Medical concept embedding of real-valued electronic health records with application to inflammatory bowel disease

Hanan Mann[1] **Aharon Bar Hillel**[1] **Raffi Lev-Tzion**[2] **Shira Greenfeld**[3] **Revital Kariv**[3]
Natan Lederman[4] **Eran Matz**[5] **Iris Dotan**[6] **Dan Turner**[2] **Boaz Lerner**[1*]

[1] Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Israel
[2] The Juliet Keidan Institute of Pediatric Gastroenterology and Nutrition, Shaare Zedek Medical Center, The Hebrew University of Jerusalem, Israel
[3] Maccabi Healthcare Services, Tel Aviv, Israel
[4] Meuhedet Health Services, Tel Aviv, Israel
[5] Leumit Health Services, Tel Aviv, Israel
[6] Division of Gastroenterology, Rabin Medical Center, Petah Tikva, and the Sackler Faculty of Medicine, Tel Aviv University, Israel

[*]Correspondence author: boaz@bgu.ac.il

Abstract

Deep learning approaches are gradually being applied to electronic health record (EHR) data, but they fail to incorporate medical diagnosis codes and real-valued laboratory tests into a single input sequence for temporal modeling. Therefore, the modeling misses the existing medical interrelations among codes and lab test results that should be exploited to promote early disease detection. To find connections between past diagnoses, represented by medical codes, and real-valued laboratory tests, in order to exploit the full potential of the EHR in medical diagnosis, we present a novel method to embed the two sources of data into a recurrent neural network. Experimenting with a database of Crohn's disease (CD), a type of inflammatory bowel disease, patients and their controls (~1:2.2), we show that the introduction of lab test results improves the network's predictive performance more than the introduction of past diagnoses but also, surprisingly, more than when both are combined. In addition, using bootstrapping, we generalize the analysis of the imbalanced database to a medical condition that simulates real-life prevalence of a high-risk CD group of first-degree relatives with results that make our embedding method ready to screen this group in the population.

Keywords *embedding, electronic health record (EHR), gated recurrent unit (GRU), Crohn's disease, lab test result, medical concept.*

1 Introduction

Electronic health records (EHRs) contain information about a patient's medical status. While the primary goal of EHRs is to monitor a patient, they can also be used to represent patients' states in data-driven medical prediction systems [1,2]. However, although an EHR contains a lot of information about a patient, it poses challenges for data-driven systems such as high-dimensionality of data, sparseness, data collected irregularly from several sources not synchronized, missing values, and imbalance. An EHR typically contains demographical data (e.g., age and sex), physical measurements (e.g., height, weight, and BMI), lifestyle information regarding smoking and drinking habits, and medical entities manifested temporarily, such as lab tests, diagnoses (ICD9 codes [3]), and evidence of medication purchase [2,4]. While some entities, such as diagnoses and medication purchases, are represented as categorical variables (by ICD9 and ATC2 codes, respectively), others, such as lab tests, have real values [1,2,4]. This undermines the combination of both entity types in a single longitudinal clinical representation for temporal modeling, extraction of connections among entities, and disease prediction.

In recent years, deep learning (DL) has been applied to EHR data [1,5–11] to extract and exploit connections among medical entities to understand disease development better and to enable early detection. However, none of these studies have incorporated medical entities based on real values with others based on categorical information into a single temporal representation of all the longitudinal medical information available in the EHR [1]. Avoiding real values such as lab test results is likely to deteriorate a model's quality, as this information is relevant for understanding

and predicting patient status. When a physician examines a lab result, they are interested in knowing if its value deviated from the appropriate reference interval describing healthy individuals, and by how much [12,13]. Therefore, providing the model with only the information that a lab test happened, without its associated value, may not be meaningful.

This work aims to explore innovative ways to embed diagnostic data for use by machine learning (ML) to promote digital healthcare. We propose a novel method to exploit EHR sequences containing both categorical and real-valued medical entities for DL, mainly recurrent neural network (RNN), modeling. The method builds on embedding both diagnosis and lab test data in a common latent representation. We consider elements of this latent representation as medical concepts and demonstrate this medical concept embedding method in predicting Crohn's disease (CD), a type of inflammatory bowel disease (IBD), in people who will be diagnosed later in life. Section 2 of the paper describes previous medical concept embedding of EHR entities, mostly for disease prediction by RNNs. Section 3 proposes our novel embedding method for the real-valued lab tests. Section 4 presents our data and methodology, while Section 5 provides results of the study in different balanced and imbalanced settings of data corresponding to practical CD screening scenarios. Section 6 concludes the study and discusses avenues for further research.

2 Background

2.1 Medical concept embedding

ML-driven feature representation of medical entities extracted from EHR data is often referred to as medical concept embedding [6–8,14–16] . For large data sets, disease-prediction tasks based on concept embedding outperformed those that used other feature-learning strategies [5,14,16,17]. It is common to learn concept embedding (not only in the medical case [18–25]) in an unsupervised setting without labels, and to feed the learned representation (embedding) into a supervised setting as an input to the predictor [5,10,11,17,18,24]. However, it is also possible to learn the representation during a supervised task [18]. Several works have used medical concept embedding to learn the representation of ICD9 codes for patient subtyping [26], to learn patient features from ECG data, albeit missing observations [27], to incorporate code co-occurrence and visit-sequence information in a two-layer neural network called Med2Ved [14], and to create pre-trained lab test embedding from lab test codes [9]. However, neither of these (or other [1]) works incorporated codes and real-valued tests.

2.2 Disease prediction using deep learning

Most DL models applied to EHR data are RNNs. An RNN is a neural network that can handle variable-length sequence inputs [28], making it suitable to handle EHRs. There are two common extensions to the vanilla RNN unit: the long short-term memory (LSTM) unit [29] and the gated

recurrent unit (GRU) [25]. The choice between LSTM and GRU should be based on the data set and corresponding task [28]. Both have been applied equally in the EHR domain, but GRU is more popular than LSTM for structured medical data like we have [1]. For this reason and because the GRU often converges faster, we chose to work with it.

Several studies have applied RNNs to structured data extracted from EHRs (mainly clinical codes such as diagnoses, medications, and procedures), particularly for disease prediction, to predict diagnoses and medications for a subsequent clinic visit using RNN [30], to predict sepsis using LSTM [31], and to predict the onset of heart failure using GRU [9,32], among other examples [1,5]. Also the non-RNN BERT model [24] was recently adopted for health care tasks using the EHR [6,10,11] as were temporal convolution networks [8]. However, real-valued measurements associated with diagnoses were neglected in all these works which model EHR as a sequence of discrete-time events [1].

3 Embedding real-valued lab tests

Our study aims to simultaneously represent diagnoses and lab test results in the EHR in a suitable way for DL. EHR of subject s was represented as a set of medical entities, ordered chronologically, $V_s = \{\boldsymbol{v}_s^1, \boldsymbol{v}_s^2, \boldsymbol{v}_s^3, \ldots, \boldsymbol{v}_s^{n_s}\}$ [10,11], where $\boldsymbol{v}_s^j$ is a medical entity, e.g., a diagnosis or a lab test result of subject s at time stamp j, and n_s is the total number of medical entities in the EHR of this subject. For diagnoses (represented by ICD9 codes), we used a one-hot-vector indicating the specific medical code ($\boldsymbol{x}_s^j$). For lab test results, we used a tuple with a one-hot-vector to represent the test type and ($\boldsymbol{v}_s^j = (\boldsymbol{x}_s^j, \alpha_s^j)$) to represent the real-value measured, α_s^j.

To handle lab test data, we may have Z-score scaled [33] any lab test to compare its value to the norm (reference) for this test. However, a challenge is that all lab test results will receive a zero value for their mean values, so the model could not distinguish mean values of different lab tests. To address this challenge, we propose using two embedding matrices, C and V. Matrix C represents the mean of the lab tests, and matrix V represents their Z-scores. Combining these matrices allows us to represent each lab test uniquely. Such a model mimics the way a physician analyzes lab test results by first recognizing the existence of the lab test and then evaluating how its value deviates from the norm. As the appendix shows, such representation also supports the unification of the ICD9 codes with the lab test results under the same setting, providing a latent concept shared between lab tests and diagnoses to holistically represent a person's medical condition.

Figure 1 illustrates our proposed method, where for each person (we drop the subject subscript s) each medical entity in the EHR is broken down into its one-hot-vector (x^j) and measured value (α^j). This broken medical entity is embedded by matrices C and V to obtain c^j and v^j. After Z-

scaling α^j to Z^j, Z^j and v^j are multiplied and the result is added to c^j. This process is repeated for each medical entity, and the final result is fed into a GRU layer for prediction.

