01]]$$

The patient visited the hospital three times; the condition 453.81 (Acute embolism and thrombosis of superficial veins of unspecified upper extremity) is chronic, condition 433.10 (Occlusion and stenosis of carotid artery without mention of cerebral infarction) was observed on the first visit only, whereas two new conditions (with codes 788.5 and 790.01) were diagnosed only in the third visit. We observe that multi-hot encoding all occurring ICD-9 codes is a fairly inefficient representation for visits - the obvious drawback being the size of the encoding vector corresponding to the size of the ICD-9 dictionary. Furthermore, this positional representation does not encode the semantic distance from ICD-9 codes - a patient with food poisoning, one with a broken hand and one with a broken wrist are equally distant from a purely Hamming-based perspective. In order to mine the semantically similar data points, we introduce an ontology-based distance metric.

Code-to-code similarity Each ICD-9 code represents a medical concept in a hierarchical ontology, these concepts are the nodes of the graph-representation of such ontology, and it is therefore possible to compute distance and similarity scores among any pair of them. Several similarity metrics could be selected; in this paper, we adopt the Wu-Palmer similarity score (WuP) [46] because it is one of the most commonly used for ICD-9 ontologies [1, 16, 23].

Given two ICD-9 nodes c_1 and c_2, let L be their lowest common ancestor (LCA) and R be the root of the ICD-9 ontology; also let $d(x, y)$ be the number of hops (steps) required to reach node y from node x following the ontology links. The WuP similarity measure between c_1 and c_2 corresponds to:

$$WuP(c_1, c_2) = \frac{2 * d(L, R)}{d(c_1, L) + d(c_2, L) + 2 * d(L, R)}$$

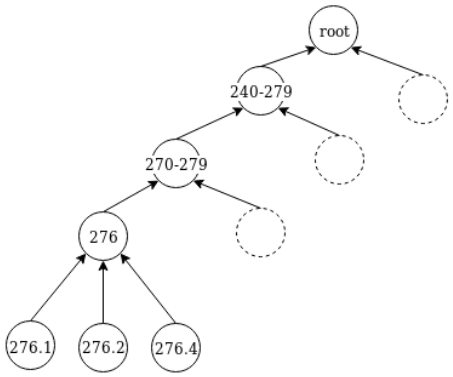
$WuP(c_1, c_2) \in [0, 1]$ for any couple of ICD-9 nodes. The lower bound 0 is obtained when $d(L, R) = 0$, that is, when the LCA of c_1 and c_2 is the root node. Conversely, a node has WuP-similarity 1 with itself. By relying on the underlying ICD-9 ontology, we can therefore use the WuP similarity to compute pairwise distances between ICD-9 codes. This yields a much more fine-grained analysis compared to a coarse Hamming similarity.

Visit-to-visit distance Having defined a code-to-code distance, the following step is to compute distances at the visit level - since visits are defined as lists of occurring ICD-9 codes. We adopted the weighted Levenshtein [28] distance, a string metric for measuring the difference between two sequences as the minimum number of single-character edits (insertions, deletions or substitutions) required to change one sequence into the other. The weighted version of the Levenshtein distance allows defining custom insertion/deletion/edit costs. We have set $1 - WuP(c_1, c_2)$ as edit cost for modifying c_1 into c_2, and 1 as insertion/deletion (indel) cost (since $WuP(c_1, c_2) \geq 0$, $1 - WUP(c_1, c_2) \leq 1$) in order to favor edits over indels. This gives us a distance metric between pairs of visits, which is based on the similarity between the ICD-9 codes occurring in each of the two visits.

Patient-to-patient distance The third step is to compute a patient-to-patient distance metric based on how similar the visits of the two patients are. In order to do so, we adopted the Dynamic Time Warping (DTW) algorithm [4], again using the pairwise visit distances provided by the weighted Levenshtein algorithm as edit distance. The sequences of visits are *warped* non-linearly in the time dimension to determine a measure of their similarity independent of certain non-linear variations in the time dimension. This final step provides us with the pairwise distances for all patients (data points) in the dataset, thus enabling us to select real neighbors with ontologically similar conditions w.r.t. the data point to explain.

3.3 Ontological Perturbation

As previously mentioned, after selecting the first real neighbors of the instance whose decision we want to explain, we perturb them in order to generate synthetic neighbors. There are mainly two ways to perform an ontology-based perturbation on an instance: by masking or replacing some conditions (ICD-9 codes) in the patient's clinical history according to their relationships in the ontology. We decided to adopt the first type of perturbation in order to limit the amount of noise injected in the training set of the interpretable classifier. The idea behind perturbing the patient's history in this way is that we want to explore how the black-box label changes if we mask all the semantically-similar items from the sequence. We decided to randomly mask all the occurrences of the items with the same least common superconcept. By doing so, we are exploring how a



ICD-9	description
root	root
240-279	Endocrine, Nutritional And Metabolic Diseases, And Immunity Disorders
270-279	Other Metabolic Disorders And Immunity Disorders
276	Disorders of fluid electrolyte and acid-base balance
276.1	Hyposmolality and/or hyponatremia
276.2	Acidosis
276.4	Mixed acid-base balance disorder

