

Medical Diagnosis From Laboratory Tests by Combining Generative and Discriminative Learning

Pengtao Xie

Carnegie Mellon University

ABSTRACT

A primary goal of computational phenotype research is to conduct medical diagnosis. In hospital, physicians rely on massive clinical data to make diagnosis decisions, among which laboratory tests are one of the most important resources. However, the longitudinal and incomplete nature of laboratory test data casts a significant challenge on its interpretation and usage, which may result in harmful decisions by both human physicians and automatic diagnosis systems. In this work, we take advantage of deep generative models to deal with the complex laboratory tests. Specifically, we propose an end-to-end architecture that involves a deep generative variational recurrent neural networks (VRNN) to learn robust and generalizable features, and a discriminative neural network (NN) model to learn diagnosis decision making, and the two models are trained jointly. Our experiments are conducted on a dataset involving 46,252 patients, and the 50 most frequent tests are used to predict the 50 most common diagnoses. The results show that our model, VRNN+NN, significantly ($p < 0.001$) outperforms other baseline models. Moreover, we demonstrate that the representations learned by the joint training are more informative than those learned by pure generative models. Finally, we find that our model offers a surprisingly good imputation for missing values.

Introduction

Misdiagnosis, where a diagnostic decision is made inaccurately, widely occurs. There are approximately 12 million US adults experiencing diagnosis errors every year, half of which could be harmful¹. As many as 40,500 adult patients in an intensive care unit (ICU) die with an misdiagnosis annually². A major source of misdiagnosis is sub-optimal interpretation and usage of clinical data³⁻⁶. Nowadays, physicians are overwhelmed by a large amount of medical data including laboratory tests, vital signs, clinical notes, medication prescriptions, etc. Among all kinds of clinical data, laboratory tests play an important role. According to American Clinical Laboratory Association, laboratory tests guide more than 70% of the diagnostic decisions. Unfortunately, comprehensively understanding the laboratory test results and discovering the underlying clinical implications are not easy. Incorrect interpretation of laboratory tests is a major breakdown point in the diagnostic process³⁻⁶.

The reasons why laboratory tests are difficult to understand are two-fold. First, missing values are pervasive. It is typical that at a certain time point, only a small subset of laboratory tests are examined, leaving the values of most tests missed. These data missing prevents physicians from getting a full picture of patients' clinical states, leading to sub-optimal decisions. Second, the laboratory test values have a complex multivariate time-series structure: during an in-hospital stay, multiple laboratory tests are examined at a particular time, and the same test may be examined multiple times at different time points. These multivariate temporal data exhibits complicated patterns along the dimensions of both time and tests. Learning these patterns is highly valuable for diagnosis, but it is technically challenging.

In this work, we study how to leverage the ability of machine learning (ML) in automatically distilling patterns from complex, noisy, incomplete and irregular laboratory test data to address the above-mentioned issues, and build an end-to-end diagnostic model to assist with diagnosis decisions. Previous studies have applied ML to perform diagnosis based on laboratory tests. In these approaches, three major tasks – handling missing values, discovering patterns from multivariate time-series and predicting diseases – were often conducted separately. However, these three tasks are tightly coupled and can mutually benefit each other. On one hand, better imputation of missing values leads to the discovery of more informative patterns, which boosts the accuracy of diagnosis. On the other hand, during model training, the supervision of diagnosis provides a guidance of pattern discovery, which further influences the imputation of missing values, tailoring the discovered patterns and imputed values to be suitable for a diagnosis task. Performing these tasks separately fails to consider their synergistic relationships and hence leads to sub-optimal solution. Another limitation exists in previous studies is that they often proposed in a discriminative structure that cannot well address the missing value problem and learn generalizable patterns in principle.

Contributions

In this paper, we develop an end-to-end deep neural model to perform diagnosis based on laboratory tests. Our model seamlessly integrates three tasks together: imputing missing values, discovering patterns from multivariate time-series data and predicting diseases, and perform them jointly. Our model combines two major learning paradigms in machine learning: generative learning and discriminative learning, where the generative learning component is utilized to deal with missing values and discover robust and generalizable patterns and the discriminative learning component is used for predicting diseases based on the patterns discovered in generative learning. We evaluate the proposed model on 46,252 patient visits in the ICU and demonstrate that our model achieves (1) significantly ($p < 0.001$) better diagnosis performance than baseline models, (2) better imputation of missing values, and (3) better discovery of patterns from the laboratory test data.

Related Works

Lasko *et al.* proposed to use a Gaussian process to model the longitudinal Electronic Medical Records (EHR) and used a standard auto-encoder (AE) to learn hidden features from raw inputs⁷, and Ghassemi *et al.* further introduced a multi-task Gaussian process to model the clinical data⁸. Che *et al.* and Miotto *et al.* took advantage of denoising auto-encoder (DAE)⁹ to learn hidden representations and then used these representations to make diagnose^{10,11}. In recent years, recurrent neural network (RNN) has demonstrated its superiority in modeling longitudinal data, like natural language¹² and speech signals¹³. RNN has also been utilized for medical diagnosis. Lipton *et al.* proposed a medical diagnosis model based on long short-term memory (LSTM)¹⁴ recurrent neural networks and obtained performance better than some strong baselines^{15,16}. In this work, missing values were addressed by heuristic forward- and back-filling. Their later work continued the LSTM-based diagnosis, but used a missing value indicator as a part of the inputs. They found missing value patterns can help with diagnosis performance¹⁷. Choi *et al.* proposed a 'Doctor AI' that used previous diagnoses to predict future diagnosis¹⁸. In their another work, a neural attention mechanism was further introduced¹⁹. Che *et al.* proposed a diagnosis model based on stacked DAE and LSTM, and Gradient Boosting Trees were introduced to make the learned features more interpretable²⁰. Their later work modified the structure of Gated Recurrent Units (GRU) to deal with incomplete inputs²¹.

All the above work are based on discriminative models, which cannot address the missing value problem very well in principle. A generative model usually works better in this aspect. A typical generative model is the Gaussian Mixture Model (GMM), with which missing values can be easily addressed by the Expectation-Maximum (EM) algorithm^{22,23}. For instance, Marlin *et al.* employed GMMs to discovery patterns from clinical data and conducted mortality prediction²⁴. These generative models, however, are mostly shallow, linear and Gaussian. Recently, researchers proposed several deep generative models, e.g., variational auto-encoder (VAE)²⁵ and variational RNN (VRNN)²⁶. Compared to conventional generative models, VAE and VRNN can model more complex conditional distributions, hence representing more complex patterns. Our work will utilize these deep generative models. However, generative models are not task-oriented²⁷⁻²⁹. Therefore, we propose an end-to-end approach that combines the advantages of both types of models and train them in a joint fashion. This can be seen as either a generative learning with the discriminative target as a guidance, or a discriminative learning with the generative target as a regularization.