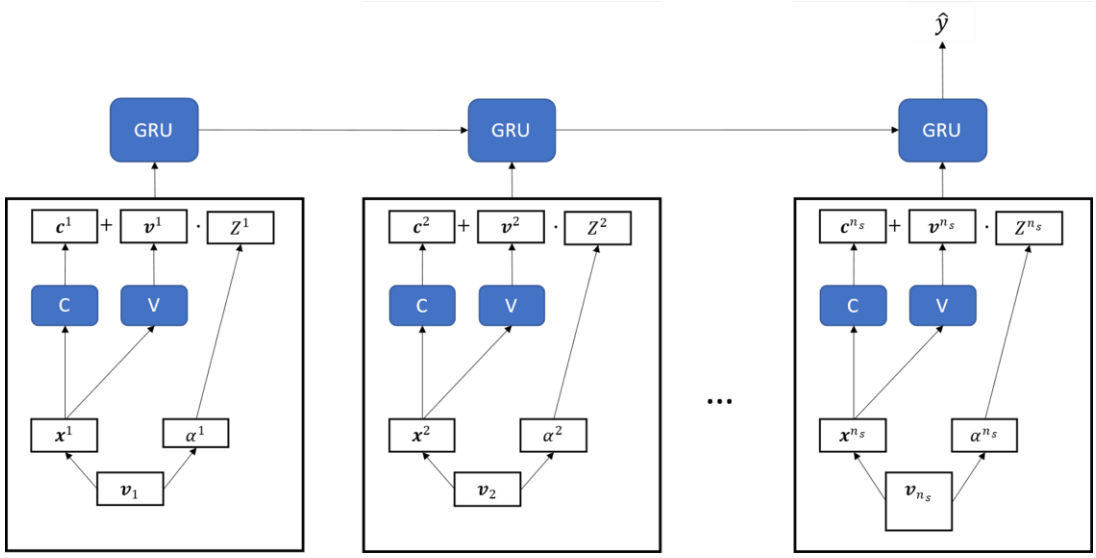

Figure 1. A suggested embedding method for DL manifested by a GRU.

4 Materials and methods

We describe here the database used for the experiments (Section 4.1), introduce the evaluation strategy (Section 4.2), and discuss the implementation of that strategy (Section 4.3).

4.1 Data understanding and preparation

The data were collected, labeled, and de-identified by Shaare Zedek Medical Center (SZMC). The data contain information from three of the four health maintenance organizations (HMOs) in Israel, covering approximately 48% of the population [34]. We focused on CD, which has more often a delay in diagnosis compared with the other IBD, ulcerative colitis, posing a larger challenge to meet by the clinical community [35]. The data contained 14,992 CD cases. Non-CD control subjects were matched by age, sex, jurisdiction, and HMO. After excluding cases without matched controls and vice versa, we were left with 12,917 CD cases and 37,900 matched controls, where 12,204 CD cases had three matched controls, 575 CD cases had two matched controls, and 138 CD cases had one matched control subject. Table 1 shows the characteristics of the study population where for the controls the age at diagnosis and time since diagnosis were calculated based on the matched CD cases.

	Crohn's disease, N (%)	**Control, N (%)**
Total	12,917 (100)	37,900 (100)
Female	6,175 (47.81)	18,168 (47.94)
Male	6,742 (52.19)	19,732 (52.06)
Age at diagnosis		
Mean (STD)	33.54 (16.72)	33.51 (16.57)
< 18	2,255 (17.46)	6,482 (17.1)
$18 - 39$	6,569 (50.86)	19,476 (51.39)
$40 - 59$	2,946 (22.81)	8,671 (22.88)
≥ 60	1,147 (8.88)	3,271 (8.63)
Year of diagnosis		
< 2000	1,460 (11.30)	4,283 (11.3)
$2000 - 2005$	2,927 (22.66)	8,551 (22.56)
$2006 - 2010$	2,494 (19.31)	7,358 (19.41)
$2011 - 2015$	3,133 (24.25)	9,202 (24.28)
$2016 - 2020$	2,903 (22.47)	8,506 (22.44)
Disease activity		
Mild	2,929 (22.68)	8,625 (22.76)
Moderate	4,695 (36.35)	13,809 (36.44)
Severe	1,384 (10.71)	4,043 (10.67)
Unknown	3,909 (30.26)	11,423 (30.14)
Socio-economic status		
Low	2,430 (18.81)	9,375 (24.74)
Medium	3,842 (29.74)	11,925 (31.46)
High	4,585 (35.5)	11,549 (30.47)
Very high	2,012 (15.58)	4,774 (12.60)
Unknown	48 (0.37)	277 (0.73)

Table 1. Study population characteristics.

We selected for a subject (prediagnostic CD case or control) all medical entities (codes and lab tests) that appeared up to one year prior to the date of CD diagnosis ("index date"). CD diagnosis was determined based on a validated case-ascertainment algorithm [36] with the first CD-related

code or CD-related medication serving as an indicator for the diagnosis date (the earlier of the two). We excluded medical entities from the year prior to diagnosis because they can be related to a late diagnosed CD (a delayed CD diagnosis of a year is common) and thereby can lead to data leakage. Note also that ICD9 codes used in the study for CD prediction and were collected during the prediagnostic period are only non-IBD codes. To capture possible trends in the EHR, we required each subject to have at least five medical entities (similar to [10]). This requirement dramatically reduced the number of subjects for modeling but allowed the collection of enough clinical evidence for the remaining subjects for all experimental settings (using ICD9, lab tests, and combined) (Table 2).

Group	Population	ICD9	Lab tests	Combined
CD	12,917	10,267	6,456	10,327
Control	37,900	24,100	12,461	24,226
Total	50,817	34,367	18,917	34,553

Table 2. Numbers of medical events for the CD and control groups for each experimental setting (only ICD9, only lab tests, and combined).

We focus on a binary classification problem (CD=1 and control=0; Section 5.1). We first randomly divided our entire case population into training and test sets (80%/20%), and then we split the training set again into actual training and validation sets (again, 80%/20%), which gives us a split of 64%–16%–20% for the training, validation, and test sets. Such a split associating a subject with a single set (training, validation, or test) across all experimental settings. That is, if a particular case subject is in the test set, it will be in the test set for all experimental settings. After this split, we associated the control subjects with the sets according to their matched case subjects (i.e., if case subject X was chosen to be in the training set, all control subjects matched to case subject X were also moved to this set). Because our primary goal is to examine how the proposed method affects the classification performance, we made sure that all codes in the test set also appeared in the training set.

To avoid overfitting the model and losing accuracy due to sparsity by using too many codes that are missing for most patients, we preferred the ICD9 codes [3] to their sub-codes. In addition, we removed all administrative codes, and were left with 618 unique ICD9 codes. We focused and included 171 unique lab tests that had at least 15 different real values and avoided others with categorical results. Finally, we removed pregnancy-related lab tests, which are missing for males.

The selection of specific ICD9 codes and lab tests affected the characteristics of patient representation in terms of the numbers of medical entities recorded and unique (not repeated)

medical entities per subject. Not all subjects had every possible code or lab test in their EHR. Also, not all subjects were diagnosed with the same ICD9 code or sent for a particular lab test the same number of times (e.g., a subject can be diagnosed with influenza several times, while another subject might be diagnosed with influenza only one time or not at all). Therefore, it is not easy to fully understand how similar the case and control populations are, based on their EHR characteristics. In Table 3, we present the median values of the numbers of medical entities recorded and of unique codes in a subject's EHR record per experimental setup (we used the median because it is more robust to outliers than the average value). As expected, CD patients had more medical entities (lab tests and diagnoses) and thus also more unique codes than their controls. This is especially true when we move from the ICD9 or Lab models to the Combined model. The additional number of diagnosis codes for CD patients may be related to misdiagnoses during the disease pre-symptomatic period. We also see that due to their increased number, lab tests as medical entities have a larger potential to contribute to subject (patient and control) representation than diagnosis codes, and that, again due to their numbers, diagnosis codes have a larger potential to contribute to patient rather than control representation.

Table 3 also shows that although we kept 618 unique ICD9 codes, only a few appear in each individual EHR. If we were to consider sub-codes, then the sparsity would be more significant, and each sub-code would be so rare that the model could not be able to learn from it.

	Group	**Total**	**Unique**
ICD9	CD	65	26
	Control	50	21
Lab tests	CD	103	39
	Control	87	37
Combined	CD	119	50
	Control	80	38

Table 3. Median values for the total numbers of medical entities and unique medical entities per experimental setting for the period of up to one year prior to the date of CD diagnosis.

Since we use medical entities that took place more than one year before the index date, subjects do not necessarily have an entity precisely one year before the index date, so the time for prediction (from the last entity in the EHR to the index date) varies among subjects. In Table 4, we present descriptive statistics for the time to prediction (i.e., the time in years from the last entity in the EHR to the index date) in the test set for the three settings. We can see that all statistics (except for the maximum value) for the time of prediction using the lab test setting are longer than those in the other settings. This may be explained by the natural order by which, following some illness

condition, the first medical entities for a patient are lab tests followed, in the case of abnormal tests, by a diagnosis. Moreover, while there are some outliers in all settings, we can see that the 75th percentile is still close to the mean value.

	ICD9	Lab tests	Combined
Mean	1.6	2.39	1.58
STD	1.37	1.8	1.36
Median	1.16	1.67	1.15
75th percentile	1.5	2.75	1.47
90th percentile	2.47	4.52	2.39
Max	17.56	16.53	17.58

Table 4. Time to prediction (years) in the test set for the three settings

4.2 Evaluation

First, we evaluate our proposed embedding method according to the actual data with no further considerations. The case-to-control ratio is a characteristic of the data collection process and the clinical protocol, and matching control subjects to case subjects is based only on age and sex. Besides those limitations, we are reluctant to give up any (surplus) data of control subjects. While the actual case-to-control ratio (1:2.2 in our study) does not represent a real-life scenario, because it is a common practice to use data as collected [2,11,16], we will use the actual database as is to evaluate the models in the first scenario.

To address the issue of this arbitrary case-to-control ratio, we propose two other scenarios to evaluate our models by bootstrapping (sampling with replacement) the test set [37,38]. Bootstrapping allows us to control the proportion of cases in the test set and simulate any disease prevalence including a real-life one. For example, when we bootstrap our test set, we can create a balanced set, a real-life disease prevalence, or any other prevalence that has some clinical meaning for a specific population (e.g., high risk population). Thus, in the second scenario, we bootstrap a balanced dataset to evaluate model accuracy in predicting subjects of both classes equally, a scenario that can be used as a reference for the other scenarios.