Figure 2: A branch of the ICD-9 ontology

general condition (a higher concept in the ontology) is affecting the black-box diagnosis. In our case, we are dealing with patients' clinical history. Each patient's clinical history is a sequence of visits, and each visit is represented by lists of ICD-9 codes. In the ICD-9 ontology, all codes are composed of a prefix and a suffix, separated by a dot: the prefix defines the general condition, and the suffix provides increasingly specific information. We show an example of the hierarchical structure of the ICD-9 ontology in Figure 2. Our implementation of the ontological perturbation is the following: we first randomly select one ICD-9 code in the clinical history of the patient we want to perturb (a leaf of the ontology), then we mask all the ICD-9 codes in the patient's history that share the same prefix (the least common superconcept). By doing so, we generate synthetic patients that lack a specific group of semantically similar conditions. Consider, for example, the following patient:

$$P = [[276.1, 276.2], [276.4, 530.1], [507, 530], [276.2, 530.19]]$$

One example of ontological perturbation is the following: we randomly select ICD-9 code 276.4 which is *mixed acid-base balance disorder*. Starting from this code we create the synthetic patient

$$P^* = [[], [530.1], [507, 530], [530.19]]$$

by masking all the ICD-9 codes related to ICD-9 276, i.e., *disorders of fluid electrolyte and acid-base balance* (the least common superconcept). Note that, without ontological information, we have 7 different codes and therefore 2^7 potential perturbations, most of which don't really isolate different conditions. Conversely, using the ontology we group the occurring ICD-9 codes in three categories {276*, 507*, 530*}: as a consequence we have 8 potential maskings, each of which isolates a subset of different conditions.

3.4 Normal Perturbation

As an alternative to the ontological perturbation of the first real neighbors of the instance under study, we performed a *normal perturbation* on such features. This perturbation applies to a broader

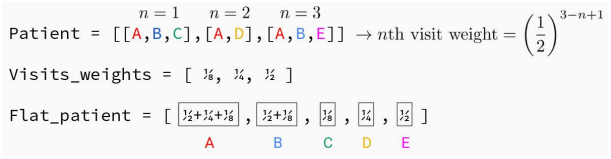


Figure 3: Example of temporal encoding for a patient

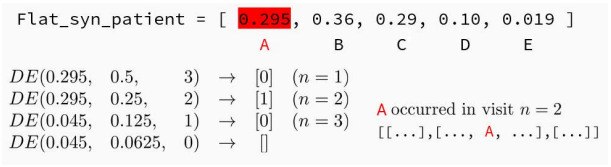


Figure 4: Example of temporal decoding for a patient

number of cases since it does not require an ontology to be performed. Given the *flattened* version of the real neighbors, the normal perturbation creates the new synthetic instances feature by feature drawing from a normal distribution with mean and standard deviation of the empirical distribution of that feature in the real neighbors. This perturbation implies the strong assumption that every feature is independent of the others.

3.5 Temporal encoding and decoding

As introduced above, the standard data type for longitudinal healthcare data is to represent a patient as a list of visits, and in turn each visit as a list of occurring conditions (in our case, ICD-9 codes). There is no inherently interpretable model able to deal with the multi-label classification of such type of input; therefore, we need to perform an input transformation that both retains its sequential information and allows to feed it into an interpretable model - a decision tree in our case. We introduce a pair of encoding-decoding algorithms so that we can *flatten* the temporal dimension when feeding our synthetic neighborhood to the interpretable model. The binary encoder implements a time-based exponential decay rooted at the last item of the sequence. Intuitively, each code c_i in visit V_j will be given a score of +.5 if V_j is the last visit, +.25 if V_j is the second-to-last visit, and so on. More formally, when encoding a patient $P = [V_1, .., V_N]$, each code $c \in P$ will be encoded as follows:

$$EN(c, P) = \sum_{i=1}^{n} (1/2^{n-i+1} \text{ if } c \in V_i \text{ else } 0)$$

The encoding is 0 for all items that never occur in that sequence, and it tends to 1 for a growing number of elements in the sequence in which that item occurs. The encoded (flattened) representation of a patient is therefore a sparse vector of real numbers, and as such it can be fed to multiple interpretable models.
Conversely, we define the decoding from a sparse vector of real numbers to a sequence of visits as:

$$DE(X, t, l) = \begin{cases} [] & \text{if } X = 0 \text{ or } l = 0 \\ append(DE(X - t, t/2, l - 1), [1]) & \text{if } X > t \\ append(DE(X, t/2, l - 1), [0]) & \text{otherwise} \end{cases}$$

where X is the value to be decoded, t is initially set at .5 and l controls the maximum length of the generated sequence (we use the average length of the real neighbors). The result of the decoding is a list of 0s and 1s that indicates the presence/absence of a certain code. We show a simple example of temporal encoding in Figure 3. In this example, the patient visited the hospital three times. Each visit contains a set of ICD-9 codes (for the sake of simplicity here represented as letters). As a first step, a weight is associated to each visit. Then the weight of each ICD-9 code is computed by adding the weights of the visits where it occurred. We also show a simple example of temporal decoding of a flat synthetic patient in Figure 4. In this example, we transform the value of the first ICD-9 code (represented by letter A) into its occurrence in the sequence. In this example we set the maximum length of the generated sequence to $l = 3$. It is important to remark that the decoding algorithm, when presented with perturbed data, might potentially produce arbitrarily long sequences, where progressively small residuals are mapped to the occurrence of the decoded ICD-9 code in progressively further away visits. The l-guard was introduced to prevent this from happening so that flattened synthetic patients match the number of visits of the flattened real neighbors.