Data Preparation

The data used in this study, opened by MIMIC-III³⁰, is publicly available. It was derived from the laboratory tests of 46,252 patients. It contains both in-hospital and outpatient records. Each in-hospital episode has 1 to 39 corresponding ICD-9 (International Classification of Diseases, 9th Edition) codes, and in this study, only the primary diagnoses will be considered. It amounts to 2,789 different diagnoses and 513 unique laboratory tests. As some diagnoses and tests are quite rare, we limit our study to the 50 most frequent diagnoses and the 50 most frequent laboratory tests. Detailed descriptions of these 50 most frequent diagnoses and 50 most frequent laboratory tests are shown in Table 4 and Table 5, respectively. For each in-hospital episode, we group the test results by day, and finally, we get 30,931 day sequences of laboratory test records, and each of them is labeled by a disease ID, from 0 to 49. The shortest sequence is in 2-day length, while the longest sequence is in 171-day length. Only 5 sequences have lengths over 100, so we set the max length as 100, i.e. for the sequences longer than 100 days, we only consider the records of latest 100 days. Figure 1 shows the number of samples over the 50 disease IDs. We random split the dataset for 5 times, and each time we keep the proportion of training (Train), development (Dev), testing (Test) sets as 65%:15%:20%. Hence, the numbers of samples in these three sets are 20,105, 4,640, 6,186, respectively.

Some of the tests are valued by discrete categories, like "ABNORMAL" and "NORMAL". We change these categories to integers, like 0 for "ABNORMAL", 1 for "NORMAL". Test results are normalized by the Z-normalization, i.e. values of each test are subtract by the mean and divided by the standard deviation. Note that a patient cannot do every test every day, so missing values are pervasive in our data. In Figure 2, we present an example of a patient's laboratory test records. It can be seen that there are a lot of missing values. A simple statistics show that in the whole dataset, the average missing value rate is about

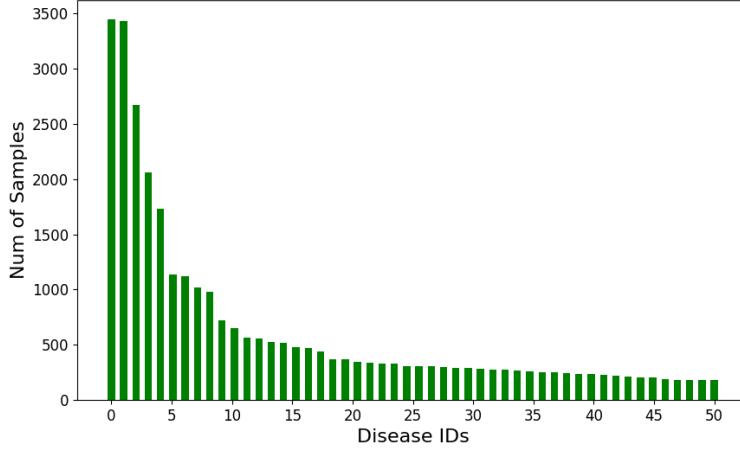


Figure 1. Number of Samples over Disease IDs

54%, i.e. on average, only 27 of the 50 laboratory tests have values in a patient’s one-day record. In our experiments, initially we impute the missing values with 0. After applying Z-normalization, the mean of each test’s values changes to zero. So, zero imputation equals to mean imputation. Moreover, since our models are situated in neural network framework, zero inputs will not introduce additional bias in computation. Note that in baseline models, this zero-imputation behaves as the solution of missing value problem, while in our models, it behaves as indicators of the missing values, and missing values will be further addressed by deep generative models.

Model Architectures

Given N i.i.d data $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N$, each \mathbf{x}_n is a temporal sequence of T in length, i.e. $\mathbf{x}_n = \{x_1, x_2, \dots, x_T\}$. Any $t \in [1, T]$, $x_t \in \mathbb{R}^M$, where M is the dimension of input features. Meanwhile, there is a class label y_n for each \mathbf{x}_n , and our purpose is to predict the class labels accurately. Specifically, \mathbf{x}_n is the longitudinal laboratory test records of a patient, and each x_t is a one-day record. y_n is the primary diagnosis. We propose two models in this study, denoted by VAE+NN and VRNN+NN, the former a static model that can demonstrate the contribution of deep generative models, while the latter a temporal model that extends the deep generative learning approach to learn long-term temporal dependency.

VAE+NN

In this model, we address the temporal records by simply averaging the vectors at all time points, i.e. $\tilde{x}_n = \text{average}(x_1, x_2, \dots, x_T)$. Although the averaging operation alleviates the missing value problem to some extent, the rate of the missing values in \tilde{x}_n is still high (about 29%). To further deal with the missing values and capture the complex patterns in data, we propose a VAE+NN model, where a VAE is the generative model used to handle missing values and discover patterns, and a standard neural network (NN) is used as a classifier, as shown in Figure 3 (a).

Referring to the idea of VAE²⁵, suppose that \tilde{x}_n is generated from a latent variable z_n , then the joint probability is defined as:

$$p(X, Z) = \prod_{n=1}^N p(z_n) p(\tilde{x}_n | z_n). \quad (1)$$

Let the prior over z_n is a centered isotropic multivariate Gaussian distribution, i.e. $p(z_n) = \mathcal{N}(z; \mathbf{0}, \mathbf{I})$. Further assume that with a fixed z_n , \tilde{x}_n also follows a Gaussian distribution, and suppose different dimensions of \tilde{x}_n are independent, then the generation of \tilde{x}_n is defined as:

$$p(\tilde{x}_n | z_n) = \mathcal{N}(\mu_x, \text{diag}(\sigma_x^2)), \text{ where } [\mu_x, \sigma_x] = \varphi_\tau^{\text{dec}}(z_n), \quad (2)$$

where μ_x and σ_x are parameters of the conditional distribution. $\varphi_\tau^{\text{dec}}$ is a feed-forward neural network. Then, the expectation of posterior distribution $p(z_n | \tilde{x}_n)$ is fed into a discriminative network φ_d to perform classification, formulated as follows:

$$p(\hat{y}_n | \mathbf{x}_n) = \text{softmax}(\varphi_d(E(z_n | \tilde{x}_n))). \quad (3)$$