Since the robustness of any evaluation score of a classifier is subject to the imbalance in the data [39], we want to evaluate, in the third scenario, how effective our proposed embedding method is for the prevalence proportion of CD, which is the fraction of the population with the disease [40]. Since it is not easy to acquire databases for disease such as CD that hold enough data for modeling the actual prevalence, as this is usually very low, most works had to ignore such diseases [10,11].

In the case of CD in Israel the prevalence is approximately 0.28% [34,41], but it might change over time due to different factors [40], such as lifestyle, that increase CD incidence and prevalence with time [41–44]. For example, the risk for CD increases in first-degree relatives (FDRs) of an existing patient. In the case of CD, the FDR prevalence is around 8% (depending on the country), and since this is a more reasonable challenge for routine disease screening than the 0.28% prevalence in the entire population, we focus here on this use case by bootstrapping the database to such a prevalence.

To summarize, we use three prevalence scenarios to evaluate the performance of our embedding method. First is the case-to-control ratio existing in the data (CD prevalence of 26%), and the other two are bootstrap scenarios: the balanced dataset scenario and the FDR CD prevalence in Israel of 8% scenario.

In all three scenarios, we evaluate performance on the test set using the receiver operating characteristic (ROC) curve along with the area under the ROC (ROC-AUC), the recall-precision curve along with the average precision (AP) score, the geometric mean (GM), the F_1 score, and the balanced accuracy (BA) score [39,45,46]. The ROC and recall-precision curves, and the ROC-AUC and AP scores, allow us to evaluate and compare the different models over different thresholds. They also assist us in selecting the desired classification threshold (to classify a "case" if its prediction is greater than the threshold) for a desired tradeoff between the true positive rate ($TPR = \frac{True\ positives}{Total\ positives}$) and false positive rate ($FPR = \frac{False\ positives}{Total\ positives}$) or for a desired tradeoff between the precision and recall. Once a classification threshold has been set, we can calculate scores that suit imbalance classification problems such as ours, e.g., the F_1

$$F_1 = 2\frac{precision \cdot recall}{precision + recall}, \tag{1}$$

GM

$$GM = \sqrt{TPR \cdot TNR}, \tag{2}$$

and BA

$$BA = \frac{TPR + TNR}{2}. \tag{3}$$

The F_1 is the harmonic mean of the precision and recall. The GM aggregates the TPR (recall) and true negative rate ($TNR = \frac{True\ negatives}{Total\ negatives}$, i.e., specificity), which often conflict (high TPR usually leads to low TNR and vice versa). Maximization of the GM addresses the goal of improving the TPR without sacrificing the TNR. The BA is the average of TPR and TNR, so like F_1 and GM, it is suitable for imbalanced data, unlike the regular accuracy score (which is sensitive to the imbalance) [39].

4.3 Implementation

The code was written in Python [47]. The models and evaluation were implemented using PyTorch [48] and Scikit-learn [49]. Bayesian optimization of the hyperparameters [50] was executed using the implementation of [51]. All experiments were executed on a machine with NVIDIA GeForce GTX 1080 Ti GPU.

5 Results

Here, we present the experiments conducted and their results. In Section 5.1, we present the experiment setup, including the model and its optimization, and define the scenarios we tested. In Section 5.2, we present results for the three scenarios: a test set with the original case-to-control ratio as provided in the data, and two bootstrapped test sets, one balancing between the classes, and another suited to the CD prevalence of FDR, simulating a real-life scenario of a high-risk CD population.

5.1 Experiment setup

We used a GRU for all experiments. Hyperparameter tuning was executed by Bayesian optimization [50] to find the optimal configuration in each experimental setting. Each Bayesian optimization execution started with 10 random initial points and 40 additional iterations. We optimized the number of hidden units in the GRU ($2^2 - 2^{10}$), the number of GRU layers (1–4), embedding dimension ($2^2 - 2^8$), dropout rate (0–0.5), learning rate, and weight decay ($10^{-5} - 10^{-1}$). We optimized the binary cross-entropy (BCE) loss,

$$BCE(\boldsymbol{y}, \hat{\boldsymbol{y}}) = -\frac{1}{n}\sum_{i=1}^{n}(w y_i \, log(\hat{y}_i) + (1 - y_i) \, log(1 - \hat{y}_i)), \quad (4)$$

measured on the validation set. y_i and $\hat{y}_i$ represent the true and predicted labels of subject i, respectively, and n is the total number of (validation) subjects over which we compute the loss. We used a weight w and defined it as the ratio between the majority class (control; $y_i = 0$) and the minority class (case; $y_i = 1$) to tackle the imbalance between the classes, so each minority example contributed w times more to the loss than a majority example. The implementation of this loss is part of PyTorch [48]. We also tried to use focal loss [52] to tackle the class imbalance and improve the classification of hard examples. However, it showed similar results to the BCE loss.

After the optimization, we chose the best hyperparameter configuration and the epoch that achieved the best loss on the validation set. Since from this point there was no longer a need for the validation set, we combined the training and validation sets into a single training set and trained the chosen model again on this larger training set. We then reported our results on the test set.

As the length of the EHR sequence representing patients varied among subjects, we limited this length by ℓ. Therefore, we converted each V_s representing subject s to $\widetilde{V_s} = \{\boldsymbol{v}_s^j\}_{j=\max(n_s-\ell,1)}^{n_s}$. For example, for $\ell = 120$, if a subject had n_s>120 ICD9 records, we used only the 120 ICD9 records with the latest timestamps (up to one year before the index [diagnosis] date), i.e., $\widetilde{V_s} = \{\boldsymbol{v}_s^{n_s-119}, \dots, \boldsymbol{v}_s^{n_s}\}$. If a subject had less than $n_s = 120$ records, we used all their records. Since this operation will exclude past diagnoses, we selected ℓ so that it will affect only a small portion of the patients. Table 5 shows that by limiting the sequence length to 120 ($\ell = 120$), we had to exclude past ICD9 records only for 20% of the population (80% of the population had on average a sequence of 114 ICD9 records). While applying $\ell = 120$ for the lab tests and combined experiments would have excluded past records for 40% of the population (60% of the patients had on average 118 and 115 records, respectively; Table 5), setting $\ell = 240$ excluded past records for only 20% of the population (similar to the ICD9 experiment). Note however that while setting $\ell = 240$ allowed the model to use more past records, it did not guarantee better results compared to the $\ell = 120$ scenario because learning long-term dependencies is challenging [53]. Later, in Section 5.2.3, we will see that the AP values for $\ell < 120$ (30, 60, 80, and 100) were lower compared to $\ell = 120$ but not so far (difference of 0.02 from best to worst), and the AP values for ℓ values of 120, 140, and 160 were the same, hence we will focus on $\ell = 120$. In addition, we compared our proposed embedding method to using only the information that a lab test took place (similar to the ICD9 scenario) without using its actual recorded value (see discussion in Section 3).

In total, we present four GRU models (all with $\ell = 120$),

- ICD9: $\widetilde{V_s}$ contains only information about ICD9 codes (baseline).
- Lab – no values: $\widetilde{V_s}$ contains only information that lab tests were made without their recorded values.
- Lab – with values: $\widetilde{V_s}$ contains only information that lab tests were made with their recorded values.
- Combined: $\widetilde{V_s}$ contains information about both ICD9 codes and lab tests with their recorded values.

The hyperparameters chosen by the Bayesian optimization in each experiment are summarized in Table 6.

Percentile	ICD9	Lab tests	Combined
25	20	36	26
50	49	88	80
60	65	118	115
80	114	225	235
85	136	276	290
100	1,456	5,068	5,360

Table 5. Average numbers of past records for selected percentiles of the patients for the three experimental settings.

	ICD9	Lab – no values	Lab – with values	Combined
Embedding dimension	2^3	2^4	2^8	2^7
Dropout rate	0.42	0.39	0.11	0.07
Learning rate	10^{-2}	10^{-2}	10^{-4}	10^{-2}
# of hidden units	2^6	2^6	2^{10}	2^8
# of GRU layers	2	2	2	4
Weight decay	10^{-4}	10^{-5}	10^{-5}	10^{-4}

Table 6. Chosen hyperparameters.

5.2 Experiment results

As described in Section 4.2, we are interested in three scenarios. First (Section 5.2.1) is the common ML scenario that is applied to healthcare data as is, which in our case represents 30% CD prevalence when using ICD9 codes and lab test results and 34% when using only lab tests. Second (Section 5.2.2) is the bootstrapped balanced test set scenario, and third (Section 5.2.3) is the actual prevalence scenario of high-risk CD patients.

We report selected classification metrics by setting thresholds that yield recall levels of 15%, 30%, and 45% in all scenarios. We are not looking for higher recall rates due to the imbalanced nature of the data and because we wish to avoid low precision levels when aiming at high recall rates. We need to keep in mind that low precision might mistakenly send many people to be examined for the suspicion of IBD, which will reduce trust in physicians, HMOs/insurers, and the entire healthcare system in the model. However, low recall might result in many people not being diagnosed on time. Therefore, in this work, we focus on models providing the highest precision given a selected recall.

5.2.1 Original database (31% CD prevalence)

In this scenario, we kept the original case-to-control ratio in the test set. As seen in Table 2, the case-to-control ratio varies between the different experiments. For the ICD9 and combined experiments, the ratio is ~1:2.35, which gives us a prevalence of approximately 30%. However, for the lab tests experiment, the case-to-control ratio is 1:1.9 (prevalence of 34%). To simplify the explanations, we will say that we have an average case-to-control ratio of 1:2.2 and an average prevalence of 31%.

