Journal of Computer Science

Original Research Paper

Predictive Modeling Applied to Structured Clinical Data Extracted from Electronic Health Records: An Architectural Hypothesis and A First Experiment

[1]Alessandra Pieroni, [2]Alessandro Cabroni, [2]Francesca Fallucchi and [3]Noemi Scarpato

[1]*Agency for Digital Italy (AgID), 21 Liszt Street, 00144 Rome, Italy*
[2]*Guglielmo Marconi University, 44 Plinio Street, 00193 Rome, Italy*
[3]*San Raffaele Roma Open University, 247 Val Cannuta Street, 00166 Rome, Italy*

Article history
Received: 13-05-2021
Revised: 26-07-2021
Accepted: 03-08-2021

Corresponding Author:
Francesca Fallucchi,
Guglielmo Marconi
University, 44 Plinio Street,
00193 Rome, Italy
Email: f.fallucchi@unimarconi.it

Abstract: Predictive analysis is one of current important issues in the healthcare context. A lot of patients' input data can be obtained from their Electronic Health Records. In our research, we propose a general architecture named Health Prediction Architecture. Initially, we consider that data refer to strongly structured health datasets (no free text). Our objectives are related to exploring some problems in the prediction context for healthcare. In particular, we consider dataset heterogeneity, accuracy together explain ability, dataset for benchmarking. After a presentation of Electronic Health Record and some useful related standards, we propose our architecture based on two principal modules. First module produces features extraction and it implements a Convolutional Neural Network or alternatively a Multi-Layer Perceptron. Second module produces predictions and it implements alternatively one from Graph Convolutional Network, Simplified Graph Transduction Game, Nearest Nodes and Classes Graph. We define the datasets randomly so to have the possibility to manage data sufficiently heterogeneous and useful for a benchmarking, without any privacy problem too. In this study, we experiment a first instantiation of the architecture, based on Multi-Layer Perceptron as first module and Simplified Graph Transduction Game as second module, considering health data related to type 2 diabetes risk, generated according to a healthcare rule. We try the architecture by slightly increasing both cardinalities of datasets and extracted features. As first results of our research, in this study we produce training and testing randomized datasets and we obtain a testing accuracy behavior generally better than using only Multi-Layer Perceptron (best accuracy with 200 labelled elements). Our architecture aims to evolve to be used as a general solution in healthcare predictions context. We are also interested in studying our solution in future works from the explain ability point of view, with particular interest in explaining the results in terms of input attributes.

Keywords: Electronic Health Record, FI Nnish Diabetes R Isk S Core, Convolutional Neural Network, Multi-Layer Perceptron, Graph Convolutional Network, Graph Transduction Game

Introduction

Patient's Electronic Health Record (EHR) is the set of clinical-health information useful for patient treatment (e.g., clinical and laboratory reports, discharge letters, emergency reports produced in the hospitals and Patient Summary (PS) produced by General Practitioner (GP). Data can have two different states: Validated (e.g., documents digitally signed by a doctor) or not validated (e.g., health data such as pressure, recorded by the patient autonomously; in this scenario, typically we talk about Personal Health Record (PHR). There are more contexts of interest and more points of view. For a patient, we often refer to concepts related to health data and documents associated to: (1) Whole hospitalization. (2) Ward (also considering more hospitalizations or outpatient episodes within the ward itself). (3) GP. (4) Whole hospital (considering more episodes on different wards for more hospitalizations too). (5) All health data and documents of a patient regardless of the structure in which they were created. In these contexts, typically Electronic Patient Record (EPR), Electronic Medical Record (EMR), EHR, PHR acronyms are used with some different meanings. In this study, we mainly use EHR term. EHRs is very important to contain the history of a patient so as to facilitate its care. Important issues related to EHRs, are the standardization of both type of documents and modalities to access them and the interoperability between them. There are different standards in healthcare, e.g., Fast Healthcare Interoperability Resources (FHIR®)[1] and Clinical Document Architecture (CDA®)[2]. Starting from data contained in EHRs, we can use different Machine Learning (ML) techniques to increase and infer knowledge. As highlighted in Shickel *et al.* (2017), many limitations arises for the research in the field of deep EHR learning, in particular: Data heterogeneity, lack of common benchmarks, model interpretability. Our aim is to define an architecture to overcome some of these difficulties too. First, we propose to use structured and standard EHRs also for trying to unify patients' representation, so to reduce data heterogeneity problem. Generally, we could have vector-based representation for health codes but reducing free text usage and better managing e.g., laboratory tests and vital signs. Predictive models improves their accuracies but often without considering e.g., the human interpretability equally important *"…We identify several limitations of current research involving topics such as model interpretability, data heterogeneity and lack of universal benchmarks …"*, Shickel *et al.* (2017). We propose a two components architecture in which generally second component could be at least slightly simpler from explain ability point of view (but improving average accuracy at the same time), although we absolutely have to better explore both components in future works for this particular issue. Moreover, in our first experiment, we use a dataset randomly generated according to a clinical rule, so to overcome privacy problem about real data and to pursue a method for defining common benchmarks based on arbitrary high dataset. First, we highlight some related works in the context of prediction for healthcare, with particular attention for diabetes prediction considered in the experiment explained in this study. Then, after introducing some useful concepts (EHR, FHIR, CDA), we propose Health Prediction Architecture (HPA) based on two principal modules. Starting from structured data in EHR, first module (based on Convolutional Neural Network (CNN) or Multi-Layer Perceptron (MLP)) extracts features to represent the different patients according to a particular health issue. Then, second module (based on Graph Convolutional Network (GCN), Simplified Graph Transduction Game (SGTG) or Nearest Nodes and Classes Graph (NNCG)) makes a prediction on a particular patient. We focus our attention in reasoning and inferencing from structured health data (no free text), so to have an optimal data quality as knowledge dataset. After defining our architecture, we propose a first experiment based on this instance: MLP and SGTG. Our experimental objective is to classify patients at risk of diabetes starting from a dataset generated by using random data defined in respect to a clinical rule (FI Nnish Diabetes R Isk S Core (FINDRISC)). In ML, there are a lot of work on healthcare, particularly about diabetes too (e.g., Dagliati *et al*., 2018). Diabetes (diabetes mellitus) affects the ability to produce the hormone insulin made by the pancreas to help glucose to get body cells from food, so to be used as energy. If blood glucose level is high, there is hyperglycemia. If a person is hyperglycemic and it is not able to regulate its blood glucose level, it is diabetic. Type 1 diabetes depends by the immune system attacking pancreatic beta cells (these produces insulin). Type 2 diabetes depends by insulin resistance. This disease makes the metabolism of carbohydrate abnormal and raise the levels of glucose in the blood (high blood sugar levels over a prolonged period) with consequent symptoms (e.g., more thirst, hunger and urination) and complications when diabetes is not treated. Therefore, diabetes depends mainly on sugar concentration, but it depends also on other factors (e.g., age, Body Mass Index (BMI), hereditary factor). FINDRISC identifies patients at high risk in the context of type 2 diabetes.