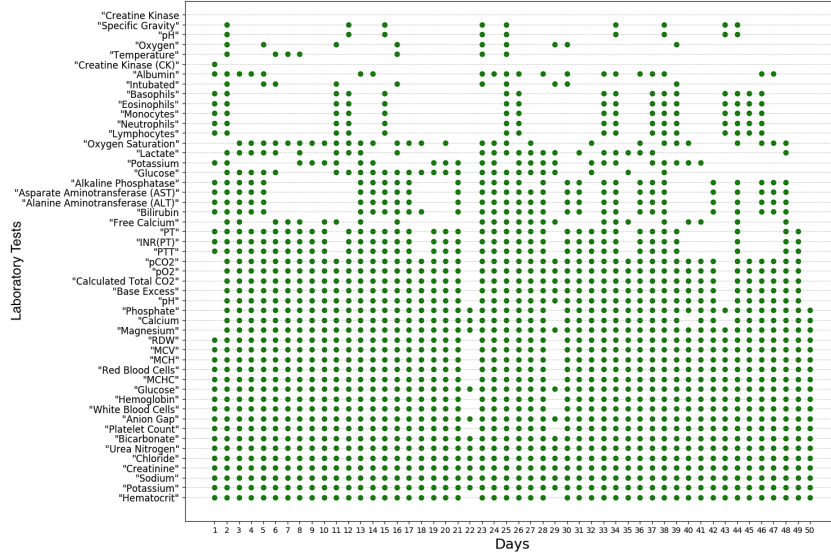


Figure 2. An example of a patient’s laboratory test records. The y axis corresponds to 50 laboratory tests we used. The x axis denotes the time of the records. The green dot means there is a value, otherwise it is missing.

where softmax is the softmax function, written by $\frac{e^k}{\sum_k e^k}$.

VRNN+NN

To better capture the long-term dependency of clinical records, previous work introduced RNN into this task^{15,17}. However, the internal transition structure of RNN is entirely deterministic which cannot efficiently model the complexity in highly structured sequential data. Therefore, Chung *et al.* extended the VAE idea to a recurrent framework for modeling high-dimensional sequences, leading to a variational RNN (VRNN)²⁶. We propose a VRNN+NN model as shown in Figure 3 (b), which involves a VRNN to generate sequential hidden features and an NN model to make decisions based on the average of these hidden features.

Assuming that the generation of each x_t in \mathbf{x}_n is conditioned on a latent variable z_t and the previous state h_{t-1} , the joint probability is defined as follows.

$$p(X, Z) = \prod_{n=1}^N \prod_{t=1}^T p(z_t; h_{t-1}) p(x_t | z_t; h_{t-1}), \quad (4)$$

where $p(z_t; h_{t-1})$ is the prior of the latent variable and $p(x_t | z_t; h_{t-1})$ is the conditional probability of the observations. We assume that the prior distribution $p(z_t; h_{t-1})$ follows a Gaussian distribution, and different dimensions of z_t are independent, i.e. the covariance matrix is diagonal. Therefore, the prior distribution is defined as follows.

$$p(z_t; h_{t-1}) = \mathcal{N}(\mu_{0,t}, \text{diag}(\sigma_{0,t}^2)), \text{ where } [\mu_{0,t}, \sigma_{0,t}] = \varphi_\tau^{\text{prior}}(h_{t-1}), \quad (5)$$

where $\mu_{0,t}$ and $\sigma_{0,t}$ represents the mean and the standard variance of the prior distribution, respectively. Again, assume different dimensions of x_t are independent, then the conditional probability for the observation is written by:

$$p(x_t | z_t; h_{t-1}) = \mathcal{N}(\mu_{x,t}, \text{diag}(\sigma_{x,t}^2)), \text{ where } [\mu_{x,t}, \sigma_{x,t}] = \varphi_\tau^{\text{dec}}(z_t, h_{t-1}), \quad (6)$$

where $\mu_{x,t}$ and $\sigma_{x,t}$ are parameters of the generating distribution. $\varphi_\tau^{\text{prior}}$ and $\varphi_\tau^{\text{dec}}$ are neural networks. The hidden states are updated by LSTM¹⁴ cells denoted by f_θ , so the recurrence equation is defined as:

$$h_t = f_\theta(x_t, z_t, h_{t-1}). \quad (7)$$

We take the average of the hidden states, $\tilde{h}_n = \text{average}(h_1, h_2, \dots, h_T)$, as the input of a discriminative NN φ_d .

$$p(\hat{y}_n | \mathbf{x}_n) = \text{softmax}(\varphi_d(\tilde{h}_n)) \quad (8)$$

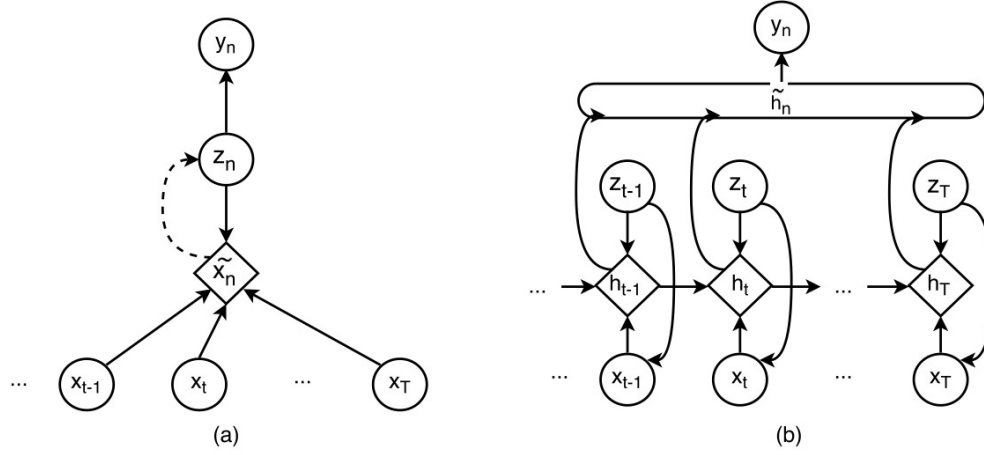


Figure 3. Model Architectures of Our Models: (a) is the architecture of VAE+NN, and (b) is the architecture of VRNN+NN.

Inference & Learning

VAE+NN

The involvement of neural networks makes the true posterior $p(z_n|\tilde{x}_n)$ is intractable. Resorting to variational inference methods, we use $q(z_n|\tilde{x}_n)$ to approximate the true posterior which is defined in the fashion of:

$$q(z_n|\tilde{x}_n) = \mathcal{N}(\mu_z, \text{diag}(\sigma_z^2)), \text{ where } [\mu_z, \sigma_z] = \varphi_\tau^{\text{enc}}(\tilde{x}_n), \quad (9)$$

where $\varphi_\tau^{\text{enc}}$ is a feed-forward neural network. Denote the parameters involved in VAE as θ_g (g for generative), learning target is $\hat{\theta}_g = \text{argmax}_{\theta_g} (\sum_n \log p(\tilde{x}_n|\theta_g))$ which can be redefined as the lower bound of $\log p(\tilde{x}_n|\theta_g)$:

$$\mathcal{L}_g(\theta_g; X) = \sum_n^N (-KL(q(z_n|\tilde{x}_n)||p(z_n)) + E_{q(z_n|\tilde{x}_n)}[\log p(\tilde{x}_n|z_n)]). \quad (10)$$