4 EXPERIMENTS AND RESULTS

4.1 Dataset

We ran our experiments on the *Multiparameter Intelligent Monitoring in Intensive Care III* (MIMIC-III) database [24]. This database contains de-identified data of over 40.000 ICU (Intensive Care Unit) patients of the Beth Israel Deaconess Medical Center data in Boston collected from 2001 to 2012. We used the information related to the hospital stay (dates and diagnosis codes) to build the patient clinical history as performed by the pre-processing script available in *Doctor AI* GitHub repository[3]. This operation removes all patients with less than two visits, some statistics about the dataset after the pre-processing procedure can be found in Table 1.

	MIMIC-III
n. of patients	7499
n. of visits	19911
avg. n. of visits per patient	2.65
min. n. of visits per patient	2
max. n. of visits per patient	42
n. of unique ICD-9 codes	4880
n. of unique CSS grouper codes	272
avg. n. of ICD-9 codes per visit	13.06

Table 1: MIMIC-III characteristics for patients with more than one visit

The clinical history of each patient is modeled as time-stamped sequence of visits. As previously mentioned, each visit is represented by a set of ICD-9 diagnosis codes, these codes are assigned to each patient at the end of his or her hospital stay, and hospitals use them to bill for care provided. They are organized in a "is-a" hierarchical tree structure[4] that places more general concepts closer to the root of the tree and more fine-grained concepts closer to the leaves of the tree. The ICD-9 taxonomy and occurring ICD-9 codes in MIMIC are visualized in Figure 5. We used this ontology to measure the similarity between patients' clinical history as described in section 3.2 and to generate the synthetic neighbors of each patient as described in section 3.3.

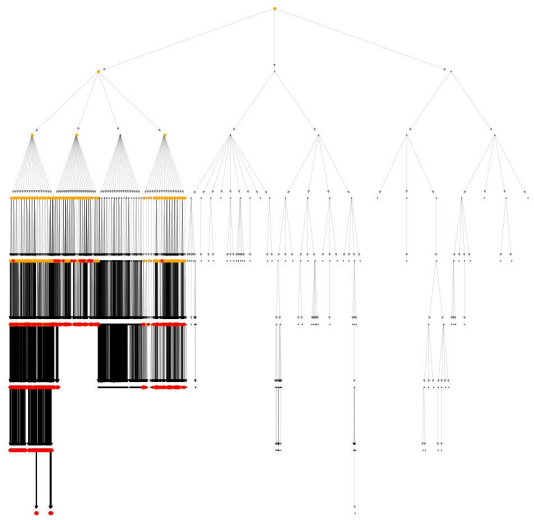


Figure 5: ICD-9 ontology. The red dots represent codes occurring in the MIMIC dataset, the orange ones their parent nodes.

4.2 Black-box classifier

We trained *Doctor AI* on *MIMIC-III* for 50 epochs, using approximately 70% of patients as the training set, 15% as the validation and 15% as the test set. We built the label for each time step of the sequence by grouping the full-length ICD-9 codes using CCS single-digit groupers[5]. By doing so, the dimensionality of the label space shrinks from 4880 codes to 272 groups of codes. We compare the predictive performance of *Doctor AI* trained by us on MIMIC-III dataset with the ones reported in the original paper in Table 2. The metric used to evaluate the predictive performance is *recall@n*= $\frac{\text{\# of true positives in the top n predictions}}{\text{\# of true positives}}$. We also trained a baseline model to imitate one of the benchmarks of the original paper. This baseline, the *Most frequent*, predicts the top-k most frequent labels observed in visits before the current visit. The fact that we trained *Doctor AI* on a much smaller dataset lowers the algorithm's predictive performance compared to the ones of the original paper. However, they are in line with the performance on the MIMIC-II dataset discussed in the original paper. Furthermore, having a good predictive performance is not our goal; we will use the black-box labels as ground-truth labels for the decision tree. In our work, we focus on explaining *Doctor AI* because of the availability of its source code and because the authors' results are easily

[3]https://github.com/mp2893/doctorai

[4]https://bioportal.bioontology.org/ontologies/ICD9CM

[5]https://www.hcup-us.ahrq.gov/toolssoftware/ccs/ccs.jsp

Table 2: *Doctor AI* performance on different datasets.

Dataset and algorithm	recall@n		
	n=10	n=20	n=30
Doctor AI: MIMIC-III	0.350	0.521	0.631
Most frequent: MIMIC-III	0.383	0.473	0.491
Doctor AI: dataset from [9]	0.643	0.743	0.796
Most frequent: dataset from [9]	0.566	0.674	0.717

reproducible using open-source data. However, we want to stress that our method is not specific to this black-box.

4.3 Experimental set-up

We decided to test our explanation method on a cohort of 1.000 randomly selected patients from the MIMIC database. We put each of these 1.000 patients through 3 different explanation pipelines and we explained their top-10 CCS-codes prediction. The first two exploit the ontological information encoded into ICD-9 codes, whereas the last one can also be used to explain sequential data classification if an ontology is missing. We aim to show that exploiting the ontological information in the data increases the explanation quality.