Related Work

This paragraph briefly presents some documents in order to frame the topic of interest also in reference to the state of art useful for carrying out our work. We begin with general prediction issues in healthcare. Rajkomar *et al.* (2018) exposed a study aimed at predicting health issues (mortality, readmission within 30 days, prolonged hospitalization and inference of discharge diagnosis). That work uses datasets from two university health institutions. The used method initially considers that health systems maintain patients' EHRs in several formats (structured and unstructured, standardized or not) within their databases. All available data of each patient are encoded in containers based on FHIR specifications. Patient's FHIR resources are placed in temporal order so to represent all patient's EHR events. Deep Learning (DL)

[1] www.hl7.org/fhir

[2] www.hl7.org/implement/standards/product_brief.cfm?product_id=7

model uses this complete history to make any kind of prediction. The work has a double contribution. First, there is a transformation process defined, which is able to take the elements in a patient's EHR as input data to produce outputs in the Health Level 7 (HL7®) FHIR format, without manual harmonization. Moreover, using the data of two hospitals, they demonstrate the effective usefulness of DL in a wide range of predictive models. The approach considered in the work improve model performance and obtains this improvement without the manual selection of the variables by an expert. Futoma *et al.* (2015) studied prediction for readmission within 30 days. Dataset used is New Zealand National Minimum Dataset (New Zealand Ministry of Health) with International Classification of Diseases (ICD) and Diagnosis Related Group (DRG) codes. Methods used, refer to the following models: Logistic Regression (LR), Logistic Regression Variable Selection (LRVS), Penalized Logistic Regression (PLR), Random Forest (RF), Support Vector Machine (SVM) and Deep Neural Network (DNN). Darabi *et al.* (2018) focused on the subject of prediction for patient mortality risk, using public dataset Medical Information Mart for Intensive Care (MIMIC) III. The methods refer to Gradient Boosted Tree (GBT) and DNN. In particular, they use data from EHR related to admission phase (demographic information and ICD codes). Pham *et al.* (2017) considered personalized predictive medicine. A deep DNN (Deep Care) is defined. They start from reading EMRs with previous history of diseases. Then they make inferences about current medical situation (disease progression) and on intervention recommendation. Furthermore, they also consider patient’s future situation prediction (future risk prediction) in terms of readmission and mortality. The model is based on Recurrent Neural Network (RNN) of Long Short Term Memory (LSTM) type and is extended, in particular, to manage variable size discrete inputs, interactions between medical situation and intervention recommendation, irregular timing of events. Manual engineering of features is not required. Choi *et al.* (2016) proposed Doctor AI model. It bases on RNN and it uses episodes recorded in EHR in order to make forecasts for next patients’ episodes. Predictions refer to diagnosis, medical prescriptions and time interval between current and following episode. Data used refers to codes (ICD code, medication code, procedure code). RNN is implemented by Gated Recurrent Unit (GRU), an alternative to LSTM. Henriques *et al.* (2014) proposed two predictive models considering both temporal and cross-attributes dependencies. These models support medical decisions from integrated healthcare databases. Huang *et al.* (2019) had the objective to map clinical notes (free text in EHRs) into ICD-9 codes. MIMIC-III dataset is used and Natural Language Processing (NLP) is based on RNN and CNN. The methodology considers three phases: Data preprocessing, features extraction, model training and testing. DL models are 1D convolution layer (Conv1D), LSTM and GRU. Zhao *et al.* (2019) proposed an alternative approach for predicting cardiovascular diseases in respect to classical techniques based on risk factors and selection of interesting data. They use ML and DL techniques starting from EHR data and genetic data too. Data refer to 109490 patients. Experiments are done first considering only EHR data and after with both EHR and genetic data. Models used are LR, RF, GBT, CNN and RNN with LSTM units. Performances are compared with standard approaches used in routine clinical practice. There are better results compared to standard approaches. Even better results emerge when using genetic data together EHR data. Now, we present some works about prediction in the context of diabetes. Contreras *et al.* (2018) reviewed Artificial Intelligence (AI) techniques related to diabetes, selecting 141 articles. Initially they explore AI techniques considering three issues: Learning from knowledge, exploration and discovery of knowledge, reasoning from knowledge. For the first issue they initially remember some types of techniques: Artificial Neural Network (ANN), SVM, RF, Evolutionary Algorithm (EA), DL, Naïve Bayes (NB), Decision Tree (DT) and regression algorithms. Then, they summarize their results considering the following categories: Blood glucose control strategies; blood glucose prediction; detection of adverse glycemic events; insulin bolus calculators and advisory systems; risk and patient personalization; detection of meals, exercise and faults; lifestyle and daily-life support in diabetes management. In conclusion, they evidence a growth of research in AI for prediction and prevention issues related to diabetes. Swapna *et al.* (2018) studied a methodology to classify diabetic and normal Heart Rate Variability (HRV) signals by using deep learning. HRV signals come from Electro Cardio Gram (ECG) signals. The architecture has three principal modules: CNN, LSTM and SVM. CNN module is composed by 5 CNN layers and each layer have a max pooling. LSTM has 70 memory blocks and there is a dropout 0.1 to remove randomly neurons. SVM is used for final classification from features extracted by CNN and LSTM modules. It uses Radial Basis Function (RBF) kernel. For the experiments, they use Graphics Processing Unit (GPU) with Tensor Flow, Keras and Scikit-learn. The architecture proposed is useful to help the diabetes diagnosis using ECG signals, with an accuracy of 95.7%. Miotto *et al.* (2016) proposed “deep patient”, a framework to model patients by general features obtained automatically from an EHR dataset using DL. They use data from Mount Sinai data warehouse. EHRs are processed in an unsupervised method with a DNN based on Stacked Denoising Autoencoder (SDA) architecture. The framework can be used for different predictions, also related to diabetes diseases. For patients in the dataset, they keep some demographic data, diagnosis (ICD-9 codes), medications, procedures, lab tests and clinical notes (free text). SDA architecture is used to obtain patient representation starting from 704857 patients in the dataset. For prediction about probability that a patient will have a particular disease, they use RF (100 trees for classifier). Sisodia *et al.* (2018) used DT, SVM and NB to predict diabetes at an early stage. Dataset derives from University of California, Irvine (UCI) repository. They measure performance by precision, accuracy, F-measure, recall and Receiver Operating Curve (ROC). For the experiment, they use Waikato Environment for Knowledge Analysis (WEKA) tool. Highest accuracy is for NB algorithm. Kavakiotis and Tsave (2017) presented a review of ML and data mining for diabetes issue about diabetic complications, genetic background and environment, healthcare and management and (most popular) prediction and diagnosis. They show that 85% of ML algorithms used are supervised algorithms and 15% unsupervised. SVM outcomes the most used and with better results. Mercaldo *et al.* (2017) studied a method to classify patients affected by diabetes using Hoeffding Tree (HT) algorithm, also known as Very Fast Decision Tree (VFDT) algorithm. The aim is to classify diabetes patients by the minimum features number to obtain a solution useful in real scenarios. They consider a set of characteristics based on World Health Organization (WHO) criteria. In particular, they consider a vector with the following attributes: Number of times pregnant, plasma glucose concentration at 2 h in an oral glucose tolerance test, diastolic blood pressure, triceps skin fold thickness, 2 h serum insulin, BMI, diabetes pedigree function, age. The study bases on UCI repository, as usual. By classification, they verify if features selected are representative to choice if a patient has diabetes or not. They use the following ML classification algorithms: J48, MLP, HT, JRip, Bayesian Network (BN) and RF. Classification is executed with WEKA tool. They obtain a precision of 0.757 and a recall of 0.762. Erdem *et al.* (2012) introduced Graph Transduction Game (GTG) as a different game-theoretic idea for the problem of graph transduction. In the following, in our work we use and describe GTG for the components of our interest. This problem is defined as a multi-player non-cooperative game and the players are the samples that have to decide their belonging classes. In the experiments done in Erdem *et al.* (2012), they consider also Diabetes dataset from UCI while evaluating the behavior of GTG in respect to other methods.