Since we assume different dimensions of \tilde{x} are independent, $p(\tilde{x}_n|z_n) = \prod_m p(\tilde{x}_{n,m}|z_n)$. Missing items in \tilde{x}_n should be dropped in the probability, i.e. $p(\tilde{x}_n|z_n) = \prod_{m \notin \tilde{\mathbf{m}}} p(\tilde{x}_{n,m}|z_n)$, $\tilde{\mathbf{m}}$ is the indexes of missing items. So, the generative loss can be rewritten as follows:

$$\mathcal{L}_g(\theta_g; X) = \sum_n^N (-KL(q(z_n|\tilde{x}_n)||p(z_n)) + E_{q(z_n|\tilde{x}_n)}[\sum_{m \notin \tilde{\mathbf{m}}} \log p(\tilde{x}_{n,m}|z_n)]). \quad (11)$$

Also, there is a discriminative loss from classification, which is the cross entropy between the true disease ID y_n and model predicted posterior probability. θ_d (d for discriminative) denotes the parameters in discriminative network.

$$\mathcal{L}_d(\theta_d; X, Y) = \sum_n \log p(y_n|\mathbf{x}_n). \quad (12)$$

Therefore, the overall loss function of the VAE+NN model is given by the sum of the two costs, denote $\theta = \{\theta_g, \theta_d\}$.

$$\mathcal{L}(\theta; X, Y) = \mathcal{L}_d(\theta_d; X, Y) + \eta * \mathcal{L}_g(\theta_g; X), \quad (13)$$

where η is a trade-off parameter of generative loss, because our final target is classification, and thus the generative loss can be taken as a regularization term.

VRNN+NN

Similarly, the true posterior $p(z_t|x_t)$ is intractable, $q(z_t|x_t)$ is used to approximate it. $q(z_t|x_t)$ and generative loss $\mathcal{L}_g(\theta_g; X)$ are defined as follows:

$$q(z_t|x_t) = \mathcal{N}(\mu_{z,t}, \text{diag}(\sigma_{z,t}^2)), \text{ where } [\mu_{z,t}, \sigma_{z,t}] = \varphi_\tau^{\text{enc}}(x_t, h_{t-1}), \quad (14)$$

$$\mathcal{L}_g(\theta_g; X) = \sum_n \sum_t^T (-KL(q(z_t|x_t)||p(z_t)) + E_{q(z_t|x_t)}[\log p(x_t|z_t)]). \quad (15)$$

Also, since different dimensions of x_t are independent, $p(x_t|z_t) = \prod_m p(x_{t,m}|z_t)$. Denote the indexes of missing values in x_t is $\tilde{\mathbf{m}}_t$, then:

$$\mathcal{L}_g(\theta_g; X) = \sum_n \sum_t^T (-KL(q(z_t|x_t)||p(z_t)) + E_{q(z_t|x_t)}[\sum_{m \notin \tilde{\mathbf{m}}_t} \log p(x_{t,m}|z_t)]), \quad (16)$$

The overall loss function is also defined as the combination of generative loss and discriminative loss.

$$\mathcal{L}_d(\theta_d; X, Y) = \sum_n \log p(y_n|\mathbf{x}_n), \quad (17)$$

$$\mathcal{L}(\theta; X, Y) = \mathcal{L}_d(\theta_d; X, Y) + \eta * \mathcal{L}_g(\theta_g; X). \quad (18)$$

Baselines

In this section, we will build several baseline models for the comparative study, denoted by NN, AE+NN, RNN+NN, respectively. NN and AE+NN models are used to compare with the VAE+NN model, we'd like to see if deep generative models can have a better performance when representing single feature vectors. RNN+NN model is similar as the model structure in previous studies^{15,17}, and it is used to compare with our VRNN+NN model.

NN

As shown in Figure 4 (a), 'NN' model is a simple multi-layer perceptron (MLP). The model and the learning target are defined as follows:

$$p(\hat{y}_n|\mathbf{x}_n) = \text{softmax}(\varphi_d(\tilde{x}_n)), \quad (19)$$

$$\mathcal{L}(\theta; X, Y) = \sum_n \log p(y_n|\mathbf{x}_n). \quad (20)$$

AE+NN

To demonstrate the superiority of VAE, we present an AE+NN baseline that is based on the standard auto-encoder, shown in Figure 4 (b). AE is similar to VAE, but its structure is deterministic, so it is less generative. In this model, we also combine the loss from AE and NN. We use mean squared error (MSE) as the training objective for AE. Also, missing items are not included in the loss function. In summary, the model and the learning target are defined as follows:

$$z_n = \varphi_\tau^{enc}(\tilde{x}_n), \quad (21)$$

$$\hat{x}_n = \varphi_\tau^{dec}(z_n), \quad (22)$$

$$p(\hat{y}_n|\mathbf{x}_n) = \text{softmax}(\varphi_d(z_n)), \quad (23)$$

$$\mathcal{L}(\theta; X, Y) = \sum_n (\log p(y_n|\mathbf{x}_n) + \eta * \sum_{m \notin \tilde{\mathbf{m}}} (\tilde{x}_{n,m} - \hat{x}_{n,m})^2). \quad (24)$$

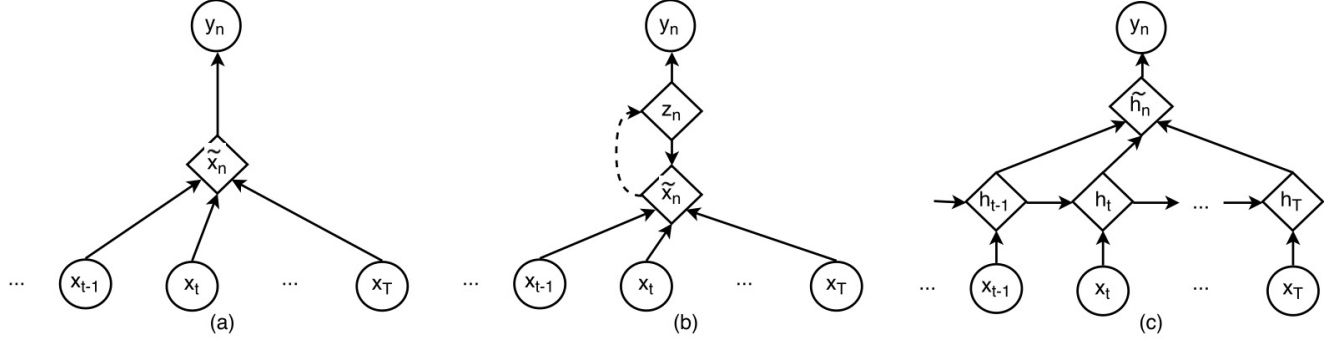


Figure 4. Model Architectures of Baseline Models: (a) is the architecture of NN, (b) is the architecture of AE+NN, and (c) is the architecture of RNN+NN.

