Optimizing Autoencoders for Learning Deep Representations From Health Data

Chongyu Zhou, *Member, IEEE*, Yao Jia, and Mehul Motani, *Fellow, IEEE*

***Abstract*—Analyzing patients' health data using machine learning techniques can improve both patient outcomes and hospital operations. However, heterogeneous patient data (e.g., vital signs) and inefficient feature learning methods affect the implementation of machine learning-based patient data analysis. In this paper, we present a novel unsupervised deep learning-based feature learning (DFL) framework to automatically learn compact representations from patient health data for efficient clinical decision making. Real-world pneumonia patient data from the National University Hospital in Singapore are collected and analyzed to evaluate the performance of DFL. Furthermore, publicly available electroencephalogram data are extracted from the UCI Machine Learning Repository to test and support our findings. Using both data sets, we compare the performance of DFL to that of several popular feature learning methods and demonstrate its advantages.**

***Index Terms*—Machine learning, feature learning, deep learning, unsupervised learning, classification, patient health data.**

I. Introduction

PATIENTS' health records, including conventional paper-based ones and modern electronic health records (EHRs), are built by healthcare institutions for clinical and billing purposes, which normally contain a large amount of data from individual patients, ranging from vital sign measurements to demographic data such as age, gender and habits, etc.. In order to precisely characterize each individual patient and better inform clinical decision making, advanced machine learning techniques provide opportunities to implement deep analysis in patient health data [1]. Being able to characterize the status of each patient helps patients get the most effective treatment and reduces the risk of adverse complications caused by prolonged hospital stays [2]–[4]. Furthermore, with better characterization of individual patients, hospitals can come up with better plans to allocate their resources such as medical equipment, clinical staff and beds [5]. With these reasons, developing efficient machine learning algorithms to analyze patient data becomes one of the key focuses in modern medicine research [6].

Manuscript received February 6, 2018; revised May 31, 2018; accepted July 7, 2018. Date of publication July 16, 2018; date of current version January 2, 2019. This work was supported in part by the Ministry of Education, Singapore, under Grant R-263-000-C81-114. *(Corresponding author: Mehul Motani.)*

The authors are with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117583 (e-mail: chongyuzhou@u.nus.edu; eleyaj@nus.edu.sg; motani@nus.edu.sg).

Digital Object Identifier 10.1109/JBHI.2018.2856820

The features extracted from health data directly determine the performance of machine learning algorithms [7]. This is because different representations can entangle and hide more or less the different explanatory factors of variation behind the data. Various feature selection algorithms have been proposed in the literature to extract efficient features from patient data. The most common type of feature selection is supervised, in which features are selected based on the knowledge of domain experts [6]. For example, a hidden Markov model (HMM) is built in [8] to discover temporal signatures from an asthma patient data set. An event detection framework is developed in [9] to mine event patterns from patient data based on nonnegative matrix factorization (NMF). Similarly, a compressive sensing (CS) based algorithm is developed in [10] to detect abnormal events in ECG measurements. The techniques used in [8]–[10] require prior knowledge on the data structure, which may not be available all the time. Therefore, these kinds of supervised feature selection methods, though appropriate in some specific situations, do not generalize well. Furthermore, as such supervised feature extraction techniques solely depend on domain knowledge, they may miss some novel features hidden inside the patient data.

To address the shortcomings of supervised feature extraction techniques, data-driven feature extraction approaches have been proposed [11]–[13]. These data-driven methods construct data representations as matrices consisting of all of the available data descriptors in the patient data. As a result, such representations are usually sparse, noisy and repetitive, making it difficult to make efficient clinical decisions.

Hence, unsupervised feature selection methods which attempt to automatically learn signatures and dependencies in the data have been recently studied. In particular, autoencoder-based feature extraction techniques, due to their ability to explicitly learn compact representations, are a type of commonly used unsupervised feature learning method [7]. By minimizing the reconstruction error through the encoder-decoder path, autoencoders automatically learn the model parameters and come up with compact data representations for efficient decision making in an unsupervised manner. Despite these advantages, autoencoder-based feature learning methods have not been broadly applied to analyze patients' health data due to the following unique challenges.

First, unlike images or videos, patient health record data tend to be highly heterogeneous [14]. Clinicians may monitor patients at different schedules and frequencies based on patients health status. For example, when a patient's health condition

starts deteriorating, clinicians may monitor the patient more frequently. As a result, the frequency of vital sign measurements and the clinical tests conducted will vary among different patients [6], [15]. Such heterogeneity prevents direct application of autoencoder-based feature learning methods.