Electronic Health Record

EHRs contain data that could come from multiple health facilities. We can access these data in accordance with privacy consent rules established by the patient himself too. For data and documents contained in EHR, we can consider two types of problems about standardization point of view: Structuring for clinical health documents and use of standard codes within the documents themselves. Generally, we have to consider that, although the tendency is to structure information as much as possible, it is necessary to assume that in EHR both structured and unstructured data can coexist together. In this study, we focus mainly on predictions starting from consolidated data and therefore we consider structured data and not free text. We suppose to consider EHRs e.g., simply based on structured but not standardized data or, even better, based on data respecting FHIR or CDA Release 2 (CDAR 2) formats.

HL7 FHIR standard was defined with the idea to have a lot of useful features: Easy to develop and with minimum constraints for the necessary tools, semantically robust, friendly for developers, with artifacts that should be clear and intuitive and which should be able to be validated automatically. It uses common formats and tools and web-based technologies for specifications. Furthermore, FHIR refers to modern technologies in the web context (e.g., hypertext Transfer Protocol (http), extensible Markup Language (XML) and JavaScript Object Notation (JSON)). Online specifications consider most formats: Unified Modeling Language™ (UML®) diagrams, XML, JSON and Turtle. The most important concepts of FHIR are essentially the following: Resource, extension, datatype, profile and bundle. A resource is the smallest unit useful for data exchange in an interoperability context. There are different categories of resources including for example medication, diagnostic, workflow management. Different resources are associated within categories. Examples of resources are the following: Patient, Encounter, Medication, Observation, Procedure, Composition and Condition. Each resource consists of several elements, each of which has a datatype. The format of a resource turns out to be understandable for a clinician, even if the specification is for a developer. FHIR considers all aspects of interoperability in the health context (e.g., clinical, administrative, research aspects) and supports interoperability through four paradigms (Representational State Transfer (REST), messaging, document, service). To define the correct use of resources in different contexts, there are the profiles. See, e.g., Pais *et al.* (2017) where they develop a data model of wellness data using FHIR standard. Patients produce their own wellness data using smart phones and portable devices, enriching their PHR and EHR. FHIR support interoperability. They consider blood glucose, blood pressure and BMI data as they relate to diseases such as diabetes and hypertension.

CDA is useful to define the specifications of health clinical documents in XML format. It uses HL 7 Reference Information Model (RIM) and local or controlled vocabularies (using, e.g., ICD, Systematized Nomenclature of Medicine - Clinical Terms (SNOMED® CT), Logical Observation Identifiers Names and Codes LOINC®). With CDAR 2 it is possible to increase interoperability by binding both structure and content of the document. Many scenarios involving CDA documents refer to repositories and registries. Document repository is the archive of health clinical documents for patients. Registry maintains links to the documents stored in different repositories. For example, we can query a registry to retrieve the list of documents associated with a patient. Then we can access the various documents by querying the repositories containing these documents. For example, repositories could belong to health facilities while the index could belong to a regional or national level. Typically, the registry maintains the address of document for its repository, identification metadata (e.g., patient code, healthcare facility and episode) and document type (e.g., discharge letter, laboratory report, radiology report). CDAR 2 document is an XML document with particular rules based on the health clinical context. It is essentially readable. However, at the application level it uses a user-oriented layout. This is possible by using a style sheet (e.g., for transformation to Portable Document Format (PDF)). A CDAR 2 document consists of a header and a body. CDAR 2 validation levels are three. At level one, the validation concerns the requirements as described for CDAR 2 specification. At level two, the validation is for a bound version of the CDAR 2 specifications (e.g., if we have an implementation guide defined for a particular type of document in a specific national context) and in this case, we usually have mandatory sections. At level three, the situation is similar to that of level two but the mandatory choices are higher as they refer to entries in terms of vocabularies or act codes (e.g., an act could be a procedure or a clinical observation). Some examples of implementation guides are the following: International PS (IPS), laboratory medicine report, radiology report, hospital discharge letter. Different types of documents emerge, related to different health topics and regional variants. e.g., already Paterson *et al.* (2002) presented a prototype using CDA in the context of discharge summaries, while Liang *et al.* (2003) studied design and implementation issues for a database to use for efficiently retrieving data from CDA documents.

Although in this study we consider a first experiment with a limited number of input attributes, in Table 1 we present an example of a hypothetical richer structured input data (format, including dimensions) when considering possible patients' EHRs with attributes corresponding to FHIR resources attributes. Similar considerations could be done considering attributes possibly defined from CDA 2 standard documents or considering other types of EHR (standard or not). For a detail of attributes/sub-attributes and format types in this example, refer to 3. Of course, we can have more instances for one single resource for the same patient (e.g.: Observations corresponding to blood tests).

Health Prediction Architecture

Our HPA has two main modules (features extraction and prediction) as described in the Fig. 1 (1 = Known classification patients; 2 = Unknown classification patients; 3 = Features extracted for known classification patients; 4 = Features extracted for unknown classification patients; 5 = Classification predictions for unknown classification patients). For each module, only one sub-module is used for the considered prediction activity.

CNN component (Fig. 2) has the following layers: CONV 2D, batch normalization, activation ("relu"), max pooling 2D, flatten, dense, batch normalization, activation ("relu"), dropout, dense (this layer produces features extraction), batch normalization, activation ("relu"), dropout, dense, activation("softmax").