RNN+NN

The RNN+NN model is shown in Figure 4 (c). In this model, an RNN processes the raw temporal features, and the average of the hidden state \tilde{h}_n is used as the input of the NN. f_θ also denotes the recurrent computation of the LSTM cells. The model can be formulated as follows:

$$h_t = f_\theta(x_t, h_{t-1}), \quad (25)$$

$$p(\hat{y}_n | \mathbf{x}_n) = \text{softmax}(\varphi_d(\tilde{h}_n)), \quad (26)$$

$$\mathcal{L}(\theta; X, Y) = \sum_n^N \log p(y_n | \mathbf{x}_n). \quad (27)$$

Experiments

Implement Details

In our experiments, models are implemented on Tensorflow r1.0. All φ_τ and φ_d are feed-forward neural networks with one hidden layer and ReLU activations. The size of hidden layer is set to 64. We use Adam³¹ as the optimizer, where the learning rate is 0.0005 and the learning rate decay is 0.99. The trade off parameter η is set to 0.5 in all the experiments.

Evaluation Metrics

F1 score

Generally, F1 score is defined as $\frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}$, where *precision* is the number of correct positive results divided by the number of all positive results, and *recall* is the number of correct positive results divided by the number of positive results that should have been returned. Since our task is a 50-class classification, we use three kinds of F1 scores: micro-F1, macro-F1, macro-F1-weighted (macro-F1-w). Micro-F1 is computed from flatten \hat{Y} and Y . Macro-F1 is the arithmetic mean of F1 scores of different classes. Macro-F1-weighted is the weighted mean of F1 scores of different classes, and the weight of class i is defined as $\frac{n_i}{N}$, where n_i is the number of samples of class i in testing set. F1 score ranges from 0 to 1, and the best performance achieves when F1 score equals 1. We simulate a blind prediction, i.e. equal probabilities on different classes, the results are micro-F1 is 0.111, macro-F1 is 0.004, macro-F1-weighted is 0.022.

AUC

AUC is the area under the ROC curve. Similarly, we also use three AUCs in evaluation: micro-AUC, macro-AUC, macro-AUC-weighted (macro-AUC-w). AUC also ranges from 0 to 1, and the best performance achieves when AUC equals 1. And in the blind prediction, all AUCs equal 0.5.

MSE

Mean Squared Error (MSE) measures the quality of an estimator. It equals the average of the squares of the differences between true values and estimated values. Assume we have N variables with values $\{v_0, v_1, \dots, v_n, \dots, v_N\}$, and an estimator estimates the values of these N variables as $\{v'_0, v'_1, \dots, v'_n, \dots, v'_N\}$, then $MSE = \frac{1}{N} \sum_{n=0}^N (v'_n - v_n)^2$. Best estimation performance arrives when MSE equals 0. In this study, we use MSE to evaluate missing value imputation performance of different imputation methods.

Diagnosis performances of different models

Model	Micro-F1	Macro-F1	Macro-F1-w	Micro-AUC	Macro-AUC	Macro-AUC-w
NN	0.376 ± 0.004	0.221 ± 0.003	0.347 ± 0.005	0.939 ± 0.001	0.905 ± 0.001	0.913 ± 0.001
AE+NN	0.366 ± 0.004	0.219 ± 0.002	0.344 ± 0.002	0.938 ± 0.001	0.903 ± 0.002	0.912 ± 0.001
VAE+NN	0.374 ± 0.003	0.226 ± 0.005	0.352 ± 0.004	0.941 ± 0.000	0.908 ± 0.001	0.916 ± 0.001
RNN+NN	0.395 ± 0.004	0.248 ± 0.003	0.373 ± 0.003	0.945 ± 0.003	0.918 ± 0.004	0.923 ± 0.003
VRNN+NN	0.426 ± 0.002	0.291 ± 0.006	0.407 ± 0.002	0.958 ± 0.000	0.937 ± 0.000	0.938 ± 0.001
VRNN+NN (early)	0.422 ± 0.005	0.285 ± 0.006	0.403 ± 0.004	0.957 ± 0.001	0.935 ± 0.001	0.937 ± 0.001
Performance of features derived from different models (with a simple NN classifier)						
$E(z_n)$ (VAE)	0.363 ± 0.004	0.195 ± 0.004	0.326 ± 0.003	0.936 ± 0.001	0.896 ± 0.003	0.906 ± 0.002
$E(z_n)$ (VAE+NN)	0.380 ± 0.004	0.228 ± 0.004	0.353 ± 0.002	0.943 ± 0.001	0.911 ± 0.003	0.918 ± 0.002
\tilde{h}_n (VRNN)	0.406 ± 0.003	0.261 ± 0.003	0.381 ± 0.003	0.953 ± 0.000	0.928 ± 0.001	0.930 ± 0.001
\tilde{h}_n (VRNN+NN)	0.427 ± 0.003	0.297 ± 0.004	0.410 ± 0.003	0.958 ± 0.001	0.936 ± 0.001	0.937 ± 0.000
Performance with different missing value imputation methods						
RNN+NN(zero)	0.395 ± 0.005	0.248 ± 0.003	0.374 ± 0.002	0.945 ± 0.003	0.918 ± 0.004	0.923 ± 0.003
RNN+NN(last&next)	0.385 ± 0.002	0.233 ± 0.003	0.360 ± 0.002	0.941 ± 0.001	0.912 ± 0.002	0.918 ± 0.001
RNN+NN(row mean)	0.393 ± 0.003	0.243 ± 0.005	0.369 ± 0.001	0.945 ± 0.002	0.917 ± 0.003	0.923 ± 0.002
RNN+NN(NOCB)	0.384 ± 0.003	0.231 ± 0.002	0.359 ± 0.001	0.941 ± 0.002	0.911 ± 0.003	0.917 ± 0.002
VRNN+NN	0.426 ± 0.002	0.291 ± 0.006	0.407 ± 0.002	0.958 ± 0.000	0.937 ± 0.001	0.938 ± 0.001

Table 1. Three sets of diagnosis performances. The top set contains the performances of our models and baselines; the middle set shows the performances of a NN model with features derived from different models; the bottom set compares performance of VRNN+NN and RNN+NN with typical heuristic imputation methods. All results are presented in the "mean value ± standard deviation" fashion.