- *Ontological pipeline with ontological perturbation - Dr.XAI.* This pipeline fully exploits the knowledge encoded into the ICD-9 ontology to create the synthetic neighborhood. Given a patient whose black-box decision we want to explain, it selects its first k neighbors in the dataset using the *ontological distance* described in section 3.2 and then it generates the synthetic neighborhood by perturbing them using the *ontological perturbations* described in section 3.3. This pipeline corresponds to the blue path of Figure 1 using the Ontological similarity.
- *Ontological pipeline with normal perturbation.* This pipeline selects the first k real neighbors of the instance to explain using the *ontological distance*, but then it creates the synthetic neighborhood by perturbing these instances using the *normal perturbation* described in section 3.4. This pipeline corresponds to the red path of Figure 1 using the Ontological similarity.
- *Non-ontological pipeline with normal perturbation.* This pipeline does not use the semantic information encoded in the ICD-9 codes. It first selects the k real neighbors of the instance to be explained using *Jaccard similarity* between each patient visit and then it perturbs them by using *normal perturbations* 3.4. This pipeline corresponds to the red path of Figure 1 using the Jaccard similarity.

By comparing the two ontological pipelines, we want to show that exploiting the semantic information encoded in the ICD-9 ontology is also useful to create the synthetic neighbors. We developed the *non-ontological pipeline* as a baseline for explanation quality. However, this last pipeline is also the most general one because it can be applied to sequential data that does not have an associated ontology. Furthermore, we wanted to show that increasing the density of the feature space around the instance to be explained by creating the synthetic neighbors actually increases the interpretable model's ability to mimic the black-box locally. For this reason, for each instance to be explained, we trained two decision trees. One decision tree is trained directly on the real neighbors of that patient from the dataset, and the other one is trained on a fraction of the augmented synthetic neighborhood. We then compare the performance of these decision trees on an out-of-sample set of synthetic neighbors.

We utilize the following metrics to evaluate and compare the different explanation pipelines.

- *Fidelity to the black-box* $\in [0, 1]$ This metric compares the predictions made by the interpretable model with the predictions made by the black-box on a synthetic neighborhood of the instance. It measures the ability of the interpretable classifier to locally mimic the black-box, and therefore it is tested on a held-out subset of the synthetic neighborhood. Since we are dealing with a multi-label classification task, we calculate the fidelity the F_1 measure with micro-averaging [50].
- *Hit* $\in [0, 1]$ This metric compares the interpretable classifier prediction y_c and the black-box prediction y_b on the instance to be explained. It tells us if the interpretable classifier predicts the same label as the black-box on the instance we want to explain. Since the prediction we are trying to explain is a multi-label classification, we calculate the hit as $1 - \text{hamming-distance}(y_b, y_c)$.
- *Explanation complexity.* This metric measures the complexity of the explanation as the number of premises in the rule-based explanation. This measure is important since we do not want to approximate the black-box with a model that loses its interpretability because of the high-dimensionality of the explanations it produces [13, 30].

4.4 Results

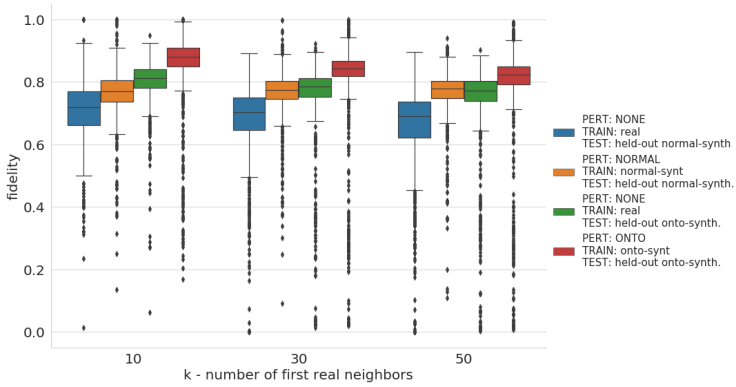


Figure 6: Fidelity distribution for the ontological pipeline with different k, perturbation type, and training/test set.

In Figure 6 we show the fidelity sample distributions at different values of k for the decision trees trained using the *ontological explanation pipelines*, i.e., the pipelines that select the first k dataset

Table 3: Mean values of fidelity

Explanation Pipeline	Fidelity					
	k=10		k=30		k=50	
	realDT	syntDT	realDT	syntDT	realDT	synDT
Ontological pipeline with ontological perturbation	0.81	0.89	0.77	0.85	0.12	0.79
Ontological pipeline with normal perturbation	0.70	0.73	0.67	0.62	0.10	0.76
Non-ontological pipeline with normal perturbation	0.71	0.77	0.69	0.47	0.68	0.78