Table 1: Example of input data extracted from a hypothetical EHR with attributes based on FHIR resources attributes

Module	Resource	Format
Clinical	Allergy intolerance Condition (problem) Procedure Family member history Care plan Goal Care team Clinical impression Adverse event Detected issue	see [4]
Diagnostics	Observation Diagnostic report Service request Media Imaging study Molecular sequence Specimen Body structure	see [5]
Medications	Medication request Medication dispense Medication administration Medication statement Medication Medication knowledge Immunization Immunization evaluation Immunization recommendation	see [6]
Workflow (clinical process)	Referrals (service request) Orders (nutrition order, vision prescription) Device request Supply request	see [7]

[3] www.hl7.org/fhir
[4] https://www.hl7.org/fhir/clinicalsummary-module.html
[5] https://www.hl7.org/fhir/diagnostics-module.html
[6] https://www.hl7.org/fhir/medications-module.html
[7] https://www.hl7.org/fhir/workflow-module.html

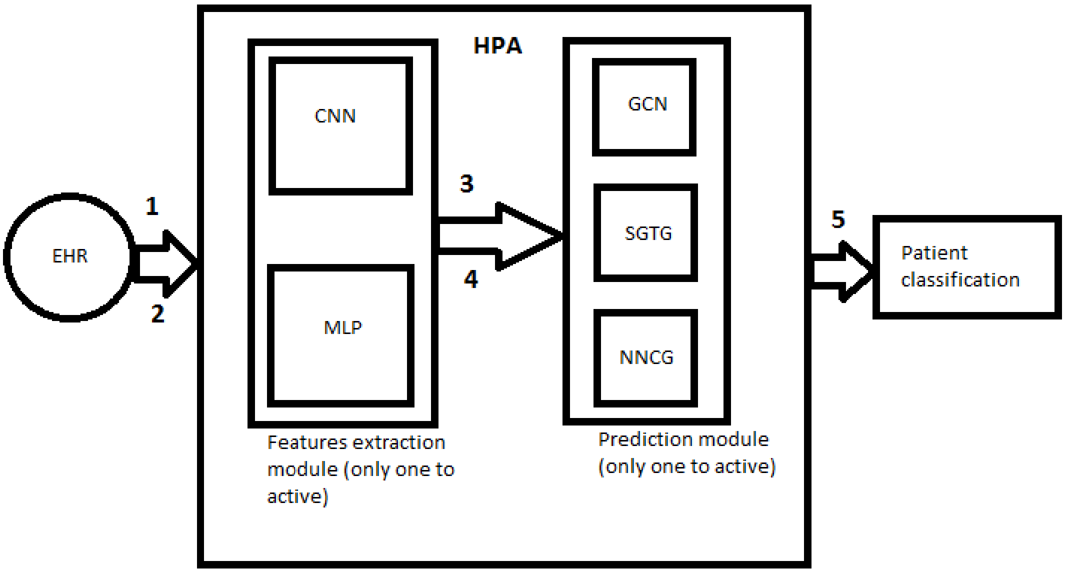
HPA
EHR
1
2
CNN
MLP
Features extraction module (only one to active)
3
4
GCN
SGTG
NNCG
Prediction module (only one to active)
5
Patient classification

Fig. 1: HPA

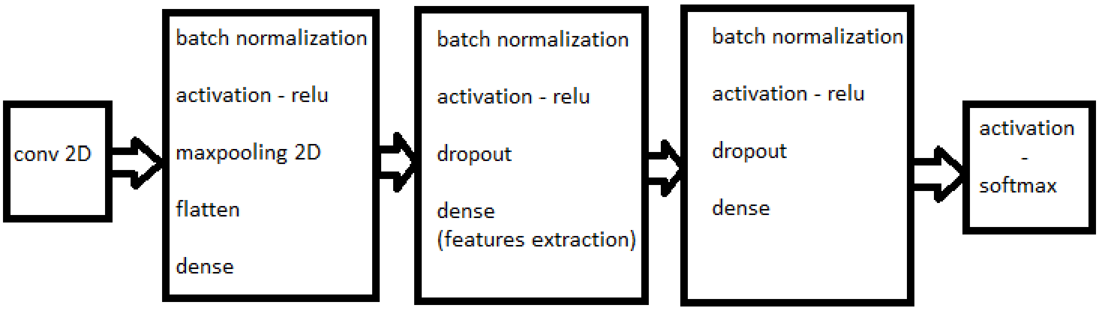

Fig. 2: CNN component

In this scenario, as already considered in other contexts (e.g., Sharma *et al.* (2019), we try to use a CNN with non-image data. A simple possible hypothesis for the dimensions of our virtual "image" refers to the dimensions width*depth*number of channels: (1) width = (n+m+1), where (a) n: Number of fixed clinical test components of interest (e.g., we consider White Blood Cells (WBC) and not Complete Blood Count (CBC)), (b) m: Number of fixed diagnosis considered for inpatient/outpatient visit or discharge, (c) 1: Gender and age range; (2) depth = k, number of the last results considered for the clinical test or the diagnosis (i.e., index 1 is for last value and index k is for k-th last value) for (a) and (b) scenarios and number of gender and age range couples identified, for (c) scenario; (3) number of channels = h, number of health facilities considered for patient data (e.g., h = 2 to consider two health facilities). We consider CNN regardless of the (high or low) number of inputs, so as to emphasize anyway the particular use of this forced representation for structured data.

MLP component has the following layers: Dense, batch normalization, activation ("relu"), dropout, dense, batch normalization, activation ("relu"), dropout, dense (this layer produces features extraction), batch normalization, activation ("relu"), dropout, dense, activation ("softmax"). In this scenario, we consider that each single clinical data is simply an input for our network.

Prediction layer of our architecture needs to evaluate the distances (or similarities) between two data elements (we also use the term node distinguishing between labelled node/element and unlabeled node/element), where each node/element includes its features extracted from first layer. Generally, we will consider different distances/similarities (Euclidean, Cosine, Gaussian kernel with Euclidean, Gaussian kernel with Cosine) and for each distance/similarity we will define two variants (with features correlations weights and without features correlations weights). Feature correlation describes how much important the particular feature is for the prediction over all training nodes. Therefore, in distance definition we can consider also this value.

About GCN component, Kipf and Welling (2016) studied the problem of classifying nodes in a graph where only few nodes have their specific known labels associated. They propose a new approach for this semi-supervised classification problem. We use a derived implementation[8]. In particular, we use GCN Layer and Net classes. This is the mathematical model of GCN:

$$H^{(l+1)} = \sigma\left(D^{-\frac{1}{2}} A D^{-\frac{1}{2}} H^{(l)} W^{(l)} \right) \quad (1)$$

$H^{(l)}$is the l^{th}layer of the network, $W^{(l)}$ is the weight matrix of l^{th} layer, D is the matrix of the graph A, is the adjacency matrix of the graph and σ is the function for the non-linearity. $H^{(0)}$ is the input shape. We can consider more layers and the last layer establishes the dimension of the feature vector for each output node. GCN determines the new representation of node *n*, by changing h_n^1 with linear operation followed by non-linearity:

$$h_n = f\left(W_n h_n^1\right) \quad (2)$$

To build the initial graph, we define this simple algorithm:

```
for i in range (0, number of nodes):
  for j in range (i, number of nodes):
    if similarity[i][j]>precision Similarity:
      add edges (i, j) and (j, i) to the graph
```

We normalize similarity (between 0 and 1), therefore precision Similarity is a fixed value between 0 and 1 (typically greater than 0.5). Each node has the features extracted by CNN or MLP component. After training of our GCN, we evaluate the accuracy of predictions made on testing dataset represented again by features extracted before.