Performance comparison

Comparison	Micro-F1	Macro-F1	Macro-F1-w	Micro-AUC	Macro-AUC	Macro-AUC-w
VAE+NN vs. NN		*		*	*	**
VAE+NN vs. AE+NN			*	*	*	*
RNN+NN vs. VAE+NN	**	**	**		**	**
VRNN+NN vs. RNN+NN	***	***	***	***	***	***
$E(z_n)$ (VAE+NN) vs. $E(z_n)$ (VAE)	**	***	***	**	***	***
\tilde{h}_n (VRNN+NN) vs. \tilde{h}_n (VRNN)	**	***	***	***	***	***

Table 2. P-values of paired t-test on diagnosis performances. Note that: *** ($p < 0.001$), ** ($p < 0.01$), * ($p < 0.05$).

Results

Table 1 shows the diagnosis performances of three sets of experiments. The top set of values are the performances of different models on the diagnosis task, measured by different variants of F1 values and AUCs. "VRNN+NN (early)" is the VRNN+NN model only using patients' first ten-day records. We would like to see if VRNN+NN model could do early prediction of diseases. Additionally, to test if the joint training can result in better representations compared to unsupervised generative models, the representations derived from VAE, VAE+NN, VRNN and VRNN+NN are used to train a new NN model for diagnosis decision. Performances are presented in the middle set. Finally, in the bottom set, to compare our VRNN+NN model's ability in dealing with missing values with some heuristic imputation methods, we investigate four imputation methods here. "zero" is the default approach for baseline models, and "last&next", "row mean" and "NOCB" are three best known imputation methods according to the previous study³²: "last&next" is the average of the last known and next known values; "row mean" is the mean of patient's values before and after; "NOCB" is the next observation carried backward.

To evaluate if the performance in Table 1 is reliable, we apply paired t-test to check if the performance difference among different models is statistically significant. The results are shown in Table 2.

Since deep generative models can reconstruct input data, we conjecture that our VRNN+NN model has the potential to impute missing values better. To test this conjecture, we first randomly drop 10% of the values from the original data, and then use the trained VRNN+NN to impute the intentionally dropped values. The results in terms of MSE are shown in Table 3, where the MSE values of the heuristic imputation methods are also presented. The paired t-test results of these methods are shown as well.

Different imputation methods		Performance comparison	
Imputation Methods	Imputation Error	Comparison	P-values
Zero	0.909 ± 0.112	VRNN+NN vs. Zero	***
Last&next	0.434 ± 0.110	VRNN+NN vs. Last&next	*
Row mean	0.541 ± 0.114	VRNN+NN vs. Row mean	***
NOCB	0.547 ± 0.112	VRNN+NN vs. NOCB	***
VRNN+NN	0.370 ± 0.110		

Table 3. The left part is imputation error of different imputation methods. The right part is the performance comparison by paired t-test. Note that: *** ($p < 0.001$), ** ($p < 0.01$), * ($p < 0.05$).

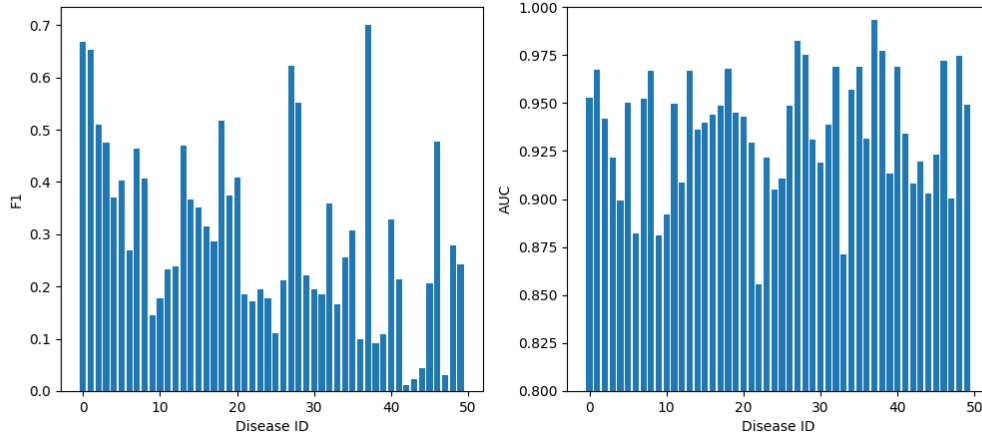


Figure 5. VRNN+NN model diagnosis performance of each diagnosis.

Discussion

From the diagnosis performances shown in the top set of Table 1, it can be observed that considering temporal dependencies, e.g., by RNN or VRNN, the diagnosis perform can be significantly improved, confirming that long-term dependency is an important property of clinical data. Secondly, involving deep generative models provides consistent performance improvement (see AE+NN vs. VAE+NN, and RNN+NN vs. VRNN+NN). This is an encouraging result and shows that the generative models (VAE and VRNN) learn better representations compared compared to the less-generative counterparts (AE and RNN).

Compared the improvement provided by the generative modeling, it can be observed that when dealing with single averaged feature vectors (VAE), the improvement is not very significant. This indicates that the data missing problem with the average vectors is not very severe, and the zero imputation is reasonably good. When dealing with temporal vector sequences (VRNN), the improvement is highly significant ($p < 0.001$). This is understandable, as the data missing problem is more serious in this case. Plus, VRNN+NN (early) model has surprisingly good performance, which is a little bit worse than VRNN+NN taking latest 100-day records into consideration. It indicates that VRNN+NN can be used in early warning of some diseases.

According to the results of the NN classifier with different features (the middle set of Table 1), we can find that the representations learned from VAE+NN performs significantly better than those learned from VAE; similarly, the representations learned from VRNN+NN are better than those learned from VRNN. This implies that the joint training that considers the classification target can lead to better feature learning. This is not surprising as the learning is now more task-oriented.

The comparison of different missing value imputation methods, as shown in the bottom set of Table 1, demonstrates that our VRNN+NN model outperforms all the heuristic imputation methods, confirming that the generative model is a more principled way for the missing data treatment. This conclusion can be drawn more explicitly from the results shown in Table 3, where the performance of different imputation methods is compared by the imputation error directly. This advantage on data imputation by itself is quite useful in practice. For example, it can help to complete the incomplete laboratory test data and give a rough range of the missing values that may assist physicians to have a better analysis.

Limitations and Future Works

Some limitations exist in current study and need to be addressed in future work. First, we only consider the most frequent 50 diagnoses, leaving the rest 2739 diseases untested. This is because there is severe data imbalance between different diseases. About 35% of 2789 diseases have only one sample. Even for the top 50 diseases selected in the study, the data distribution is still quite biased. As shown in Figure 1, the disease with ID=0 has 3566 samples, while the disease with ID=49 only has 178 samples. Unbalanced data casts a big challenge for model training, and the resultant models tend to score higher for frequent labels than infrequent ones. In this study, we didn't apply any additional preprocessing, like over-sampling and under-sampling, to address the data unbalance problem. Future work will address the data unbalance problem, by either sampling or re-weighting for infrequent classes.