Secondly, modern patient data, such as lab test results and vital sign measurements, are stored as high-dimensional multivariate time series [16], which contains both temporal and spatial dependencies. Therefore, learning efficient features requires extracting useful information from both temporal and spatial domains, which largely depends on the structure of the autoencoder used. Recent works that used autoencoders to extract features from patient data [17]–[20] proposed autoencoder structures only in an ad-hoc manner, which may not be effective to extract useful features. The problem of optimizing the structure of the autoencoders for learning effective features from patient data is largely ignored in the literature.

In this paper, we present a novel scheme to learn an efficient and compact representations from patient health data for decision making and diagnosis. The scheme, called DFL, adopts a deep learning approach to construct a feature learning block, built upon the idea of stacked denoising autoencoders, that is able to learn features from patient health data in an unsupervised way for more efficient classification or prediction. Furthermore, we propose a novel mutual information-based structure optimization algorithm to optimize the structure of the autoencoder used in the DFL framework to further improve the classification performance.

To evaluate the effectiveness of the proposed DFL framework, we apply it in two different classification problems using two real-world patient data sets. First, we have collected real-world pneumonia patient data from the National University Hospital (NUH) in Singapore, where DFL is applied to extract features in a problem of predicting the patients' length of stay in the hospital. Secondly, a publicly available electroencephalogram (EEG) data set is extracted from the UCI Machine Learning Repository. DFL is applied to extract features from the EEG data set to determine whether a patient is under alcoholic control or not. Through experiments on both data sets, we compare DFL with various existing feature selection methods proposed in the literature and illustrate the advantages of DFL through classification performance.

The rest of this paper is organized as follows. Section II presents some background information on autoencoders. In Section III, we introduce the system model of the proposed DFL framework. Section IV describes the data sets used and the classification problems studied. In Sections V and VI, we carry out experiments using two different data sets and evaluate the performance of the proposed DFL framework. Finally, Section VII concludes the paper. A preliminary version of this work has been presented in [20].

II. Background

In this section, we present some background information about autoencoders. In particular, we first introduce the structure of denoising autoencoders (DAEs). Then, we continue to discuss how to stack a set of DAEs to form a stacked denoising autoencoder (SDAE).

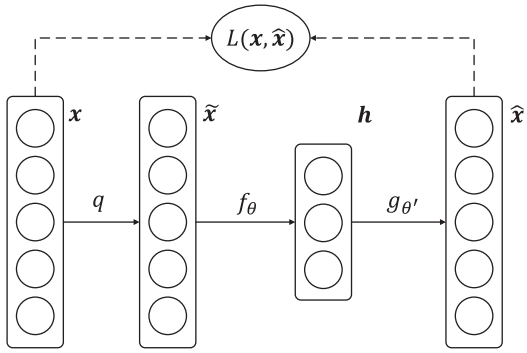

Fig. 1. Training a denoising autoencoder.

A. Denoising Autoencoders

In this paper, we use DAEs [21] as basic building blocks of the proposed DFL framework. Fig. 1 presents the structure of a DAE. In order to learn features that are robust to input corruption, the DAE tries to reconstruct the input data $\mathbf{x} \in \mathbb{R}^D$ from an encoded representation $\mathbf{h} = f_\theta(\tilde{\mathbf{x}}) \in \mathbb{R}^N$ of a corrupted version of the input $\tilde{\mathbf{x}} \in \mathbb{R}^D$ via a decoding function $g_{\theta'}(\mathbf{h})$. The parameters of the encoder and the decoder functions in the DAE are trained by minimizing the reconstruction error. Specifically, four components exist in each DAE:

- A probabilistic distribution $q(\tilde{\mathbf{x}}|\mathbf{x})$ that adds stochastic noise (perturbations) into the input data $\mathbf{x}$ to generate a corrupted version $\tilde{\mathbf{x}}$;
- An encoding function $f_\theta(\tilde{\mathbf{x}}) : \tilde{\mathbf{x}} \in \mathbb{R}^D \mapsto \mathbf{h} \in \mathbb{R}^N$ that generates a hidden representation of the input data;
- A decoding function $g_{\theta'}(\mathbf{h}) : \mathbf{h} \in \mathbb{R}^N \mapsto \hat{\mathbf{x}} \in \mathbb{R}^D$ that recovers the input data from the hidden representation $\mathbf{h}$;
- A cost function $L(\mathbf{x}, \hat{\mathbf{x}})$ measuring the dissimilarity between the original input and the reconstructed data.