About SGTG component, we must refer to Erdem *et al.* (2012) and we can simply refer to other works useful for our context, e.g., Schiavinato (2014) and Urbani (2018). Initially, refer to the cited works also for the Discrete-Replicator-Dynamics algorithm subsequently used in our work to implement the SGTG component. In ML there are semi-supervised approaches for labelling (give labels to a set of objects). In graph modelling, the objects are nodes and the weights over the edges are similarity measures between nodes. Here, it is of interest to consider Graph Transduction method. The most important assumption is the cluster assumption, which corresponds to the behavior of individuals to bind with similar individuals. In a graph transduction model, there are a set of objects (nodes) with their own features, a subset of these nodes are already labelled and the other subset is not labelled. There is an adjacency matrix W containing the similarities between each couple of nodes. The idea for Graph Transduction is to estimate a consistent labelling for all nodes, starting from the cluster assumption. By GTG, it is possible to define Graph transduction problem as a non-cooperative game. Objects (nodes of graph) are the players of a normal form game. Ideally, labels are the pure strategies. A player, which plays a pure strategy, declare his labelling. The idea is that for a player its payoff is high if the chosen label is the one chosen by similar players. However, to relax this formulation of the problem, it is necessary to consider mixed strategies. Strategies are mixed for unlabeled nodes, while are pure and defined for labelled nodes. To define payoff functions, it is assumed that interactions are only for couples of players. The solution of this game is a Nash equilibrium. After obtaining this equilibrium, then for each player it is chosen as pure strategy (label) the one with the highest probability in the mixed strategy. In our context, the player is a patient represented by its features extracted from CNN or MLP. The mixed strategy profile contains the probabilities over patients' classification and corresponds to the players' choices to maximize their payoff. We achieve a high payoff when the player choices a mixed strategy such that the corresponding similar patient choices are similar. We define a simplified version of GTG, named SGTG, essentially adjusting the similarity matrix as follows (only for pairs of testing, unlabeled, nodes):

```
for i in range (0, number of testing nodes):
  for j in range (i, number of testing nodes):
    if W[i][j]>precision Similarity:
         W[i][j]=1.0
         W[j][i]=1.0
 else:
         W[i][j]=0.0
         W[j][i]=0.0
```

We normalize similarity (between 0 and 1), therefore precision Similarity is a fixed value between 0 and 1 (typically greater than 0.5). After initialization of mixed strategies X (unlabeled nodes: 1/m for each possible label, where m is the number of possible labels; labelled node: 1 for the right label, 0 otherwise), the implementation of prediction algorithm (SGTG derived from GTG/Discrete-Replicator-Dynamics) is the following:

```
for t in range (0, int (number of testing nodes**(1/2))):
  for i in range (0, number of testing nodes):
```

[8] https://docs.dgl.ai/tutorials/models/1_gnn/1_gcn.html

```
   S = 0
  for j in range (0, number of all nodes):
   for h in range (0, m):
    S+ = X[i][h]*W[i][j]*X[j][h]
  for h in range (0, m):
   S1 = 0
   for j in range (0, number of all nodes):
    S1+ = W[i][j]*X[j][h]
  X[i][h] = X[i][h]*S1/S
```

For replicator dynamics, its fixed points are Nash equilibrium. The simplified algorithm starts from the operator used to have a discretization for time ($t \in N$):

$$x_i^{(t+1)}\left(h\right) = x_i^{(t)}\left(h\right)\frac{u_i\left(e_i^h, x_{-i}^{(t)}\right)}{u_i\left(x^{(t)}\right)} \tag{3}$$

In this formula, u is the payoff function of all players, x is the mixed strategy profile. For more details, see Erdem *et al.* (2012) too.

After execution of this code, we evaluate the accuracy of predictions obtained for initially unlabeled nodes, simply considering argmax.

We define NNCG algorithm as a strong simplification of GTG algorithm. In particular, for each unlabeled node we consider a similarity array containing the similarities between the particular unlabeled node and all labelled nodes. Considering to have the same number of known nodes for each possible labels, we simply determine the label for an unknown node as follows:

```
S1 = [0.0 for i in range (0, m)]
S1Temp = [0.0 for i in range (0, m)]
for i in range (0, training Nodes Numbers[n]):
  for h in range (0, m):
    if h==X [i]:
      S1Temp[h]+=W[i]
break
```

When classes are ordered (e.g.: Diabetes risk levels), we adopt an adding code as follows to consider similarity between labels too:

```
for h in range (0, m):
 for z in range (0, m):
 S1[h]+ = S1Temp[h]*(1-abs(z-h)/m)
```

After execution of this code, we evaluate the accuracy of predictions obtained for initially unlabeled nodes, simply considering argmax.

In Fig. 3, we present a graph representation example valid for both GCN and SGTG from a general conceptual point of view. Notice that the arc is present if the similarity is sufficiently high between two nodes. Node n is highly similar to an unlabeled node and to two labelled nodes not similar between them (and belonging to different classes). In Fig. 4, we present an example for NNCG. In this case, we have an edge for all couples of labelled and unlabeled nodes, but the weight of each edge depends on the similarity between the two nodes. Here, we are not interested in similarity between two unlabeled nodes.

First Experiment

We are doing different experiments with possible instantiations of our architecture, initially considering as dataset a simple version of XML documents or simpler data too. More generally, as methodology for our research we have operated essentially with these tasks:

1) Define features extraction module (sub-modules) as variant from literature
2) Define prediction module (sub-modules) as variant from literature
3) Loop
   - Instance the architecture
   - Test the architecture
   - Redefine module instantiation parameters according to testing results until stable and satisfactory results

In this study, we present a first simple experiment based on the instantiation of MLP for features extractions module and SGTG for prediction module. In this experiment we generate randomly an EHR limited to the useful clinical data and without a FHIR or CDAR 2 (or simply XML) representation. Useful clinical data refers to diabetes patients' classification problem. In particular, we have to identify patients at different risk level in the context of type 2 diabetes. We consider accuracy for our evaluation, but using a different definition due to the ordered labels for this particular problem. For this experiment, we use Collaborator 9 as environment to execute our code, selecting Python™ 3 and using GPU as runtime environment.

About FINDRISC, it usually comprises the following eight items: Age (years) (A), Body Mass Index (BMI, weight (kg)/height squared (m2)) (B), Waist circumference (W) (differentiating for Gender (G)), Use of blood pressure medication (U), History of high blood glucose (H), Physical activity expressed in hours/week (P), Daily consumption of vegetables, fruits or berries (D), Family history of diabetes (F). We use a derivate algorithm[10] to produce our training and testing datasets. All input data are normalized to [0,1] and are balanced in respect to the diabetes risk. The maximum achievable score is 26 and there are five risk levels in respect to score: Very low (0-3), low (4-8), moderate (9-12), high (13-20) and very high (21-26).