Table 4: Mean values of hit

Explanation Pipeline	Hit					
	k=10		k=30		k=50	
	realDT	syntDT	realDT	syntDT	realDT	synDT
Ontological pipeline with ontological perturbation	1.00	1.00	1.00	1.00	0.93	1.00
Ontological pipeline with normal perturbation	1.00	0.98	1.00	0.99	0.93	0.98
Non-ontological pipeline with normal perturbation	1.00	0.99	1.00	0.99	1.00	0.99

neighbors of the instance to be explained using the *ontological distance*. The first observation is that the decision trees trained directly on the k real neighbors (blue and green boxplots) generally have a lower fidelity to the black-box compared to the ones trained on the augmented synthetic neighborhood (orange and red boxplots). This trend is true for all values of k and for both the *ontological pipeline with ontological perturbation* and the *ontological pipeline with normal perturbation*. The fidelity values of each decision tree have been evaluated on an held-out test set of synthetic neighbors. This trend confirms that increasing the local density of points in the feature space around the instance to be explained helps the interpretable model to understand the black-box behavior. The second observation is that the fidelity of the decision tree trained using the *ontological pipeline with ontological perturbation* (red boxplot) is generally higher compared to all the other explanation pipelines. This observed tendency confirms that exploiting the ontological information during the synthetic neighborhood creation allows the decision tree to better approximate the local black-box decision boundary.

In Figure 7 we show the fidelity sample distributions at different values of k for the decision trees trained using the *non-ontological explanation pipeline*, i.e., the pipeline that selects the first k dataset neighbors of the instance to be explained using the *Jaccard similarity* between patients' visits. We developed this explanation pipeline that does not use the semantic information encoded into the ICD-9 codes as a baseline to prove that an approach that does not exploit this information has lower performance. This is true if we compare this explanation pipeline with the fully-ontological one (the *ontological pipeline with ontological perturbation*). However, the fidelity performance of this non-ontological pipeline is comparable to the ones of the *ontological pipeline with normal perturbation*. The high values of fidelity achieved by this pipeline prove that we developed a *trustable* explainability technique applicable to any black-box that takes as input any sequential data, even when there is no ontology associated with the items of the sequence. Furthermore, it is important to notice that, also for this pipeline, the values of fidelity to the black-box increase after the synthetic neighborhood augmentation (the orange boxplot).

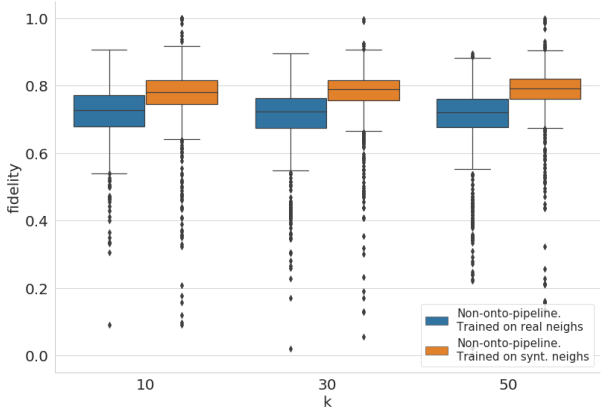


Figure 7: Fidelity distribution for the non-ontological pipeline at different values of k and training set.

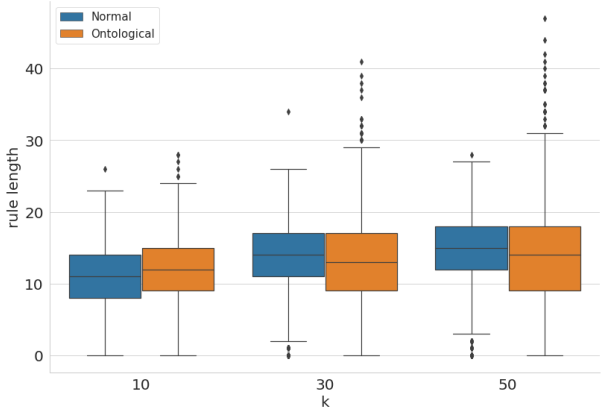


Figure 8: Explanation complexity for the ontological pipelines

In Figure 8 we show the sample distribution of *explanation complexity*, i.e., the number of premises in the rule-based explanations at different values of k for the two ontological explanation pipelines. As expected, we see how the length of the explanation increases as k increases. This happens because if we start from a high number of first real dataset neighbors we are trying to approximate a larger portion of the decision boundary of the black-box with the interpretable classifier. We could say that we are not restricting ourselves to the *local* decision boundary close to the instance whose decision we want to explain. Therefore, since we are trying to approximate a more complex decision boundary the dimensionality/complexity of the decision tree grows and consequentially the length of the rule increases. From this plot it is also possible to see that the explanation length of the explanations extracted from the *ontological pipeline with ontological perturbation* (orange boxplot) is more variable than the ones extracted using the *ontological pipeline with normal perturbation* for large values of k. Aggregated statistics of fidelity and of hit for all the explanation pipelines are shown in Tables 3 and 4: we can observe that the value of hit is consistently high for all explanation pipelines and across all values of k.