Similar to [21], we consider the conditional distribution $q(\tilde{\mathbf{x}}|\mathbf{x})$ to independently perturb each dimension of the input data, i.e., $q(\tilde{\mathbf{x}}|\mathbf{x}) = \prod_{i=1}^{D} q(\tilde{x}_i|x_i)$. By adding the perturbations, the representation $\mathbf{h}$ is generated from a corrupted input $\tilde{\mathbf{x}}$. Therefore, it forces the DAE to learn a cleverer mapping than the identity, which extracts useful features for denoising and benefits the classification problem later on.

After adding noise to the input signal $\mathbf{x} \in \mathbb{R}^D$, the perturbed signal $\tilde{\mathbf{x}} \in \mathbb{R}^D$ is transformed to a hidden representation $\mathbf{h} \in \mathbb{R}^N$ as follows:

$$\mathbf{h} = f_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}) = s(\mathbf{W}\tilde{\mathbf{x}} + \mathbf{b}), \tag{1}$$

where $\boldsymbol{\theta} = (\mathbf{W}, \mathbf{b})$, $\mathbf{W} \in \mathbb{R}^{N \times D}$ is a weight coefficient matrix, $\mathbf{b} \in \mathbb{R}^N$ is a hidden bias vector, and $s(\cdot)$ is a non-linear activation function. Then, the hidden representation $\mathbf{h}$ is transformed back to a reconstructed signal $\hat{\mathbf{x}} \in \mathbb{R}^D$ via

$$\hat{\mathbf{x}} = g_{\boldsymbol{\theta}'}(\mathbf{h}) = s'(\mathbf{W}^T\mathbf{h} + \mathbf{c}), \tag{2}$$

where $\boldsymbol{\theta}' = (\mathbf{W}^T, \mathbf{c})$, $\mathbf{c} \in \mathbb{R}^D$ is an input bias vector and $s'(\cdot)$ is a non-linear map at the decoder, which may not be the same as the one at the encoder. The parameters $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ are adapted to minimize the reconstruction error, measured by a loss metric $L(\mathbf{x}, \hat{\mathbf{x}})$, over a given training set. In our work, we use *mean*

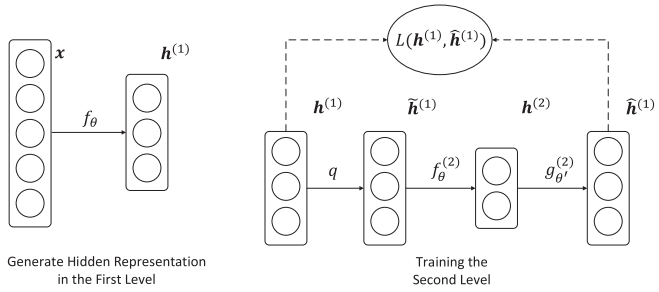

Fig. 2. Training process in SDAE.

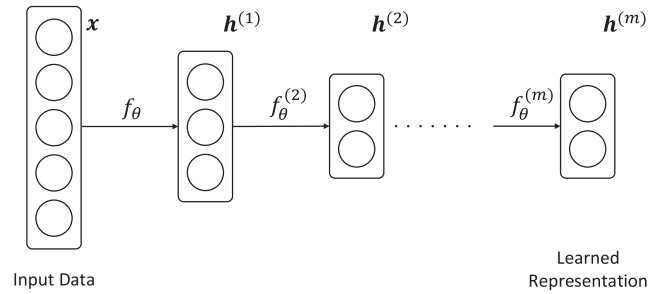

Fig. 3. DFL feature learning using stacked denoising autoencoders.

squared error as the loss metric to train the parameters of each DAE.

B. Stacked Denoising Autoencoders

DAEs are usually stacked together to construct a deep network, referred to as SDAE, to extract deeper features from the input data. The DAEs at different levels of the SDAE are trained independently level by level, i.e., after training the DAE at level i, the hidden representation $\mathbf{h}^{(i)}$ will be used as the input to train the DAE at level $i+1$.