[9] https://colab.research.google.com/

[10] https://www.mdcalc.com/findrisc-finnish-diabetes-risk-score

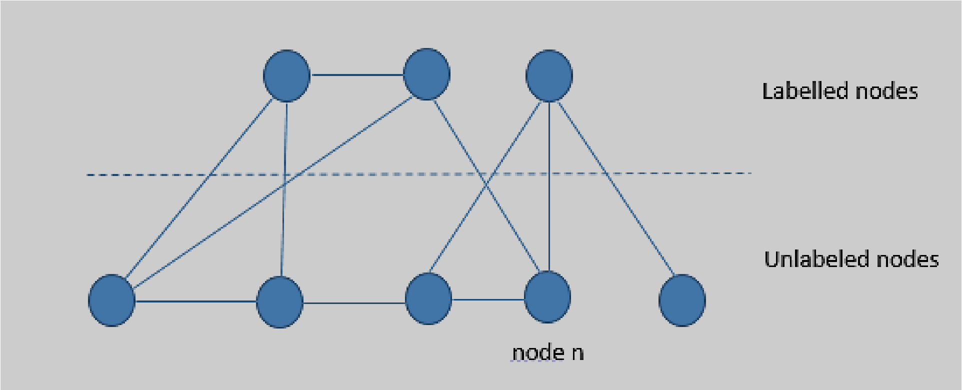

Fig. 3: Graph representation for GCN and SGTG

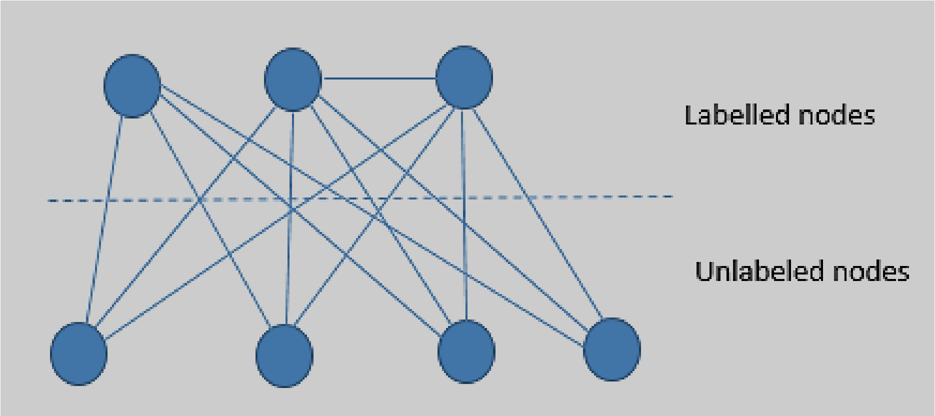

Fig. 4: Graph representation for NNCG

Considering the possibility to order the labels values, we establish the definition of accuracy, named accuracy', to evaluate our model:

$$accuracy' = \frac{\sum_{i=1}^{np}\left(1-\frac{\left|\left(\arg\max_{j\in\{1,\dots,m\}} PLi(j)\right)-RLi\right|}{m-1}\right)}{np} \qquad (4)$$

Where:

- np: Total number of predictions (unlabeled nodes)
- RL: Right label for *i* node
- PLi (j): Risk j level prediction for *i* node
- m: Number of possible labels (in our study, m = 5)

For our experiment, we consider the following datasets, all equally distributed in term of possible labels/risk levels (we generate randomly datasets using FINDRISC clinical rule):

1) Testing dataset: It is the same during all the experiment and it contains 400 nodes (unlabeled patients)
2) Training dataset: It is generated for each value of number of nodes (labeled patients) and for each value of extracted features number

In particular: (1) We consider these possible labelled nodes numbers: 100, 125, 150, 175, 200; (2) we consider these possible extracted features numbers: 25, 30, 35. Our interest is related to a scenario for which testing dataset is greater than training dataset. Moreover, initially we consider relatively low cardinality for the considered population. In Table 2, we present data distribution for all datasets. Mean and standard deviation have been truncated from the third decimal digit. FINDIRSC parameters have been normalized in the range [0,1]. FINDRISC parameters are:

- 0 = gender
- 1 = age
- 2 = body Mass Index
- 3 = waist Circumference
- 4 = blood Pressure Medication
- 5 = blood Glucose

- 6 = physical activity
- 7 = daily vegetables
- 8 = family history

For second layer, as similarity measure, in this first experiment we choose simply Euclidean distance without features correlations weights. We also normalize the values (W is the adjacency, distance/similarity, matrix for nodes):

$$w_{i,j} = 1 - \frac{\sqrt{\sum f\left(F_{i,f} - F_{j,f}\right)^2}}{\max_{i,j}\sqrt{\sum_f\left(F_{i,j} - F_{j,f}\right)^2}} \qquad (5)$$

In this experiment, we instantiate MLP as follows: (1) Number of units for first dense layer: 1280; (2) Number of units for second dense layer: 160; (3) Number of units for third dense layer: Equals to the number of extracted features; (4) number of units for fourth dense layer: 5; (5) dropout parameter: 0.25. Input shape for first dense layer has dimension 9, corresponding to the number of clinical attributes related to FINDRISC clinical rule. We use Adam optimizer with learning rate equals to 0.01, compiling with sparse categorical cross entropy for loss and accuracy as metrics. We fit our model using 1000 epochs, 0.2 as validation split, 100 as batch size. For SGTG, we instantiate precision Similarity parameter with the value of 0.75.

In Table 3 and Fig. 4, 5, 6, we present the results.

We can see that generally the behavior of HPA is slightly better than MLP alone. By a simple analysis of results, we can see that we have the best results of accuracy (both for HPA and only MLP) with a training dataset of 200 elements (the maximum number of elements) and with 25 extracted features (the minimum number of extracted features). Moreover, only 25 and 30 extracted features scenarios, have and increment of accuracy from MLP to HPA in all cases.

Medium execution times for training and prediction (for MLP we consider both training and prediction together, while for SGTG we have only prediction) are:

1) MLP = 41.67 sec (min = 17, max = 60)
2) SGTG = 34.67 sec (min = 28, max = 41)

SGTG has a relatively good response time for prediction in this experiment, but we have to consider that the number of nodes is not high and, simplifying, the cost is about $O(n^{5/2})$, usually high.

As we can see, already in this first experiment of the architecture, we obtained a better accuracy then using only MLP and we established a method of testing based on randomized controlled datasets, so to better evaluate the model.

Table 2: Data distribution for all datasets

Number of elements	Findrisc parameter	Mean	Standard deviation
400	0	0.49	0.49
400	1	0.50	0.30
400	2	0.51	0.32
400	3	0.44	0.29
400	4	0.47	0.49
400	5	0.53	0.49
400	6	0.50	0.31
400	7	0.52	0.49
400	8	0.60	0.40
100	0	0.53	0.49
100	1	0.51	0.30
100	2	0.49	0.28
100	3	0.50	0.29
100	4	0.54	0.49
100	5	0.53	0.49
100	6	0.49	0.33
100	7	0.52	0.49
100	8	0.56	0.42
125	0	0.52	0.49
125	1	0.49	0.28
125	2	0.48	0.33
125	3	0.46	0.30
125	4	0.50	0.49
125	5	0.49	0.49
125	6	0.50	0.33
125	7	0.55	0.49
125	8	0.59	0.39
150	0	0.53	0.49
150	1	0.50	0.29
150	2	0.50	0.32
150	3	0.48	0.30
150	4	0.49	0.49
150	5	0.53	0.49
150	6	0.51	0.31
150	7	0.58	0.49
150	8	0.63	0.39
175	0	0.44	0.49
175	1	0.49	0.30
175	2	0.50	0.31
175	3	0.44	0.31
175	4	0.43	0.49
175	5	0.57	0.49
175	6	0.52	0.33
175	7	0.56	0.49
175	8	0.58	0.41
200	0	0.54	0.49
200	1	0.48	0.29
200	2	0.49	0.29
200	3	0.46	0.29
200	4	0.52	0.49
200	5	0.56	0.49
200	6	0.53	0.31
200	7	0.48	0.49
200	8	0.56	0.39