4.5 Explanation example

We show in Figure 9 an explanation example extracted with the *ontological pipeline with ontological perturbation* with $k = 10$. In order to make it more comprehensible for readers not familiar with ICD-9 codes, we enriched the rule-based explanation with the ICD-9 codes semantic. The original decision rule extracted from the decision tree can be seen at the top of the figure with the fidelity of the decision tree and its hit value. There are several ways to read this rule since it contains many layers of information. The decision rule is the decision tree pathway that leads from the root of the tree to the leaf containing the black-box decision; for this reason, all inequalities are to be considered in conjunction - furthermore, the ICD-9 codes occurring in the rule are ranked in order of information gain. Each conjunct of the rule follows the pattern

$$\text{ICD-9\_code} = observed\_value \gtrless threshold\_value$$

The *observed value* is the value of that ICD-9 code for the patient whose decision we want to explain. Recall that the temporal encoding or *flattening* procedure described in Section 3.5 assigns to each ICD-9 code a weight according to the visit in which it was observed (diagnosed). The *threshold value* is the split value assigned by the decision tree to that ICD-9 code. Both these values can be interpreted as the presence of the ICD-9 code in a set of visits. The patient under examination had four visits. The ICD-9 codes describing the diagnoses associated with each visit are represented in the timeline just below the decision rule. Recall that we are explaining the top-10 CCS-codes predicted by *Doctor AI*. The ICD-9 codes considered meaningful by the black-box have been colored to enhance the readability. The explanation of each real and threshold value can be found in the list below the timeline. For example, the ICD-9 code 584.5 has an observed value of 0.25, which means that it was observed in the second-to-last visit (visit 3). Its threshold value is 0.12, whose closest value among those generated in the temporal encoding process is 0.125 which represents the third-to-last visit (visit 2). For this reason, even if this ICD-9 code was observed in the penultimate visit, the interpretation of the first rule conjunct is *584.5 has to have been observed at least once in the last three visits*.

The code to run our experiments as well as our results are available on GitHub[6].

5 CONCLUSIONS AND FUTURE WORK

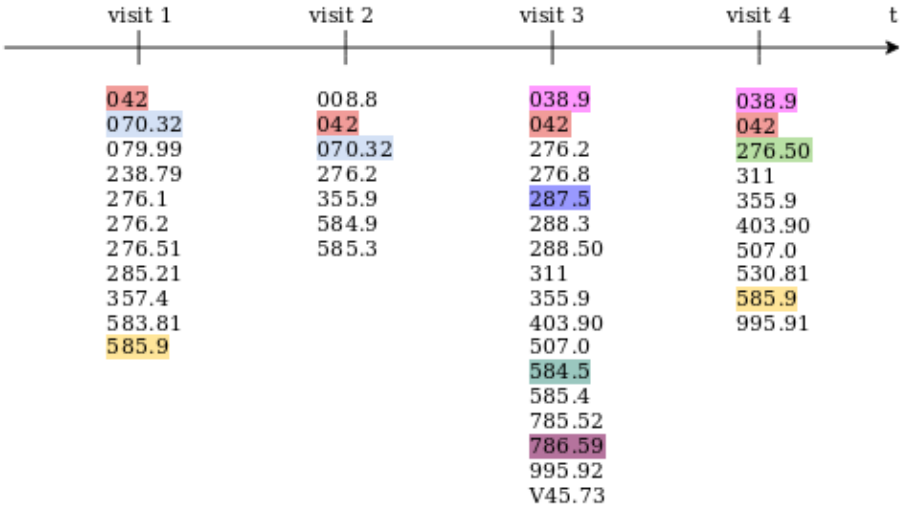
In this paper, we presented **Doctor XAI**, the first agnostic explanation technique suitable for any black-box classifier that deals with data having one or more of these characteristics: sequential, multi-labeled, and ontology-linked. These features are typical of healthcare data. Our technique first generates a set of synthetic instances close to the instance whose black-box decision we need to explain, then it trains an interpretable classifier - a decision tree - on such neighborhood, and finally, it extracts a rule-based explanation from it. We studied the behavior of the interpretable classifier varying the hyper-parameter k - the number of first neighbors in the real dataset that are considered in the synthetic neighborhood generation. In particular, we showed that, for all values of k, the synthetic neighborhood generation procedure which exploits the ontological information encoded in the ICD-9 codes achieves better performance in approximating the local behavior of the black-box if compared to a procedure which does not have access to the ontology. Furthermore, the synthetic augmentation of the interpretable classifier training set allows it to increase its fidelity to the black-box. We also tested the sequential-only version of our explanation technique showing that it achieves good fidelity to the black-box, while also confirming that the ontology-enriched approach achieves a better score.

Application scenario. We believe that doctors and patients would both benefit from such an explanation of the black-box behavior. Ideally, doctors (the target users of our method) would be able to have a higher understanding of the decision support system they are using (the black-box). This means that the ultimate decision would be more informed and ultimately better than the decision that the human decision-maker would have made without the black-box, as well as better than the automated decision by the black-box alone. Such an informed decision would also benefit the patient because of the increased quality of care provided by the doctor. In this paper we focused on the medical domain, but since our method is agnostic w.r.t. the black-box, the possible applications cover several scenarios where we can identify a sequence of events linked to ontology concepts, such as online market basket analysis [19] or Wikipedia user behavior prediction [27]

Concerning directions for future work. We will focus on studying other kinds of synthetic neighbors generation for sequential data. Furthermore, we would like to better assess the impact of the random components of the synthetic neighbors' generation procedure on the quality of the explanations. Right now, *Doctor XAI* can explain only black-box classifiers, but with a simple extension, we would be able to explain black-box regressors producing continuous outcomes - this is another common healthcare task, for instance for predicting risk stratification.