In Figure 2, we see that the ICD9 model is inferior to all other models that used lab tests. By replacing the ICD9 codes with instance of lab tests, the ROC-AUC improves from 0.67 (ICD9 model) to 0.71 ("Lab–no values" model). However, when we added the recorded values of the lab tests, the ROC-AUC improved to 0.77, a higher value even than that of the Combined model (ROC-AUC of 0.72). A possible reason might be that when we combined two information sources (ICD9 codes and lab tests), we used less records from each source by trying to meet the fixed-length representation ($\ell = 120$), which reduced the amount of essential lab test information provided to the model. Similar behavior can also be seen in Figure 3, which presents the recall-precision curve.

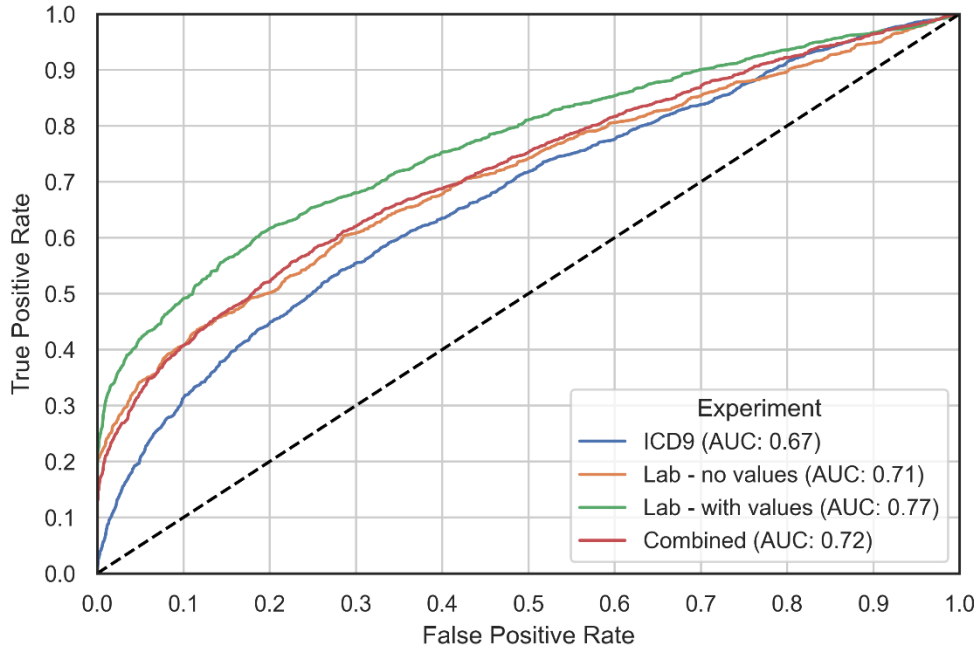

Figure 2. ROC for the original database. The black dashed line indicates a random model.

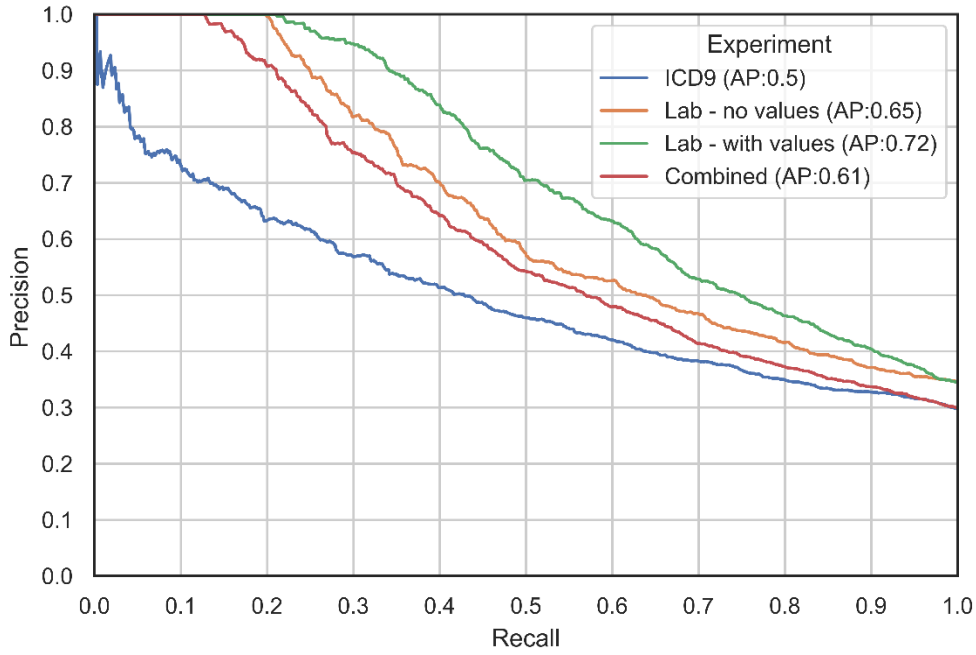

Figure 3. Recall-precision curve for the original database.

Recall	Experiment	Precision	F_1	GM	BA
0.15	ICD9	0.69	0.25	0.38	0.56
	Lab – no values	**1**	**0.26**	**0.39**	**0.58**
	Lab – with values	**1**	**0.26**	**0.39**	**0.58**
	Combined	0.97	**0.26**	**0.39**	**0.58**
0.3	ICD9	0.57	0.39	0.52	0.60
	Lab – no values	0.82	0.44	0.54	0.63
	Lab – with values	**0.95**	**0.46**	**0.55**	**0.65**
	Combined	0.75	0.43	0.54	0.63
0.45	ICD9	0.48	0.47	0.6	0.62
	Lab – no values	0.63	0.53	0.62	0.66
	Lab – with values	**0.76**	**0.57**	**0.63**	**0.69**
	Combined	0.59	0.51	**0.63**	0.66

Table 7. Classification performance measures for the original test set. Best results per measure and recall value are in boldface.

In Table 7, we see the classification report for the three selected recall values. Our embedding method that used lab tests with their values outperformed the other models: only ICD9 codes, only indications of lab result, and combination of codes and the recorded lab results (except for the 15% recall case in which all models that used lab results performed similarly).

5.2.2 50% CD prevalence

In this scenario, we bootstrapped $n = 500$ samples from each of the case and control populations and unified the samples in a single test set with an equal number of case and control subjects. We repeated this procedure 5,000 times and averaged results across the 5,000 bootstrapped test sets.

Because the ROC is robust to imbalance (it presents performance for all thresholds on the probability of CD, which is equivalent to checking all case-to-control ratios), the curve in this scenario is like the one we saw before.

Figure 4 shows that the recall-precision results evaluated on the balanced dataset reveal the same trend seen in Figure 3 for the original imbalanced database (ICD9<Combined<Lab–no values<Lab–with values) but with higher values than those evaluated on the imbalanced database (AP values of: 0.69 vs. 0.5 (ICD9), 0.76 vs. 0.61 (Combined), 0.76 vs. 0.65 (Lab–no values), and 0.81 vs. 0.72 (Lab–with values).

Table 8 presents the average results over the 5,000 bootstrapping iterations. Excluding the 15% recall case, where there was almost no difference between the models (except of the ICD9 model), the model based on lab test values was better than all other models for all performance measures. Together with results from the previous section, these suggest that our embedding method is better regardless of the prevalence of the disease and could also be used in other cases such as the real-life prevalence.

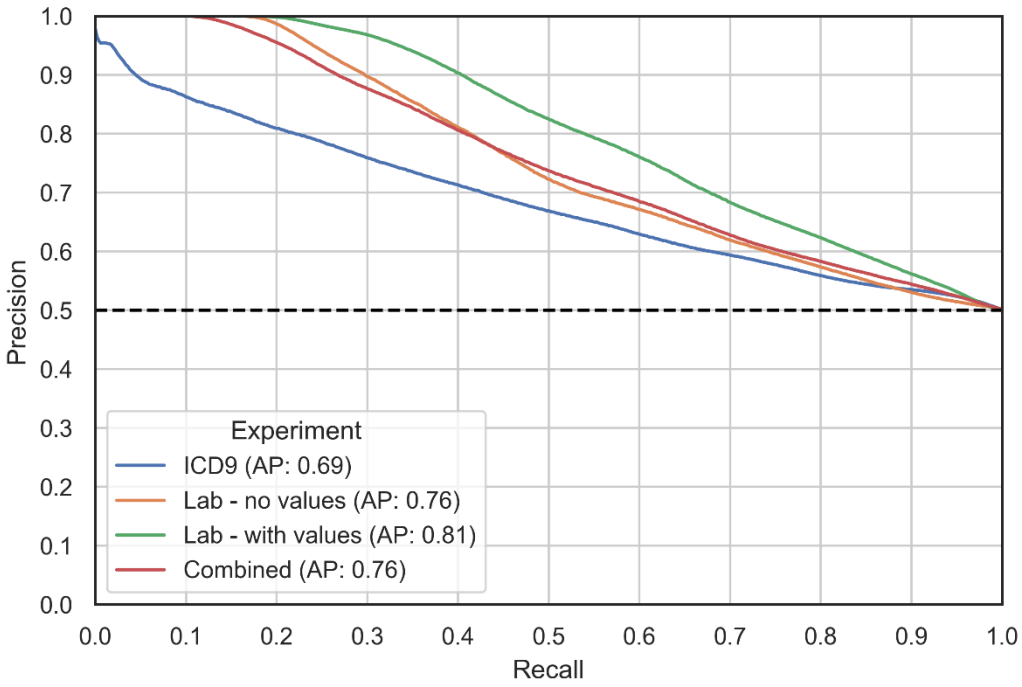

Figure 4. Recall-precision curve for the 50% prevalence scenario. The black dashed line indicates a random model.