Table 3: Accuracy’ results for testing (unlabeled) datasets

Number of labelled elements	Number of extracted features	MLP accuracy’	HPA accuracy’
100	25	0.865625	0.878750
100	30	0.863125	0.870625
100	35	0.861875	0.876875
125	25	0.883125	0.887500
125	30	0.885625	0.889375
125	35	0.886875	0.886875
150	25	0.896875	0.901250
150	30	0.896875	0.903750
150	35	0.896875	0.898125
175	25	0.883750	0.896875
175	30	0.891875	0.900625
175	35	0.894375	0.902500
200	25	0.913125	0.917500
200	30	0.906250	0.912500
200	35	0.908750	0.908125

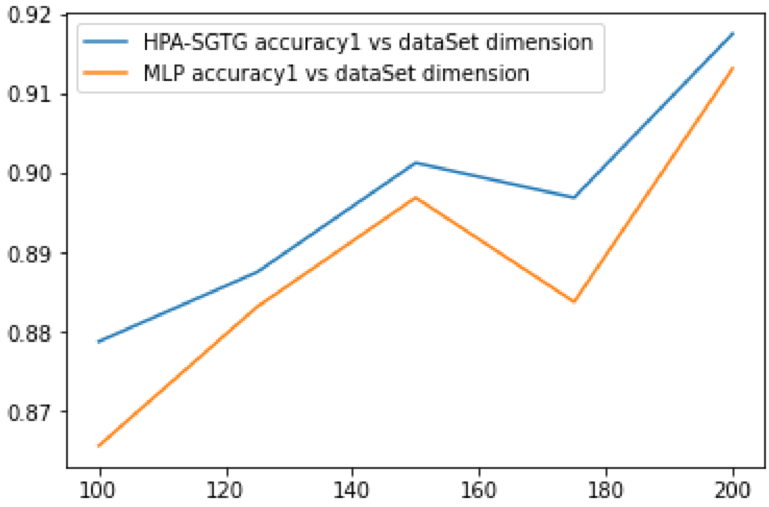

Fig. 4: Accuracy' vs training data Set cardinality (number of features extracted = 25)

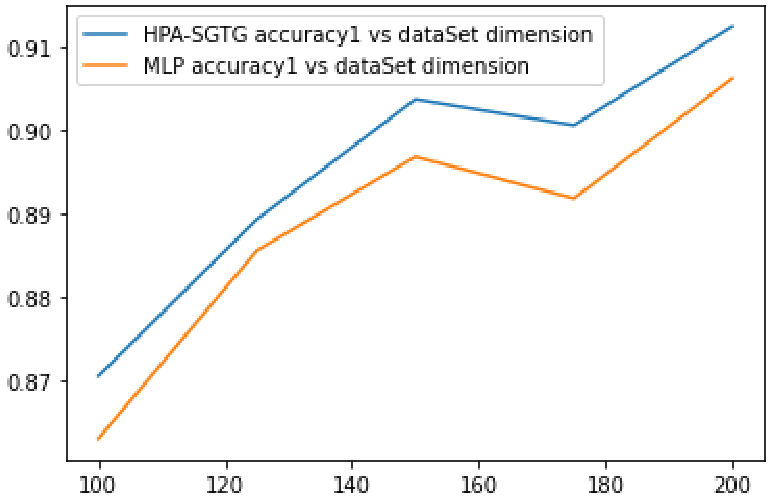

Fig. 5: Accuracy' vs training data Set cardinality (number of features extracted = 30)

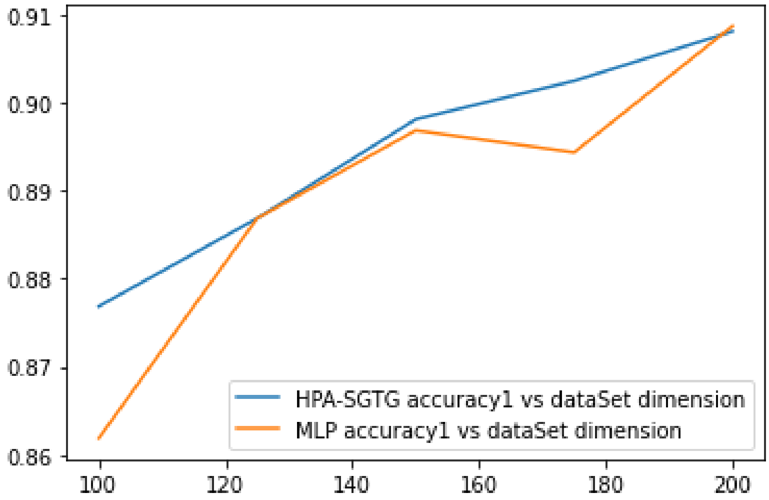

Fig. 6: Accuracy' vs training data set cardinality (number of features extracted = 35)

Conclusion

In this study, we proposed a general architecture named HPA which is composed of two principal modules. First module, based alternatively on CNN or MLP, produces features extraction to represent nodes. Second module, based alternatively on GCN, SGTG or NNCG, produces predictions for nodes. The context of interest belongs to healthcare for patient classification using structured health data. We presented EHR and some related standards (FHIR and CDAR 2). We propose our architecture to manage some issues (dataset heterogeneity, accuracy but considering explain ability too, dataset for benchmarking). We are interested in structured data and for this reason we are interested in FHIR and CDAR 2 too, so to evaluate more standardized dataset. In this study, we also implemented a particular instance of our architecture as example test. In particular, we considered the problem of patient classification for diabetes: (1) According to a clinical rule (FINDRISC), we produced a balanced set of labelled and unlabeled diabetes patients as initial dataset for our test; (2) we defined and trained a MLP to extract features for our nodes; (3) we implemented SGTG to do predictions about diabetes risk using Euclidean distance for similarity matrix and using Discrete-Replicator-Dynamics algorithm with some variants. We obtained some results in term of accuracy' (variant of accuracy definition) comparing MLP predictions with HPA predictions. In future work, we are interested in presenting new instances of our HPA, trying to optimize accuracy and showing the results of other current and future experiments. In particular, we also want to verify the best solutions according to the number of dataset elements (training and testing), comparing GCN, SGTG and NNCG results and considering CNN and MLP. We are also interested in evaluating the behavior according to the use of different similarity/distance definitions. We want to evaluate the behavior of HPA with known datasets. We have to consider training dataset with few patients in respect to testing dataset and evaluate the behavior when considering high amount of data. Moreover, we are also interested in evaluating HPA as explainable in term of input variables (patient parameters), using Deep Taylor decomposition too (e.g., Montavon *et al.* (2017). The initial idea is to define a first explainable layer specific for prediction sub-modules and a second explanation layer specific for features extraction sub-modules. Especially for each prediction sub-modules, it could be necessary to define a particular solution based on the theory related to the considered sub-module. Subsequently, Deep Taylor decomposition could be mainly used e.g., for features extraction sub-modules.