Second, after looking into the VRNN+NN model performance of each disease, as shown in Figure 5, we find that the F1 scores and AUCs of different diseases vary dramatically. Some diagnoses, like "Alcohol withdrawal" (ID=37) and "Single liveborn without cesarean section" (ID=0), are quite accurate, while some others, like "Gram-neg septicemia NEC" (ID=42) and "Mal neo upper lobe lung" (ID=47), obtain F1 scores close to 0. This phenomenon implies that some diseases are difficult to detect by our model. It may result from our limitation to 50 most frequent laboratory tests, or the deficiency of other clinical data. More laboratory tests will be considered and other clinical information will be included in the future study.

Third, we only used the in-hospital records that have diagnosis labels, but there are also a lot of outpatient test results in MIMIC-III that possess no labels. In our future work, we will investigate the semi-supervised learning to enhance the capability of our model by utilizing the unlabelled data.

Conclusion

The longitudinal, incomplete, and noisy laboratory test data casts a big difficulty for automatically medical diagnosis. In this study, we proposed to utilize a deep sequential generative model in the form of VRNN to deal with the missing data problem and learn the complex temporal patterns in clinical data, and this generative model is trained jointly with the back-end discriminative model for making diagnosis decision. This leads to an end-to-end system that takes advantage of both generative learning and discriminative learning. Our experiments show that the VRNN+NN model significantly ($p < 0.001$) surpasses all the baselines and its non-temporal version, the VAE+NN model. We also find that deep generative models can help with imputing missing values in clinical laboratory tests and distilling more informative patterns for diagnosis, and the combination of generative and discriminative models leads to improvement on both feature learning and diagnosis decision. Future work involves addressing data unbalance and utilizing unlabeled data.

References

1. Singh, H., Meyer, A. N. D. & Thomas, E. J. The frequency of diagnostic errors in outpatient care: estimations from three large observational studies involving us adult populations. *BMJ Qual. & Saf.* (2014). DOI 10.1136/bmjqs-2013-002627.
2. Winters, B. *et al.* Diagnostic errors in the intensive care unit: a systematic review of autopsy studies. *BMJ Qual Saf* **21**, 894–902 (2012).
3. Schiff, G. D. *et al.* Diagnostic error in medicine: analysis of 583 physician-reported errors. *Arch. internal medicine* **169**, 1881–1887 (2009).
4. RR., G. M. N. Diagnostic error in internal medicine. *Arch Intern Med* **165**, 1493–1499 (2005).
5. Singh, H. *et al.* Types and origins of diagnostic errors in primary care settings. *Jama Intern. Medicine* **173**, 418 (2013).
6. TK, G., A, K., EJ, T. & *et al.* Missed and delayed diagnoses in the ambulatory setting: A study of closed malpractice claims. *Annals Intern. Medicine* **145**, 488–496 (2006). DOI 10.7326/0003-4819-145-7-200610030-00006.
7. Lasko, T. A., Denny, J. C. & Levy, M. A. Computational phenotype discovery using unsupervised feature learning over noisy, sparse, and irregular clinical data. *PloS one* **8**, e66341 (2013).
8. Ghassemi, M. *et al.* A multivariate timeseries modeling approach to severity of illness assessment and forecasting in icu with sparse, heterogeneous clinical data. In *AAAI*, 446–453 (2015).
9. Vincent, P., Larochelle, H., Bengio, Y. & Manzagol, P.-A. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, 1096–1103 (ACM, 2008).

10. Che, Z., Kale, D., Li, W., Bahadori, M. T. & Liu, Y. Deep computational phenotyping. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 507–516 (ACM, 2015).
11. Miotto, R., Li, L., Kidd, B. A. & Dudley, J. T. Deep patient: An unsupervised representation to predict the future of patients from the electronic health records. *Sci. reports* **6**, 26094 (2016).
12. Mikolov, T., Karafiát, M., Burget, L., Cernocký, J. & Khudanpur, S. Recurrent neural network based language model. In *Interspeech*, vol. 2, 3 (2010).
13. Graves, A., Mohamed, A.-r. & Hinton, G. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 IEEE international conference on*, 6645–6649 (IEEE, 2013).
14. Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural computation* **9**, 1735–1780 (1997).
15. Lipton, Z. C., Kale, D. C., Elkan, C. & Wetzell, R. Learning to diagnose with lstm recurrent neural networks. *arXiv preprint arXiv:1511.03677* (2015).
16. Lipton, Z. C., Kale, D. C. & Wetzell, R. C. Phenotyping of clinical time series with lstm recurrent neural networks. *arXiv preprint arXiv:1510.07641* (2015).
17. Lipton, Z. C., Kale, D. C. & Wetzell, R. Modeling missing data in clinical time series with rnns. *Mach. Learn. for Healthc.* (2016).
18. Choi, E., Bahadori, M. T., Schuetz, A., Stewart, W. F. & Sun, J. Doctor ai: Predicting clinical events via recurrent neural networks. In *Machine Learning for Healthcare Conference*, 301–318 (2016).
19. Choi, E. *et al.* Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. In *Advances in Neural Information Processing Systems*, 3504–3512 (2016).
20. Che, Z., Purushotham, S., Khemani, R. & Liu, Y. Distilling knowledge from deep networks with applications to healthcare domain. *arXiv preprint arXiv:1512.03542* (2015).
21. Che, Z., Purushotham, S., Cho, K., Sontag, D. & Liu, Y. Recurrent neural networks for multivariate time series with missing values. *arXiv preprint arXiv:1606.01865* (2016).
22. Ghahramani, Z. & Jordan, M. I. *Learning from Incomplete Data* (Massachusetts Institute of Technology, 1994).
23. Ghahramani, Z. & Jordan, M. I. Supervised learning from incomplete data via an em approach. In *Advances in neural information processing systems*, 120–127 (1994).
24. Marlin, B. M., Kale, D. C., Khemani, R. G. & Wetzell, R. C. Unsupervised pattern discovery in electronic health care data using probabilistic clustering models. In *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, 389–398 (2012).
25. Kingma, D. P. & Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).
26. Chung, J. *et al.* A recurrent latent variable model for sequential data. In *Advances in neural information processing systems*, 2980–2988 (2015).
27. Yakhnenko, O., Silvescu, A. & Honavar, V. Discriminatively trained markov model for sequence classification. In *Data Mining, Fifth IEEE International Conference on*, 8–pp (IEEE, 2005).
28. Bernardo, J. *et al.* Generative or discriminative? getting the best of both worlds. *Bayesian Stat* **8**, 3–24 (2007).
29. Yogatama, D., Dyer, C., Ling, W. & Blunsom, P. Generative and discriminative text classification with recurrent neural networks. *arXiv preprint arXiv:1703.01898* (2017).
30. Johnson, A. E. *et al.* Mimic-iii, a freely accessible critical care database. *Sci. data* **3** (2016).
31. Kingma, D. & Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
32. Engels, J. M. & Diehr, P. Imputation of missing longitudinal data: a comparison of methods. *J. clinical epidemiology* **56**, 968–976 (2003).