[6] https://github.com/CeciPani/DrXAI

decision rule : {584.5 = 0.25 > 0.12, 070.32 = 0.1875 > 0.09, 042 = 0.9375 > 0.47, 287.5 = 0.25 > 0.12, 585.9 = 0.5625 > 0.28 , 276.50 = 0.5 > 0.25, 038.9 = 0.75 > 0.38, 786.59 = 0.25 > 0.12} → [259, 237, 2, 157, 59, 55, 158, 3, 5, 6]
fidelity = 0.89
hit = 0.99



584.5 = 0.25 > 0.12
code 584.5: "**Acute kidney failure with lesion of tubular necrosis**"
was observed in visit 3 and in particular, it was observed after visit 2

070.32 = 0.1875 > 0.09
code 070.32: "**Chronic viral hepatitis B without mention of hepatic coma without mention of hepatitis delta**"
was observed in visits [1 2], and in particular, it was observed in visit 2

042 = 0.9375 > 0.47
code 042: "**Human immunodeficiency virus [HIV] disease**"
was observed in visits [1 2 3 4], and in particular, it was observed in the last visit

287.5 = 0.25 > 0.12
code 287.5: "**Thrombocytopenia, unspecified**"
was observed in visit 3, and in particular, it was observed after visit 2

585.9 = 0.5625 > 0.28
code 585.9: "**Chronic kidney disease, unspecified**"
was observed in visits [1 4], and in particular, it was observed after visit 3

276.50 = 0.5 > 0.25
code 276.50: "**Volume depletion, unspecified**"
was observed in visit 4, and in particular, it was observed after visit 3

038.9 = 0.75 > 0.38
code 038.9: "**Unspecified septicemia**"
was observed in visits [3 4], and in particular, it was observed in visit 3

786.59 = 0.25 > 0.12
code 786.59: "**Other chest pain**"
was observed in visit 3, and in particular, it was observed after visit 2

Figure 9: Explanation example

REFERENCES

[1] Ahmad Fayez S Althobaiti. 2017. Comparison of Ontology-Based Semantic-Similarity Measures in the Biomedical Text. *Journal of Computer and Communications* 5, 02 (2017), 17.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[3] Tian Bai, Shanshan Zhang, Brian L Egleston, and Slobodan Vucetic. 2018. Interpretable representation learning for healthcare via capturing disease progression through time. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 43–51.

[4] Donald J. Berndt and James Clifford. 1994. Using Dynamic Time Warping to Find Patterns in Time Series. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining (AAAIWS'94)*. AAAI Press, 359–370. http://dl.acm.org/citation.cfm?id=3000850.3000887

[5] Olivier Bodenreider. 2004. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic acids research* 32, suppl_1 (2004).

[6] Gino Brunner, Yang Liu, Damián Pascual, Oliver Richter, and Roger Wattenhofer. 2019. On the Validity of Self-Attention as Explanation in Transformer Models. *arXiv preprint arXiv:1908.04211* (2019).

[7] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. 2015. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1721–1730.

[8] Zhengping Che, David Kale, Wenzhe Li, Mohammad Taha Bahadori, and Yan Liu. 2015. Deep computational phenotyping. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM.

[9] Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F Stewart, and Jimeng Sun. 2016. Doctor ai: Predicting clinical events via recurrent neural networks. In *Machine Learning for Healthcare Conference*. 301–318.

[10] Edward Choi, Mohammad Taha Bahadori, Le Song, Walter F Stewart, and Jimeng Sun. 2017. GRAM: graph-based attention model for healthcare representation learning. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 787–795.

[11] Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter Stewart. 2016. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. In *Advances in Neural Information Processing Systems*. 3504–3512.

[12] Edward Choi, Andy Schuetz, Walter F Stewart, and Jimeng Sun. 2016. Medical concept representation learning from electronic health records and its application on heart failure prediction. *arXiv preprint arXiv:1602.03686* (2016).

[13] Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608* (2017).

[14] Wenjuan Fan, Jingnan Liu, Shuwan Zhu, and Panos M Pardalos. 2018. Investigating the impacting factors for the healthcare professionals to adopt artificial intelligence-based medical diagnosis support system (AIMDSS). *Annals of Operations Research* (2018), 1–26.

[15] Samuel G Finlayson, John D Bowers, Joichi Ito, Jonathan L Zittrain, Andrew L Beam, and Isaac S Kohane. 2019. Adversarial attacks on medical machine learning. *Science* 363, 6433 (2019), 1287–1289.

[16] Dominic Girardi, Sandra Wartner, Gerhard Halmerbauer, Margit Ehrenmüller, Hilda Kosorus, and Stephan Dreiseitl. 2016. Using concept hierarchies to improve calculation of patient similarity. *Journal of biomedical informatics* 63 (2016).

[17] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Dino Pedreschi, Franco Turini, and Fosca Giannotti. 2018. Local Rule-Based Explanations of Black Box Decision Systems. *arXiv preprint arXiv:1805.10820* (2018).

[18] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Dino Pedreschi, and Fosca Giannotti. 2018. A Survey Of Methods For Explaining Black Box Models. *ACM CSUR* 51, 5, Article 93 (Aug. 2018), 42 pages.

[19] Riccardo Guidotti, Giulio Rossetti, Luca Pappalardo, Fosca Giannotti, and Dino Pedreschi. 2017. Market basket prediction using user-centric temporal annotated recurring sequences. In *2017 IEEE International Conference on Data Mining (ICDM)*. IEEE, 895–900.