Recall	Experiment	Precision	F_1	GM	BA
0.15	ICD9	0.84 (0.04)	0.25 (*)	0.38 (*)	0.56 (*)
	Lab – no values	**1** (*)	**0.26** (*)	0.38 (*)	**0.57** (*)
	Lab – with values	**1** (*)	**0.26** (*)	0.38 (*)	**0.57** (*)
	Combined	0.99 (0.02)	**0.26** (*)	0.38 (*)	**0.57** (*)
0.3	ICD9	0.76 (0.03)	0.44 (0.01)	0.53(0.01)	0.6 (0.01)
	Lab – no values	0.89 (0.03)	**0.46** (*)	0.54 (*)	0.63 (0.01)
	Lab – with values	**0.97** (0.02)	**0.46** (*)	**0.55** (*)	**0.65** (*)
	Combined	0.87 (0.03)	0.45 (*)	0.54 (*)	0.63 (0.01)
0.45	ICD9	0.69 (0.03)	0.54 (0.01)	0.6 (0.01)	0.62 (0.01)
	Lab – no values	0.77 (0.04)	0.57 (0.01)	0.62 (0.01)	0.65 (0.01)
	Lab – with values	**0.86** (0.03)	**0.59** (0.01)	**0.65** (0.01)	**0.69** (0.01)
	Combined	0.77 (0.03)	0.57 (0.01)	0.62 (0.01)	0.66 (0.01)

Table 8. Average (STD) classification performance measures for the 50% prevalence scenario. Best results per measure and recall value are in boldface (indicates STD < 0.01).*

5.2.3 8% CD prevalence

The national prevalence of CD in Israel is approximately 0.28% [34,41]. Screening the overall population with such low prevalence is a challenging task that will also be difficult to evaluate by bootstrapping. If we were to bootstrap the population to 0.28% prevalence, we would need approximately 356 control subjects for each CD case subject. Hence, we would be limited by the population size we can bootstrap. Moreover, executing the screening process over the entire population of Israel might lead to many false alarms, which would reduce trust in the system. A possible solution is to execute this screening process over a population with a higher risk for CD.

One important population with a higher risk for CD is those who have FDR diagnosed with CD. The risk for CD among FDR changes according to several factors [54–59]. One of these factors is the genetic background; we know that there is a higher prevalence of CD in the Jewish population compared to the Arab population in Israel [41]. Also, studies from North America show that the Jewish population has a higher risk for CD when an FDR is also diagnosed [57]. Another factor to consider is the identity of the FDR; the prevalence of CD is higher when there is a diagnosed sibling compared to a diagnosed parent [54,56,57]. It is not trivial, therefore, to estimate the prevalence of CD when an FDR is positive for CD because there is a wide range of results in the literature. FDR of patients with IBD are 3 to 20 times more likely to develop an IBD compared to the general population; for the siblings of a CD patient the risk is up to 35 times more than the population risk [54]; while [55] and [56] identify the risk for CD when there is a positive FDR as 8 and 22.1 times greater, respectively. These differences are because the studies were conducted in different countries prone to genetic differences. We estimate that the prevalence of CD among Jews in Israel with a positive sibling might be between 5% and 10% so we analyzed according to 8% prevalence, which is approximately 28.5 times greater than the prevalence of CD in the entire Israeli population.

While the information about FDR with CD does not exist in our database, it is available for the HMOs that will be able to screen for this specific population. Therefore, in this (third) scenario, we bootstrapped 200 samples from the case population and 2,300 samples from the control population and unified both populations to a single test set of size $N = 2{,}500$ with an 8% prevalence simulating the CD FDR population in Israel. Note that it is the size of the control population that limited us to only 200 samples from the case population. Compared with the 1:2.2 and 1:1 case-to-control ratios for the original and balanced databases, respectively, this 1:11.5 ratio scenario is complicated for our model, and thus, we expect our metrics to be lower compared to those in the previous two sections. We repeated this procedure 5,000 times and report the average scores over these repetitions.

The ROC for this scenario is the same as for the previous scenarios since it is robust to imbalance. Figure 5 shows results of the precision-recall curve that, as expected, are lower than those in Figure

3. Yet the model based on our embedding method outperforms the rest of the models, and all models are better than the random model. Our model's AP (0.46) is almost six times better than that of the random model (which is equal to the prevalence, 0.08) and 2.3 times better than that of the ICD9 model (0.2). Moreover, our model does not make any mistakes up to a recall of approximately 0.17.

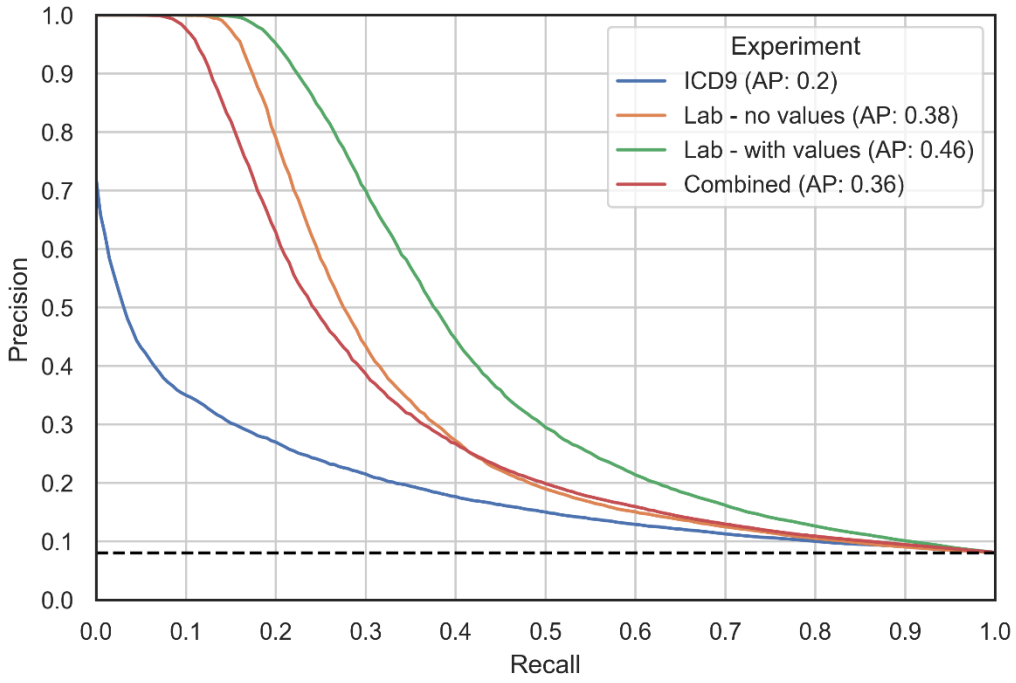

Figure 5. Recall-precision curve for a CD FDR 8% prevalence. The black dashed line indicates a random model.

The results for this scenario are summarized in Table 9. For all recall values and performance measures, the embedding method outperforms all other models (except for similar results with some other models for 15% recall). While the 8% prevalence is low and thus challenging, we have achieved satisfactory performance using our embedding method. For example, in Figure 5, we see that we achieve a precision of 0.8 with a recall of approximately 0.27; this means that for a prevalence of 8,000/100,000 (8%), we will capture 2,160 CD FDR cases with only 540 false alarms.

Recall	Experiment	Precision	F_1	GM	BA
0.15	ICD9	0.31 (0.06)	0.2 (0.01)	**0.38** (*)	0.56 (*)
	Lab – no values	0.99 (0.05)	**0.25** (*)	**0.38** (*)	**0.57** (*)
	Lab – with values	**1** (0.01)	**0.25** (*)	**0.38** (*)	**0.57** (*)
	Combined	0.85 (0.13)	**0.25** (0.01)	**0.38** (*)	**0.57** (*)
0.3	ICD9	0.21 (0.03)	0.25 (0.02)	0.52 (0.01)	0.6 (0.01)
	Lab – no values	0.44 (0.1)	0.35 (0.03)	0.54 (*)	0.63 (0.01)
	Lab – with values	**0.72** (0.11)	**0.42** (0.02)	**0.55** (*)	**0.65** (*)
	Combined	0.38 (0.08)	0.34 (0.03)	0.54 (*)	0.63 (0.01)
0.45	ICD9	0.16 (0.02)	0.24 (0.02)	0.6 (0.01)	0.62 (0.02)
	Lab – no values	0.23 (0.05)	0.3 (0.04)	0.62 (0.01)	0.65 (0.02)
	Lab – with values	**0.36** (0.08)	**0.4** (0.05)	**0.64** (0.01)	**0.69** (0.01)
	Combined	0.23 (0.04)	0.3 (0.03)	0.62 (0.01)	0.66 (0.02)

*Table 9. Average (STD) classification performance measures for the 8% prevalence scenario. Best results per measure and recall value are in boldface. * indicates STD < 0.01.*

5.2.4 Sensitivity analysis

Figure shows precision, recall, and AP values for prediction times of 1-5 years before the diagnosis date based on the Lab–with values data, $\ell = 120$, and the bootstrap method described in Section 5.2.3 with 8% prevalence. When the prediction time is of X years before the diagnosis date, it means that all data up to until X years before the diagnosis were used for the prediction. Earlier prediction times use data of less subjects with less records per subject than later prediction times. Performance for all prediction times was measured by the same test set based on data of only subjects who had clinical entities at least 5 years before the diagnosis date, data which were common to all prediction times. The figure shows that the AP decreases with the prediction time from 0.49 for a year before the diagnosis date to 0.41 for 5 years before the diagnosis date.