Appendix

ID	ICD-9	Description
0	V3000	Single liveborn, born in hospital, delivered without mention of cesarean section
1	41401	Coronary atherosclerosis of native coronary artery
2	V3001	Single liveborn, born in hospital, delivered by cesarean section
3	0389	Unspecified septicemia
4	41071	Subendocardial infarction, initial episode of care
5	4241	Aortic valve disorders
6	51881	Acute respiratory failure
7	431	Intracerebral hemorrhage
8	V3101	Twin birth, mate liveborn, born in hospital, delivered by cesarean section
9	486	Pneumonia, organism unspecified
10	5070	Pneumonitis due to inhalation of food or vomitus
11	4280	Congestive heart failure, unspecified
12	4240	Mitral valve disorders
13	430	Subarachnoid hemorrhage
14	5849	Acute kidney failure, unspecified
15	41011	Acute myocardial infarction of other anterior wall, initial episode of care
16	41041	Acute myocardial infarction of other inferior wall, initial episode of care
17	5789	Hemorrhage of gastrointestinal tract, unspecified
18	1983	Secondary malignant neoplasm of brain and spinal cord
19	5770	Acute pancreatitis
20	41519	Other pulmonary embolism and infarction
21	43411	Cerebral embolism with cerebral infarction
22	43491	Cerebral artery occlusion, unspecified with cerebral infarction
23	42731	Atrial fibrillation
24	99859	Other postoperative infection
25	03842	Septicemia due to escherichia coli [E. coli]
26	85221	Subdural hemorrhage following injury without mention of open intracranial wound, with no loss of consciousness
27	25013	Diabetes with ketoacidosis, type I [juvenile type], uncontrolled
28	4373	Cerebral aneurysm, nonruptured
29	42823	Acute on chronic systolic heart failure
30	99662	Infection and inflammatory reaction due to other vascular device, implant, and graft
31	56212	Diverticulosis of colon with hemorrhage
32	42833	Acute on chronic diastolic heart failure
33	49121	Obstructive chronic bronchitis with (acute) exacerbation
34	4271	Paroxysmal ventricular tachycardia
35	5712	Alcoholic cirrhosis of liver
36	51884	Acute and chronic respiratory failure
37	29181	Alcohol withdrawal
38	V3401	Other multiple birth (three or more), mates all liveborn, born in hospital, delivered by cesarean section
39	4321	Subdural hemorrhage
40	43310	Occlusion and stenosis of carotid artery without mention of cerebral infarction
41	99811	Hemorrhage complicating a procedure
42	03849	Other septicemia due to gram-negative organisms
43	85220	Subdural hemorrhage following injury without mention of open intracranial wound, unspecified state of consciousness
44	03811	Methicillin susceptible Staphylococcus aureus septicemia
45	042	Human immunodeficiency virus [HIV] disease
46	4414	Abdominal aneurysm without mention of rupture
47	1623	Malignant neoplasm of upper lobe, bronchus or lung
48	0380	Streptococcal septicemia
49	5715	Cirrhosis of liver without mention of alcohol

Table 4. Disease ID and ICD-9 Code Description

ID	ITEMID	LABEL	FLUID	CATEGORY	LOINC_CODE
0	51221	Hematocrit	Blood	Hematology	4544-3
1	50971	Potassium	Blood	Chemistry	2823-3
2	50983	Sodium	Blood	Chemistry	2951-2
3	50912	Creatinine	Blood	Chemistry	2160-0
4	50902	Chloride	Blood	Chemistry	2075-0
5	51006	Urea Nitrogen	Blood	Chemistry	3094-0
6	50882	Bicarbonate	Blood	Chemistry	1963-8
7	51265	Platelet Count	Blood	Hematology	777-3
8	50868	Anion Gap	Blood	Chemistry	1863-0
9	51301	White Blood Cells	Blood	Hematology	804-5
10	51222	Hemoglobin	Blood	Hematology	718-7
11	50931	Glucose	Blood	Chemistry	2345-7
12	51249	MCHC	Blood	Hematology	786-4
13	51279	Red Blood Cells	Blood	Hematology	789-8
14	51248	MCH	Blood	Hematology	785-6
15	51250	MCV	Blood	Hematology	787-2
16	51277	RDW	Blood	Hematology	788-0
17	50960	Magnesium	Blood	Chemistry	2601-3
18	50893	Calcium, Total	Blood	Chemistry	2000-8
19	50970	Phosphate	Blood	Chemistry	2777-1
20	50820	pH	Blood	Blood Gas	11558-4
21	50802	Base Excess	Blood	Blood Gas	11555-0
22	50804	Calculated Total CO2	Blood	Blood Gas	34728-6
23	50821	pO2	Blood	Blood Gas	11556-8
24	50818	pCO2	Blood	Blood Gas	11557-6
25	51275	PTT	Blood	Hematology	3173-2
26	51237	INR(PT)	Blood	Hematology	5895-7
27	51274	PT	Blood	Hematology	5902-2
28	50808	Free Calcium	Blood	Blood Gas	1994-3
29	50885	Bilirubin, Total	Blood	Chemistry	1975-2
30	50861	Alanine Aminotransferase (ALT)	Blood	Chemistry	1742-6
31	50878	Asparate Aminotransferase (AST)	Blood	Chemistry	1920-8
32	50863	Alkaline Phosphatase	Blood	Chemistry	6768-6
33	50809	Glucose	Blood	Blood Gas	2339-0
34	50822	Potassium, Whole Blood	Blood	Blood Gas	6298-4
35	50813	Lactate	Blood	Blood Gas	32693-4
36	50817	Oxygen Saturation	Blood	Blood Gas	20564-1
37	51244	Lymphocytes	Blood	Hematology	731-0
38	51256	Neutrophils	Blood	Hematology	761-7
39	51254	Monocytes	Blood	Hematology	742-7
40	51200	Eosinophils	Blood	Hematology	711-2
41	51146	Basophils	Blood	Hematology	704-7
42	50812	Intubated	Blood	Blood Gas	
43	50862	Albumin	Blood	Chemistry	1751-7
44	50910	Creatine Kinase (CK)	Blood	Chemistry	2157-6
45	50825	Temperature	Blood	Blood Gas	
46	50816	Oxygen	Blood	Blood Gas	19994-3
47	51491	pH	Urine	Hematology	5803-2
48	51498	Specific Gravity	Urine	Hematology	5811-5
49	50911	Creatine Kinase, MB Isoenzyme	Blood	Chemistry	6773-6

Table 5. Labitem ID and Description