We note that corruption of input is applied to the denoising-training of each level at the initial training stage only, so as to learn meaningful feature extractors. Once the mapping f_θ has been trained, the uncorrupted inputs will then be used to generate the hidden representation. Fig. 2 shows the complete procedure for training and stacking several levels of DAEs. After training the first level DAE (see Fig. 1), the learned encoding function f_θ is applied to the clean input $\mathbf{x}$ to generate the hidden representation of the first level DAE $\mathbf{h}^{(1)}$ (see the left graph in Fig. 2). The hidden representation $\mathbf{h}^{(1)}$ is used as input to train the second level DAE (see the right graph in Fig. 2) to learn the encoding function of the second level $f_\theta^{(2)}$. The training procedure repeats for the following levels. After training each level, the learned encoding functions are stacked to generate a deep representation from the input data for the classification task, as shown in Fig. 3.

We note that the architecture of the SDAE affects the quality of the learned features. As the architecture of the SDAE depends on the structures of the DAEs at different levels, i.e., the number of nodes $N^{(i)}$ in the hidden layer of each DAE i, finding the optimal number $N^{(i)}$ of each DAE i is important for extracting good data representations. Cross-validation is commonly used for model selection in machine learning tasks. However, in a stacked architecture like SDAE, selecting the optimal model structure becomes a combinatorial problem and cross-validation becomes computationally prohibitive. To address these issues, in Section III, we propose a novel algorithm to optimize the structure of SDAE for feature learning.

III. Methodology

In this section, we present the system model of the proposed DFL framework (see Fig. 4). There are in total three data processing blocks in the DFL framework. The pre-processing block cleans and interpolates the raw measurements in order to regularize the patient health data[1]. The feature learning block determines efficient patient representations (features). The classification block generates the decisions based on the computed patient representations.

A. Pre-Processing

Heterogeneity of the recorded medical observations creates challenges for patient health data analysis. Since different patients have their own symptoms and health status, clinicians may decide different monitoring plans (i.e., monitoring periods and measuring frequencies) for each patient. Furthermore, even for the same patient, the vital signs monitoring plan can evolve with the patient condition. This irregularity in the patient health data leads to a lack of structure that complicates feature learning and classification.

In our work, we address these problems by interpolating the irregular patient health data, i.e., so that all vital signs are measured at the same time and frequency. We mark points when vital signs are not taken as missing and interpolate using spline interpolation and Gaussian process regression.

1) Spline Interpolation: Spline interpolation is a well-known signal processing technique for data interpolation. By using piecewise polynomials called splines, spline interpolation fills in the "missing" measurements and regularizes the patient health data used for feature learning. We refer the reader to [22] for more information.

2) Gaussian Process Regression: Gaussian process regression (GPR) [23] is another well-known method for data interpolation, in which Gaussian processes are used to model the spatial correlation between the different types of health data, as well as the temporal correlation between adjacent measurements of each type of health data. Based on the learned Gaussian process model, we can estimate the "missing" measurements. The detailed formulation of the Gaussian process regression can be found in [20].

We are aware of the rich literature on data imputation that focuses on recovering the missing measurements in the patient data, such as [24], [25]. We note that those researches are orthogonal to our study in this paper. With our proposed DFL

[1]Pre-processing on the raw data may be optional. For the experiments in Section VI, as the EEG data set is already cleaned and structured, we do not do any pre-processing on the raw data.

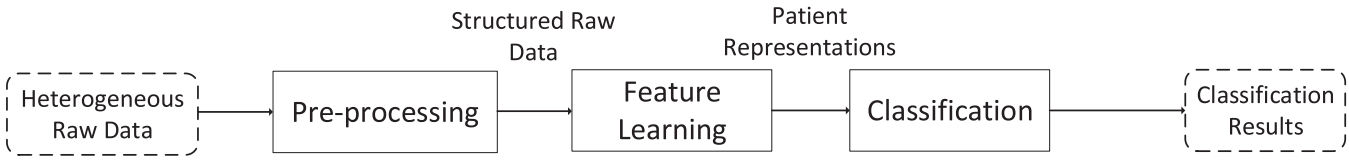

Fig. 4. Overview of the DFL framework including the various data processing blocks.

framework, we focus on feature learning rather than data imputation. Therefore, in this paper, we only use the above two baseline pre-processing techniques for simplicity. The data imputation techniques proposed in the literature can be applied in the data pre-processing block of DFL to potentially further boost the performance of DFL.