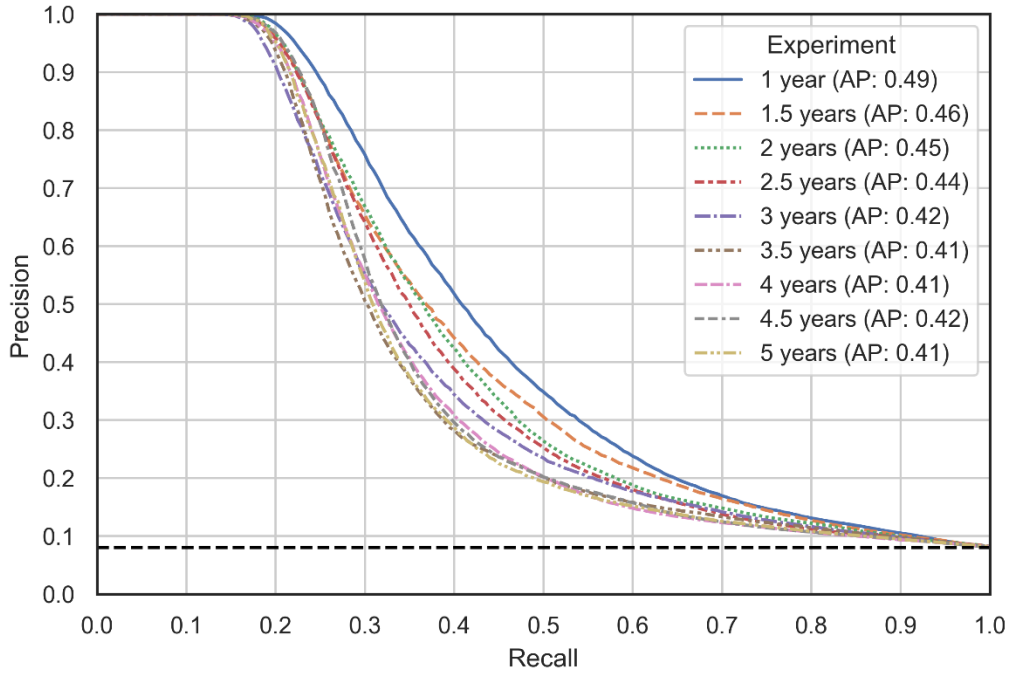

Figure 6. Recall-precision curve for 8% prevalence and prediction times of 1-5 years before diagnosis. The black dashed line indicates a random model.

Table 10 summarizes the results of this experiment for prediction times between 1 and 2.5 years for three levels of recall 0.15, 0.3, and 0.45. As expected, the performance for prediction time of 1 year is the highest regardless of the recall value, and the tradeoff between precision and recall is evident. Because the precision is less sensitive to the prediction time, also the other performance measures, F1, GM, and BA are less sensitive to this time. Note that even for a recall of 0.45 and prediction time of 2.5 years, the precision is 0.28, which is 3.5 times better than if we declare all subjects as positive (for 8% patient prevalence).

Recall	Experiment (years)	Precision	F_1	GM	BA
0.15	1	**1 (0.01)**	**0.25 (*)**	**0.38 (*)**	**0.57 (*)**
	1.5	0.99 (0.03)	**0.25 (*)**	**0.38 (*)**	**0.57 (*)**
	2	1 (0.02)	**0.25 (*)**	**0.38 (*)**	**0.57 (*)**
	2.5	0.99 (0.03)	**0.25 (*)**	**0.38 (*)**	**0.57 (*)**
0.3	1	**0.72 (0.11)**	**0.42 (0.02)**	**0.55 (*)**	**0.65 (*)**
	1.5	0.58 (0.12)	0.4 (0.03)	0.54 (*)	0.64 (*)
	2	0.59 (0.12)	0.4 (0.03)	0.54 (*)	0.64 (0.01)
	2.5	0.56 (0.14)	0.39 (0.03)	0.54 (*)	0.64 (0.01)
0.45	1	**0.36 (0.08)**	**0.4 (0.05)**	**0.64 (0.01)**	**0.69 (0.01)**
	1.5	0.35 (0.07)	0.39 (0.05)	**0.64 (0.01)**	0.68 (0.01)
	2	0.29 (0.06)	0.35 (0.04)	**0.64 (0.01)**	0.67 (0.01)
	2.5	0.28 (0.06)	0.34 (0.04)	0.63 (0.01)	0.67 (0.01)

Table 100. Average (STD) classification performance measures for the 8% prevalence scenario and prediction times between 1 and 2.5 years before diagnosis. Best results per measure and recall value are in boldface (indicates STD < 0.01).*

6 Conclusions and Future Work

In this work, we developed a novel embedding method for deep learning representation of medical codes and real-valued lab tests. It allows easy incorporation of lab test results into deep learning models, which so far was possible only with feature engineering. The suggested method was applied to CD data using a GRU model, and it provided significant accuracy improvement compared to previous methods that used only ICD9 codes. While the ICD9 codes describe a patient's medical state (as usually provided after lab results were measured), they cannot describe disease deterioration because they are indicators providing no real values. Here is where the embedding of real-valued lab tests makes the difference. One of the advantages of the proposed method is that by using a simple preprocessing step, it allows models based exclusively on ICD9 codes or lab results, and on their combination. Since the combination of ICD9 codes (which describe the physician's knowledge and experience) and lab results (which can describe patient's deterioration over time) could not achieve more precise results in this study than just using the lab information, it opens avenues for future research.

This study has several limitations. First, the suggested method do not show embeddings of never-seen-before lab test results. Since we analyzed data of the entire Israel population, we did not miss

any lab test result, but when this method will be applied to other databases, this issue should be considered. Second, the method might be sensitive to different populations or measurement methods and units, which requires training for each population and data source separately. Also, we did not use information about demographics, medications, medical procedures and hospitalization, and lifestyle. A future study could use this information to improve patient representation by a richer embedding. Our database was significantly smaller compared to [10,11], hence, we chose to use GRU, which has fewer parameters than self-attention models have. However, for larger databases than ours the self-attention mechanism will add interpretability, which is crucial in the medical domain. A larger database might also allow to introduce sub-codes.

The main strengths of our suggested embedding method are that it is model-agnostic and disease-agnostic. Model-agnostic because it is a pre-processing stage of the clinical data that may precede the application of any DL model. Disease-agnostic because it embeds lab test results with diagnoses that are both part of the EHR holding information on all diseases in the patient record.

Validation of our method using an independent cohort is necessary before incorporating it in clinical practice to predict CD, for instance in first degree relatives of CD patients which are at particular risk for developing CD.

This study can easily be extended to other deep learning architectures (such as BERT's architecture [10,11,24]) by feeding the embedding result into other than the GRU model. Moreover, our suggest embedding method can also be extended to other medical cases and to non-medical domains.

Acknowledgments

None

Data statement

Access to the data used in this study will be granted upon reasonable request and by emailing gilif@szmc.org.il

Funding sources

This research was partially supported by the Israeli Council for Higher Education (CHE) via the Data Science Research Center, Ben-Gurion University of the Negev, Israel and by a research grant from the Leona M. and Harry B. Helmsley Charitable Trust.

Declaration of competing interest

None

References

[1] Si Y, Du J, Li Z, Jiang X, Miller T, Wang F, et al. Deep representation learning of patient data from electronic health records (EHR): A systematic review. J Biomed Inform 2021;115:103671. https://doi.org/10.1016/j.jbi.2020.103671.

[2] Cheng Y, Wang F, Zhang P, Hu J. Risk prediction with electronic health records: A deep learning approach. 16th SIAM Int Conf Data Min 2016, SDM 2016 2016:432–40. https://doi.org/10.1137/1.9781611974348.49.

[3] Henry OA, Gregory KD, Hobel CJ, Platt LD. Using ICD-9 codes to identify indications for primary and repeat cesarean sections: agreement with clinical records. Am J Public Health 1995;85:1143–6. https://doi.org/10.2105/AJPH.85.8_Pt_1.1143.

[4] Wu J, Roy J, Stewart WF. Prediction modeling using EHR data: Challenges, strategies, and a comparison of machine learning approaches. Med Care 2010;48:S106–13. https://doi.org/10.1097/MLR.0b013e3181de9e17.

[5] Xiao C, Choi E, Sun J. Opportunities and challenges in developing deep learning models using electronic health records data: A systematic review. J Am Med Informatics Assoc 2018;25:1419–28. https://doi.org/10.1093/jamia/ocy068.

[6] Wang L, Wang Q, Bai H, Liu C, Liu W, Zhang Y, et al. EHR2Vec: Representation learning of medical concepts from temporal patterns of clinical notes based on self-attention mechanism. Front Genet 2020;11:1–9. https://doi.org/10.3389/fgene.2020.00630.

[7] Zhu Z, Yin C, Qian B, Cheng Y, Wei J, Wang F. Measuring patient similarities via a deep architecture with medical concept embedding. Proc - IEEE Int Conf Data Mining, ICDM 2017:749–58. https://doi.org/10.1109/ICDM.2016.90.

[8] Yang J, Wang H. Medical concept integrated residual short-long temporal convolutional networks for predicting clinical events. Concurr Comput Pract Exp 2022:e7055.