Funding Information

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

Author’s Contributions

Alessandra Pieroni: Designed the research plan and organized the study; coordinated the data-analysis and contributed to the writing of the manuscript.

Alessandro Cabroni: Designed the research plan and organized the study; coordinated the data-analysis and contributed to the writing of the manuscript; participated in all the experiments.

Francesca Fallucchi: Designed the research plan and organized the study; coordinated the data-analysis and contributed to the writing of the manuscript.

Noemi Scarpato: Designed the research plan and organized the study.

Ethics

This study neither has been published nor is under review elsewhere.

References

Choi, E., Bahadori, M. T., Schuetz, A., Stewart, W. F., & Sun, J. (2016, December). Doctor ai: Predicting clinical events via recurrent neural networks. In Machine learning for healthcare conference (pp. 301-318). PMLR. http://proceedings.mlr.press/v56/Choi16

Contreras, I., & Vehi, J. (2018). Artificial intelligence for diabetes management and decision support: literature review. Journal of medical Internet research, 20(5), e10775. https://www.jmir.org/2018/5/e10775/

Dagliati, A., Marini, S., Sacchi, L., Cogni, G., Teliti, M., Tibollo, V., ... & Bellazzi, R. (2018). Machine learning methods to predict diabetes complications. Journal of diabetes science and technology, 12(2), 295-302. https://journals.sagepub.com/doi/full/10.1177/1932296817706375

Darabi, H. R., Tsinis, D., Zecchini, K., Whitcomb, W. F., & Liss, A. (2018). Forecasting mortality risk for patients admitted to intensive care units using machine learning. Procedia Computer Science, 140, 306-313. doi.org/10.1016/j.procs.2018.10.313

Sisodia, D., & Sisodia, D. S. (2018). Prediction of diabetes using classification algorithms. Procedia computer science, 132, 1578-1585. doi.org/10.1016/j.procs.2018.05.122

Erdem, A., & Pelillo, M. (2012). Graph transduction as a noncooperative game. Neural Computation, 24(3), 700-723. https://ieeexplore.ieee.org/abstract/document/6797349/

Futoma, J., Morris, J., & Lucas, J. (2015). A comparison of models for predicting early hospital readmissions. Journal of biomedical informatics, 56, 229-238. doi.org/10.1016/j.jbi.2015.05.016

Henriques, R., & Antunes, C. (2014, January). Learning predictive models from integrated healthcare data: Extending pattern-based and generative models to capture temporal and cross-attribute dependencies. In 2014 47th Hawaii International Conference on System Sciences (pp. 2562-2569). IEEE. https://ieeexplore.ieee.org/abstract/document/6758922/

Huang, J., Osorio, C., & Sy, L. W. (2019). An empirical evaluation of deep learning for ICD-9 code assignment using MIMIC-III clinical notes. Computer methods and programs in biomedicine, 177, 141-153. doi.org/10.1016/j.cmpb.2019.05.024

Kavakiotis, I., Tsave, O., Salifoglou, A., Maglaveras, N., Vlahavas, I., & Chouvarda, I. (2017). Machine Learning and Data Mining Methods in Diabetes Research. Computational and Structural Biotechnology Journal, 15, 2017. doi.org/10.1016/j.csbj.2016.12.005

Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907. https://arxiv.org/abs/1609.02907

Liang, Z., Bodorik, P., & Shepherd, M. (2003, January). Storage model for cda documents. In 36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the (pp. 10-pp). IEEE. https://ieeexplore.ieee.org/abstract/document/1174352/

Mercaldo, F., Nardone, V., & Santone, A. (2017). Diabetes mellitus affected patients classification and diagnosis through machine learning techniques. Procedia computer science, 112, 2519-2528. doi.org/10.1016/j.procs.2017.08.193

Miotto, R., Li, L., Kidd, B. A., & Dudley, J. T. (2016). Deep patient: an unsupervised representation to predict the future of patients from the electronic health records. Scientific reports, 6(1), 1-10. https://www.nature.com/articles/srep26094

Montavon, G., Lapuschkin, S., Binder, A., Samek, W., & Müller, K. R. (2017). Explaining nonlinear classification decisions with deep Taylor decomposition. Pattern Recognition, 65, 211-222. doi.org/10.1016/j.patcog.2016.11.008

Pais, S., Parry, D., & Huang, Y. (2017, January). Suitability of fast healthcare interoperability resources (FHIR) for wellness data. In Proceedings of the 50th Hawaii International Conference on System Sciences. https://scholarspace.manoa.hawaii.edu/handle/10125/41581

Paterson, G. I., Shepherd, M., Wang, X., Watters, C., & Zitner, D. (2002, January). Using the XML-based clinical document architecture for exchange of structured discharge summaries. In Proceedings of the 35th Annual Hawaii International Conference on System Sciences (pp. 1200-1209). IEEE. https://ieeexplore.ieee.org/abstract/document/994069/

Pham, T., Tran, T., Phung, D., & Venkatesh, S. (2017). Predicting healthcare trajectories from medical records: A deep learning approach. Journal of biomedical informatics, 69, 218-229. doi.org/10.1016/j.jbi.2017.04.001

Rajkomar, A., Oren, E., Chen, K., Dai, A. M., Hajaj, N., Hardt, M., ... & Dean, J. (2018). Scalable and accurate deep learning with electronic health records. NPJ Digital Medicine, 1(1), 1-10. ttps://www.nature.com/articles/s41746-018-0029-1%22

Schiavinato, M. (2014). A Game-Theoretic Approach to Graph Transduction: An Experimental Study (Bachelor's thesis, Università Ca'Foscari Venezia). http://dspace.unive.it/handle/10579/4381

Sharma, A., Vans, E., Shigemizu, D., Boroevich, K. A., & Tsunoda, T. (2019). DeepInsight: A methodology to transform a non-image data to an image for convolution neural network architecture. Scientific Reports, 9(1), 1-7. https://www.nature.com/articles/s41598-019-47765-6

Shickel, B., Tighe, P. J., Bihorac, A., & Rashidi, P. (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), 1589-1604. https://ieeexplore.ieee.org/abstract/document/8086133

Swapna, G., Vinayakumar, R., & Soman, K. P. (2018). Diabetes detection using deep learning algorithms. ICT Express, 4(4), 243-246. doi.org/10.1016/j.icte.2018.10.005

Urbani, P. (2018). Combining Deep Learning and Game Theory for Music Genre Classification (Bachelor's thesis, Università Ca'Foscari Venezia). http://157.138.7.91/handle/10579/12034

Zhao, J., Feng, Q., Wu, P., Lupu, R. A., Wilke, R. A., Wells, Q. S., & Wei, W. Q. (2019). Learning from longitudinal data in electronic health record and genetic data to improve cardiovascular event prediction. Scientific Reports, 9(1), 1-10. https://www.nature.com/articles/s41598-018-36745-x