Patient data are usually multi-modal where different data types are in different ranges. For example, the SpO2 measurements are in a different range from body temperature measurements. Therefore, to reduce the weighting effect in different modalities, we normalize the patient data in each modality into the same range between 0 and 1 after interpolation.

B. Feature Learning

We next present the feature learning block. The feature learning block is based on the SDAE introduced in Section II. The perturbation process used in each DAE is set to be a simple additive isotropic Gaussian noise, i.e., $q(\tilde{\mathbf{x}}|\mathbf{x}) = \mathcal{N}(\mathbf{x}, \boldsymbol{\sigma}^2 I)$, where $\boldsymbol{\sigma}$ is a vector of noise standard deviation for each dimension of the input data. In this paper, we use rectified-linear functions to do the non-linear transform for both encoder and decoder in each DAE at each level of SDAE.

The main problem in constructing the feature learning block is to determine an optimal SDAE architecture. Suppose we would like to add m levels to hidden layers in the SDAE. Let U be a set of hidden layers that we could choose to add into the SDAE. The conventional way of determining the optimal architecture of a SDAE is through cross-validation: dividing total data set into a training set, a test set and a validation set. The optimal model structure is chosen as the one giving the best classification performance on the validation set

$$\begin{aligned} \min_{\mathcal{T}} \quad & F(\mathcal{T}, \boldsymbol{\theta}; \mathbf{x}_{val}, \mathbf{y}_{val}) \\ \text{s.t.} \quad & \boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}} L(\mathcal{T}, \boldsymbol{\theta}; \mathbf{x}_{train}) \\ & \mathcal{T} = \{T_1, T_2, ..., T_m\} : T_i \in U, \forall i = 1, ..., m \end{aligned} \quad (3)$$

where $\mathcal{T}$ is the structure of the whole SDAE; T_i is the structure, i.e., the number of hidden units, at i-th hidden layer; $\mathbf{x}_{val}$ and $\mathbf{y}_{val}$ are the data and labels in the validation set and $\mathbf{x}_{train}$ are the data in the training set. We note that it is possible that different hidden layers have the same number of hidden units. In (3), $F(\mathcal{T}, \boldsymbol{\theta}; \mathbf{x}_{val}, \mathbf{y}_{val})$ is a function evaluating the classification cost (e.g., error rate) on the validation set based on the selected model structure $\mathcal{T}$ and the resulting model parameters $\boldsymbol{\theta}$. $L(\mathcal{T}, \boldsymbol{\theta}; \mathbf{x}_{train})$ is a loss function to train to parameters based on the training data. Direct optimizing (3) is usually computationally prohibitive due to the following reasons. First, evaluating $F(\mathcal{T}, \boldsymbol{\theta}; \mathbf{x}_{val}, \mathbf{y}_{val})$ usually requires training a classifier (e.g., a support vector machine) to obtain the classification results, which can be computationally intensive. Secondly, selecting the model structure is a combinatorial problem whose complexity is $O(|U|^m)$, which is exponential to the number of layers m. Due to these reasons, directly solving (3) is usually intractable. To overcome these challenges, in the following, we propose an efficient mutual information-based algorithm to search for the optimal architecture for the SDAE.

Let $\mathbf{x}$ be the original data and $\mathbf{y}$ be the labels. It has been shown in [26] that the generalization error of a deep learning framework depends on the mutual information between the hidden representation in the last hidden layer $\mathbf{h}^{(m)}$ and the labels $\mathbf{y}$, i.e., $I(\mathbf{h}^{(m)}; \mathbf{y})$. Such finding is intuitive to understand. As the hidden representation at the last hidden layer $\mathbf{h}^{(m)}$ is the feature set passed into the classifiers, a higher mutual information between $\mathbf{h}^{(m)}$ and $\mathbf{y}$ means that $\mathbf{h}^{(m)}$ contains more relevant information about $\mathbf{y}$, which makes it easier for the classifiers to make classification decisions. Therefore, we set the objective of our structure optimization algorithm to be maximizing the mutual information between $\mathbf{h}^{(m)}$ and $\mathbf{y}$

$$\begin{aligned} \max_{\mathcal{T}} \quad & I(\mathbf{h}^{(m)}; \mathbf{y}_{train} | \mathcal{T}, \boldsymbol{\theta}, \mathbf{x}_{train}, \mathbf{y}_{train}) \\ \text{s.t.} \quad & \boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}} L(\mathcal{T}, \boldsymbol{\theta}; \mathbf{x}_{train}) \\ & \mathbf{h}^{(m)} = \mathcal{T}(\boldsymbol{\theta}, \mathbf{x}_{train}) \\ & \mathcal{T} = \{T_1, T_2, ..., T_m\} : T_i \in U, \forall i = 1, ..., m. \end{aligned} \quad (4)$$

We note that, in our scheme, we do not need a separate validation set. The whole data set can only be split into a training and a test set. Training the model parameters and determining the model structure can be done together based on the training set only. The reason is that, in (4), the objective function can be evaluated in closed form without training any classifier, which greatly reduces the computational complexity.