[9] Zhang T, Chen M, Bui AAT. Diagnostic prediction with sequence-of-sets representation learning for clinical events. Artif Intell Med Conf Artif Intell Med, vol. 12299 LNAI, 2020, p. 348–58. https://doi.org/10.1007/978-3-030-59137-3_31.

[10] Li Y, Rao S, Solares JRA, Hassaine A, Ramakrishnan R, Canoy D, et al. BEHRT: Transformer for electronic health records. Sci Rep 2020;10:1–12. https://doi.org/10.1038/s41598-020-62922-y.

[11] Rasmy L, Xiang Y, Xie Z, Tao C, Zhi D. Med-BERT: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. Npj Digit Med 2021;4. https://doi.org/10.1038/s41746-021-00455-y.

[12] Geffré A, Friedrichs K, Harr K, Concordet D, Trumel C, Braun JP. Reference values: A review. Vet Clin Pathol

2009;38:288–98. https://doi.org/10.1111/j.1939-165X.2009.00179.x.

[13] Horn PS, Pesce AJ. Reference intervals: An update. Clin Chim Acta 2003;334:5–23. https://doi.org/10.1016/S0009-8981(03)00133-5.

[14] Choi E, Bahadori MT, Searles E, Coffey C, Thompson M, Bost J, et al. Multi-layer representation learning for medical concepts. Proc ACM SIGKDD Int Conf Knowl Discov Data Min 2016;13-17-Augu:1495–504. https://doi.org/10.1145/2939672.2939823.

[15] Shickel B, Tighe PJ, Bihorac A, Rashidi P. Deep EHR: A survey of recent advances in deep learning techniques for electronic health record (EHR) analysis. IEEE J Biomed Heal Informatics 2018;22:1589–604. https://doi.org/10.1109/JBHI.2017.2767063.

[16] Choi E, Schuetz A, Stewart WF, Sun J. Medical concept representation learning from electronic health records and its application on heart failure prediction. ArXiv Prepr ArXiv160203686 2016.

[17] Miotto R, Li L, Kidd BA, Dudley JT. Deep patient: An unsupervised representation to predict the future of patients from the electronic health records. Sci Rep 2016;6:1–10. https://doi.org/10.1038/srep26094.

[18] Bengio Y, Courville A, Vincent P. Representation learning: A review and new perspectives. IEEE Trans Pattern Anal Mach Intell 2013;35:1798–828. https://doi.org/10.1109/TPAMI.2013.50.

[19] Wu L, Fisch A, Chopra S, Adams K, Bordes A, Weston J. StarSpace: Embed all the things! 32nd AAAI Conf Artif Intell AAAI 2018 2018:5569–77.

[20] Church KW. Word2Vec. Nat Lang Eng 2017;23:155–62.

[21] Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. 1st Int Conf Learn Represent ICLR 2013 - Work Track Proc 2013:1–12.

[22] Mikolov T, Yih WT, Zweig G. Linguistic regularities in continuous space word representations. Proc. 2nd Work. Comput. Linguist. Lit. CLfL 2013 2013 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. NAACL-HLT 2013, 2013, p. 746–51.

[23] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. Adv Neural Inf Process Syst 2017;2017-Decem:5999–6009.

[24] Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. NAACL HLT 2019 - 2019 Conf North Am Chapter Assoc Comput Linguist Hum Lang Technol - Proc Conf 2019;1:4171–86.

[25] Cho K, van Merriënboer B, Bahdanau D, Bengio Y. On the properties of neural machine translation: Encoder–decoder approaches. Proc SSST 2014 - 8th Work Syntax Semant Struct Stat Transl 2014:103–11. https://doi.org/10.3115/v1/w14-4012.

[26] Baytas IM, Xiao C, Zhang X, Wang F, Jain AK, Zhou J. Patient subtyping via time-aware LSTM networks. Proc ACM SIGKDD Int Conf Knowl Discov Data Min 2017;Part F1296:65–74. https://doi.org/10.1145/3097983.3097997.

[27] Jia Y, Zhou C, Motani M. Spatio-temporal autoencoder for feature learning in patient data with missing observations. Proc - 2017 IEEE Int Conf Bioinforma Biomed BIBM 2017 2017;2017-Janua:886–90. https://doi.org/10.1109/BIBM.2017.8217773.

[28] Chung J, Gulcehre C, Cho K, Bengio Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. ArXiv Prepr ArXiv14123555 2014:1–9. https://doi.org/10.48550/arxiv.1412.3555.

[29] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput 1997;9:1735–80. https://doi.org/10.1162/neco.1997.9.8.1735.

[30] Choi E, Bahadori MT, Schuetz A, Stewart WF, Sun J. Doctor AI: Predicting clinical events via recurrent neural networks. JMLR Workshop Conf Proc 2016;56:301–18.

[31] Kam HJ, Kim HY. Learning representations for the early detection of sepsis with deep neural networks. Comput Biol Med 2017;89:248–55. https://doi.org/10.1016/j.compbiomed.2017.08.015.

[32] Choi E, Schuetz A, Stewart WF, Sun J. Using recurrent neural network models for early detection of heart failure onset. J Am Med Informatics Assoc 2017;24:361–70. https://doi.org/10.1093/jamia/ocw112.

[33] Nawi NM, Atomi WH, Rehman MZ. The effect of data pre-processing on optimized training of artificial neural networks. Procedia Technol 2013;11:32–9. https://doi.org/10.1016/j.protcy.2013.12.159.

[34] Friedman MY, Leventer-Roberts M, Rosenblum J, Zigman N, Goren I, Mourad V, et al. Development and validation of novel algorithms to identify patients with inflammatory bowel diseases in Israel: An epi-IIRN group study. Clin Epidemiol 2018;10:671–81. https://doi.org/10.2147/CLEP.S151339.

[35] Khalilipour BS, Day AS, Kenrick K, Schultz M, Aluzaite K. Diagnostic delay in paediatric inflammatory bowel Disease—A systematic investigation. J Clin Med 2022;11. https://doi.org/10.3390/jcm11144161.

[36] Friedman MY, Leventer-Roberts M, Rosenblum J, Zigman N, Goren I, et al. Development and validation of novel algorithms to identify patients with inflammatory bowel diseases in Israel: an epi-IIRN group study. Clin Epidemiol. 2018 Jun 7;10:671-681. doi: 10.2147/CLEP.S151339. PMID: 29922093; PMCID: PMC5995295.

[37] Hadad B, Lerner B. Domain adaptation from clinical trials data to the tertiary care clinic - Application to ALS. Proc - 19th IEEE Int Conf Mach Learn Appl ICMLA 2020 2020:539–44. https://doi.org/10.1109/ICMLA51294.2020.00090.

[38] Hothorn T, Jung HH. RandomForest4Life: A random forest for predicting ALS disease progression. Amyotroph Lateral Scler Front Degener 2014;15:444–52. https://doi.org/10.3109/21678421.2014.893361.

[39] Tharwat A. Classification assessment methods. Appl Comput Informatics 2018;17:168–92. https://doi.org/10.1016/j.aci.2018.08.003.

[40] Rothman KJ. Epidemiology: An introduction. Oxford university press; 2012.

[41] Stulman MY, Asayag N, Focht G, Brufman I, Cahan A, Ledderman N, et al. Epidemiology of inflammatory bowel diseases in Israel: A nationwide epi-Israeli IBD Research Nucleus Study. Inflamm Bowel Dis 2021;27:1784–94. https://doi.org/10.1093/ibd/izaa341.

[42] Studd C, Cameron G, Beswick L, Knight R, Hair C, Mcneil J, et al. Never underestimate inflammatory bowel disease: High prevalence rates and confirmation of high incidence rates in Australia. J Gastroenterol Hepatol 2016;31:81–6. https://doi.org/10.1111/jgh.13050.

[43] Busingye D, Pollack A, Chidwick K. Prevalence of inflammatory bowel disease in the Australian general practice population: A cross-sectional study. PLoS One 2021;16:e0252458. https://doi.org/10.1371/journal.pone.0252458.

[44] Cosnes J, Gowerrousseau C, Seksik P, Cortot A. Epidemiology and natural history of inflammatory bowel diseases. Gastroenterology 2011;140:1785-1794.e4. https://doi.org/10.1053/j.gastro.2011.01.055.

[45] Zhu M. Recall, precision and average precision. Dep Stat Actuar Sci … 2004;2:1–11.

[46] Fawcett T. An introduction to ROC analysis. Pattern Recognit Lett 2006;27:861–74. https://doi.org/10.1016/j.patrec.2005.10.010.

[47] Van Rossum G, Drake Jr FL. Python 3 reference manual, version 3.7.3. Scotts Valley, CA: CreateSpace; 2009.

[48] Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. PyTorch: An imperative style, high-performance deep learning library. In: Wallach H, Larochelle H, Beygelzimer A, Alché-Buc F, Fox E, Garnett R, editors. Adv. Neural Inf. Process. Syst., vol. 32, Curran Associates, Inc.; 2019.

[49] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in Python. J Mach Learn Res 2011;12:2825–30.

[50] Snoek J, Larochelle H, Adams RP. Practical Bayesian optimization of machine learning algorithms. In: Pereira F, Burges CJC, Bottou L, Weinberger KQ, editors. Adv. Neural Inf. Process. Syst., vol. 4, Curran Associates, Inc.; 2012, p. 2951–9.

[51] Fernando N. Bayesian optimization: Open source constrained global optimization tool for Python. Https://GithubCom/Fmfn/BayesianOptimization 2014.

[52] Lin TY, Goyal P, Girshick R, He K, Dollar P. Focal loss for dense object detection. IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, 2020, p. 318–27. https://doi.org/10.1109/TPAMI.2018.2858826.