[20] M Hutson. 2018. AI researchers allege that machine learning is alchemy. *Science* 360, 6388 (2018), 861.

[21] Abhyuday N Jagannatha and Hong Yu. 2016. Bidirectional RNN for medical event detection in electronic health records. In *Proceedings of the conference. Association for Computational Linguistics. North American Chapter. Meeting*, Vol. 2016. NIH Public Access, 473.

[22] Sarthak Jain and Byron C Wallace. 2019. Attention is not explanation. *arXiv preprint arXiv:1902.10186* (2019).

[23] Zheng Jia, Xudong Lu, Huilong Duan, and Haomin Li. 2019. Using the distance between sets of hierarchical taxonomic clinical concepts to measure patient similarity. *BMC Medical Informatics and Decision Making* 19, 1 (2019), 91. https://doi.org/10.1186/s12911-019-0807-y

[24] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. MIMIC-III, a freely accessible critical care database. *Scientific data* 3 (2016), 160035.

[25] Thomas A Lasko, Joshua C Denny, and Mia A Levy. 2013. Computational phenotype discovery using unsupervised feature learning over noisy, sparse, and irregular clinical data. *PloS one* 8, 6 (2013), e66341.

[26] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436.

[27] Janette Lehmann, Claudia Müller-Birn, David Laniado, Mounia Lalmas, and Andreas Kaltenbrunner. 2014. Reader preferences and behavior on Wikipedia. In *Proceedings of the 25th ACM conference on Hypertext and social media*. ACM.

[28] Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, Vol. 10. 707–710.

[29] Znaonui Liang, Gang Zhang, Jimmy Xiangji Huang, and Qmming Vivian Hu. 2014. Deep learning for healthcare decision making with EMRs. In *Bioinformatics and Biomedicine (BIBM), 2014 IEEE International Conference on*. IEEE, 556–559.

[30] Zachary C Lipton. 2016. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490* (2016).

[31] Zachary C Lipton, David C Kale, Charles Elkan, and Randall Wetzel. 2015. Learning to diagnose with LSTM recurrent neural networks. *arXiv preprint arXiv:1511.03677* (2015).

[32] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*. 4765–4774.

[33] Fenglong Ma, Radha Chitta, Jing Zhou, Quanzeng You, Tong Sun, and Jing Gao. 2017. Dipole: Diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 1903–1911.

[34] Gianclaudio Malgieri and Giovanni Comandè. 2017. Why a Right to Legibility of Automated Decision-Making Exists in the General Data Protection Regulation. *International Data Privacy Law* 7, 4 (2017). https://doi.org/10.1093/idpl/ipx019

[35] Riccardo Miotto, Li Li, Brian A Kidd, and Joel T Dudley. 2016. Deep patient: An unsupervised representation to predict the future of patients from the electronic health records. *Scientific reports* 6 (2016).

[36] Cecilia Panigutti, Riccardo Guidotti, Anna Monreale, and Dino Pedreschi. 2019. Explaining multi-label black-box classifiers for health applications. In *International Workshop on Health Intelligence*. Springer, 97–110.

[37] Mihail Popescu and Mohammad Khalilia. 2011. Improving disease prediction using ICD-9 ontological features. In *2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)*. IEEE, 1805–1809.

[38] Alvin Rajkomar et al. 2018. Scalable and accurate deep learning with electronic health records. *npj Digital Medicine* 1, 1 (2018), 18.

[39] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 1135–1144.

[40] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-precision model-agnostic explanations. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[41] Sofia Serrano and Noah A Smith. 2019. Is Attention Interpretable? *arXiv preprint arXiv:1906.03731* (2019).

[42] Benjamin Shickel, Patrick James Tighe, Azra Bihorac, and Parisa Rashidi. 2017. Deep EHR: a survey of recent advances in deep learning techniques for electronic health record (EHR) analysis. *IEEE journal of biomedical and health informatics* 22, 5 (2017), 1589–1604.

[43] Barry Smith, Michael Ashburner, Cornelius Rosse, Jonathan Bard, William Bug, Werner Ceusters, Louis J Goldberg, Karen Eilbeck, Amelia Ireland, Christopher J Mungall, et al. 2007. The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nature biotechnology* 25, 11 (2007), 1251.

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[45] Scott Wisdom, Thomas Powers, James Pitton, and Les Atlas. 2016. Interpretable recurrent neural networks using sequential sparse recovery. *arXiv preprint arXiv:1611.07252* (2016).

[46] Zhibiao Wu and Martha Palmer. 1994. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 133–138.

[47] Yanbo Xu, Siddharth Biswal, Shriprasad R Deshpande, Kevin O Maher, and Jimeng Sun. 2018. Raim: Recurrent attentive and intensive model of multimodal patient monitoring data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2565–2573.

[48] Pranjul Yadav, Michael Steinbach, Vipin Kumar, and Gyorgy Simon. 2018. Mining electronic health records (EHRs): a survey. *ACM Computing Surveys (CSUR)* 50, 6 (2018), 85.

[49] Kun-Hsing Yu, Andrew L Beam, and Isaac S Kohane. 2018. Artificial intelligence in healthcare. *Nature biomedical engineering* 2, 10 (2018), 719.

[50] Min-Ling Zhang and Zhi-Hua Zhou. 2014. A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering* 26, 8 (2014).