[53] Shi Y, Tian Y, Wang Y, Zeng W, Huang T. Learning long-rerm dependencies for action recognition with a

biologically-inspired deep network. Proc. IEEE Int. Conf. Comput. Vis., vol. 2017- Octob, 2017, p. 716–25. https://doi.org/10.1109/ICCV.2017.84.

[54] Kevans D, Silverberg MS, Borowski K, Griffiths A, Xu W, Onay V, et al. IBD genetic risk profile in healthy first-degree relatives of Crohn's disease patients. J Crohns Colitis 2016;10:209–15. https://doi.org/10.1093/ecco-jcc/jjv197.

[55] Moller FT, Andersen V, Wohlfahrt J, Jess T. Familial risk of inflammatory bowel disease: A population-based cohort study 1977-2011. Am J Gastroenterol 2015;110:564–71. https://doi.org/10.1038/ajg.2015.50.

[56] Kim HJ, Shah SC, Hann HJ, Kazmi SZ, Kang T, Lee JH, et al. Familial risk of inflammatory bowel disease: A population-based cohort study in South Korea. Clin Gastroenterol Hepatol 2021;19:2128-2137.e15. https://doi.org/10.1016/j.cgh.2020.09.054.

[57] Ben-Horin S, Avidan B, Yanai H, Lang A, Chowers Y, Bar-Meir S. Familial clustering of Crohn's disease in Israel: Prevalence and association with disease severity. Inflamm Bowel Dis 2009;15:171–5. https://doi.org/10.1002/ibd.20740.

[58] Probert CSJ, Jayanthi V, Hughes AO, Thompson JR, Wicks ACB, Mayberry JF. Prevalence and family risk of ulcerative colitis and Crohn's disease: An epidemiological study among Europeans and South Asians in Leicestershire. Gut 1993;34:1547–51. https://doi.org/10.1136/gut.34.11.1547.

[59] Halfvarson JF, Ludvigsson JF, Bresso F, Askling J, Sachs MC, Olén O. Age determines the risk of familial inflammatory bowel disease—A nationwide study. Aliment Pharmacol Ther 2022:1–10. https://doi.org/10.1111/apt.16938.

Appendix

Similarly to [10,11], we represent each subject's EHR as $V_s = \{\boldsymbol{v}_s^1, \boldsymbol{v}_s^2, \boldsymbol{v}_s^3, .., \boldsymbol{v}_s^{n_s}\}$, where $\boldsymbol{v}_s^j$ is a medical entity such as a diagnosis or a lab test of subject s at time stamp j, and n_s is the total number of medical entities in subject s's EHR. The subject's entities are ordered chronologically; therefore, $\boldsymbol{v}_s^1$ and $\boldsymbol{v}_s^{n_s}$ are the entities with the earliest and latest time stamps for subject s, respectively. If multiple medical entities were recorded in a single visit, then the intra-visit order of these entities will be arbitrary.

To represent an entity of a diagnosis, $\boldsymbol{v}_s^j$ is a one-hot-vector of dimension d_C (ICD9 vocabulary size), with the value 1 in the index corresponding to the ICD9 code and 0 elsewhere. To represent an entity of a lab test (e.g., glucose, albumin, aspartate aminotransferase, etc.), we represent $\boldsymbol{v}_s^j$ by a tuple containing the one-hot-vector of dimension d_L (lab test vocabulary size), with the value 1 in the index corresponding to the type of lab test made and 0 elsewhere, and the result of that test.

This representation is a derivative of the characteristics of the lab test's data. On the one hand, we need to indicate the type of lab test (for instance, glucose or albumin), and on the other hand, we need to indicate the corresponding value measured. We denote the one-hot-vector by $\boldsymbol{x_s^j}$ and the measured real value by α_s^j. Thus, for the lab test, we define $\boldsymbol{v_s^j} = \left(\boldsymbol{x_s^j}, \alpha_s^j\right)$.

To allow both ICD9 codes and lab tests to be represented in the same way, we first unify both ICD9 code and lab test vocabularies, and from this stage, we define that the one-hot-vector $\boldsymbol{x_s^j}$ will be of dimension $d = d_C + d_L$. Secondly, we extend the notation for ICD9 codes in such a way that α_s^j will be a constant number for any $\boldsymbol{v_s^j}$ that represent an ICD9 code. This constant number will allow us to create a mathematical formulation which is similar to the ICD9 codes and lab tests.

The real-valued dense vector $\boldsymbol{x_s^{j^T}} \cdot E$ of dimension d_{emb} is the result of embedding the one-hot-vector $\boldsymbol{x_s^j}$ of dimension d using an embedding matrix $E \in \mathbb{R}^{dXd_{emb}}$. This embedding formulation does not allow the introduction of the real values of lab tests. To extend the formulation, we consider also $\alpha^j \in \mathbb{R}$ (following, we drop the subject subscript *s*), which is the real value for a lab test recorded at time stamp *j*. We would like to use the real value in a way that indicates how much the result deviated from the norm, while keeping the original formulation for ICD9 codes, which do not have real values associated with them. Usually, it might indicate a medical condition when the value recorded deviates from the range associated with a healthy individual (i.e., the reference interval) [12]. However, we do not incorporate information about the reference intervals, as they have some limitations [13]. Instead, we wish to represent the data as is to enable the model to learn this relationship.

The first naïve solution would be the multiplication $\alpha^j \boldsymbol{x^{j^T}} \cdot E$. In this notation, we can set $\alpha^j = 1$ for ICD9 codes, and by doing so, we are not changing the original formulation. However, each lab test has its own characteristics and scale, and the multiplication might yield results of different magnitudes. To address this problem, we apply Z-score scaling [33] per lab test (e.g., an albumin value of 4.5 and an aspartate aminotransferase value of 19 for a subject become 0.46 and -0.51, respectively after Z-score scaling). The formulation then becomes $Z^j \boldsymbol{x^{j^T}} \cdot E$. However, this representation has a significant flaw. After Z-scaling, the mean value of the recorded lab test at timestamp *j* will be zero. Therefore, the result of embedding an entity in which the recorded value of a lab test is the mean value will be $0 \in \mathbb{R}^{d_{emb}}$, and the model would get a zero vector, which does not allow distinction between different types of lab tests. In the example above, if the Z-score of albumin and aspartate aminotransferase were zero, then the embedded representation would be $0 \cdot \boldsymbol{x^{j^T}} = 0$, and the model cannot see the difference between the two (different) clinical entities.

Therefore, we suggest using two embedding matrices to represent lab tests and their real values. The first matrix, $C \in \mathbb{R}^{dXd_{emb}}$, will be used to represent the mean value of the lab test, and the

second matrix, $V \in \mathbb{R}^{dX d_{emb}}$, will represent the Z scores recorded. Considering the scenario described earlier, we suggest a medical record measured at time stamp j will be represented by a one-hot-vector and Z-scored, Z^j, as:

$$\tilde{\boldsymbol{v}}_j = \boldsymbol{x^{jT}} \cdot C + Z^j \boldsymbol{x^{jT}} \cdot V. \qquad (5)$$

In the case α^j is the mean value of lab test recorded at time stamp j ($Z^j = 0$), the second term in (1) vanishes, and the representation will be $\boldsymbol{x^{jT}} \cdot C$; thus, we obtain a unique representation for each lab test. However, if α^j is not the mean value of lab test recorded at time stamp j ($Z^j \neq 0$), then we can consider (1) as a deviation (represented by $Z^j \boldsymbol{x^{jT}} \cdot V$) from the mean value of the lab test recorded at timestamp j (represented by $\boldsymbol{x^{jT}} \cdot C$). In the case of the representation of ICD9 codes using one-hot-vectors, we set $Z^j = 0$ and restore the original representation of medical codes. In practice, the number of rows in V can be smaller than the number in □, as we do not need to store embedding vectors in □ for ICD9 codes because they do not have an associated real value.

We could also think of (1) in the following way: $\boldsymbol{x^{jT}} \cdot C$ represents the existence of lab test, and by adding it to $Z^j \boldsymbol{x^{jT}} \cdot V$, we can discriminate between the cases of no lab test and lab test with $Z^j = 0$. Thus, we train our model to mimic the physician in such a way that they are first aware of the existence of a lab result ($\boldsymbol{x^{jT}} \cdot C$), and afterward, they examine how far this result deviates from its mean value ($Z^j \boldsymbol{x^{jT}} \cdot V$). Furthermore, in practice, physicians compare lab test results to an appropriate reference interval (the "normal" interval of values in which one is considered healthy), so the learned weights in V could be considered as a term that represents the significance of each Z-score for the specific lab test and patient and create a learned reference interval which is more suitable than the known reference interval for the general population. Overall, the result of (1) is a latent representation, we considered as a medical concept, common to both lab tests and diagnoses, which manifests our medical concept embedding method.

In Figure 1 we can see an illustration of the proposed method. For the EHR V of we take each medical entity v_j (concept given at timestamp j) and break it into $\boldsymbol{x^j}$ (the one-hot-vector representation) and α^j (the measured value, or 0 in case of ICD code). $\boldsymbol{x^j}$ is embedded using the embedding matrices C and V resulting in $\boldsymbol{c^j}$ and $\boldsymbol{v^j}$. α^j is Z-scaled and becomes Z^j. Afterwards Z^j is multiplied by $\boldsymbol{v^j}$. and the result is a $\boldsymbol{c^j}$. The result ($\tilde{\boldsymbol{v}}_j$ in (1)) is fed into a GRU layer. We preform this action for each medical entity. In the last step we take the GRU result (denoted by $\tilde{y}$) for the prediction task.