However, solving (4) is still NP-hard due to the combinatorial nature of the problem. Therefore, in order to solve the problem efficiently, we propose the following heuristic called OSDAE for selecting the structure of each layer in the SDAE:

$$T_i = \arg\max_{T_i \in U} I(\mathbf{h}^{(i)}; \mathbf{y}_{train} | \boldsymbol{\theta}^{(i)}, \mathbf{h}^{(i-1)}), \quad (5a)$$

$$\text{s.t.} \quad \boldsymbol{\theta}^{(i)} = \arg\min_{\boldsymbol{\theta}} L(T_i, \boldsymbol{\theta}^{(i)}; \mathbf{h}^{(i-1)}), \forall i = 1, ..., m \quad (5b)$$

where $\mathbf{h}^{(i-1)}$ is the hidden representation from the previous hidden layer ($\mathbf{h}^{(0)} = \mathbf{x}_{train}$); and $\mathbf{h}^{(i)}$ is the hidden representation from the current hidden layer, which is obtained by training the encoder of the DAE at the current level of the SDAE using $\mathbf{h}^{(i-1)}$ as the input. The implementation details of the proposed OSDAE scheme is illustrated in Algorithm 1. We note that the

Algorithm 1: OSDAE Scheme.

Input: U, m, $\mathbf{x}_{train}$, $\mathbf{y}_{train}$
Output: Structure and parameter of the constructed SDAE $\mathcal{T} = \{T_1, T_2, ..., T_m\}$, $\boldsymbol{\theta} = \{\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, ..., \boldsymbol{\theta}^{(m)}\}$

Initialize $\mathbf{h}^{(0)} = \mathbf{x}_{train}$
for i from 1 to m **do**
  Initialize $maxMI = 0$; $T_i = \emptyset$; $\boldsymbol{\theta}^{(i)} = \emptyset$; $\mathbf{h}^{(i)} = \emptyset$
  for j from 1 to $|U|$ **do**
    Select a layer structure $T_{temp} = U(j)$
    Train parameters $\boldsymbol{\theta}^{temp} = \arg\min_{\boldsymbol{\theta}} L(T_{temp}, \boldsymbol{\theta}^{temp}; \mathbf{h}^{(i-1)})$
    Compute hidden representation $\mathbf{h}^{temp} = T_{temp}(\boldsymbol{\theta}^{temp}, \mathbf{h}^{(i-1)})$
    Compute mutual information with the label $MI^{temp} = I(\mathbf{h}^{temp}; \mathbf{y}_{train})$
    if $MI^{temp} > maxMI$ **then**
      $maxMI = MI^{temp}$
      $T_i = T_{temp}$
      $\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}^{temp}$
      $\mathbf{h}^{(i)} = \mathbf{h}^{temp}$
    end
  end
end
Return $\mathcal{T} = \{T_1, T_2, ..., T_m\}$, $\boldsymbol{\theta} = \{\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, ..., \boldsymbol{\theta}^{(m)}\}$

computational complexity of the solution method in Algorithm 1 is $O(m|U|)$, where $|U|$ is the cardinality of the available hidden layer set U.

C. Classification

We now discuss the classifiers used in the classification block. Two types of classifiers are considered, i.e., support vector machines (SVMs) and artificial neural networks (ANNs).

1) Support Vector Machines: Support vector machines (SVMs) are a class of supervised machine learning algorithms for classification. In this paper, we use SVMs with polynomial kernel functions [27].

2) Artificial Neural Networks: For comparison purposes, we also construct a neural network classifier, referred to as ANN. Based on preliminary experiments aimed at finding a good structure for the ANN, we used a network with one input and output layer, and three hidden layers with 50 neurons at each of the hidden layers. The activation functions used in the hidden layers are rectified linear units (ReLUs) and the activation function used in the output layer is the Sigmoid.

IV. Data Sets and Classification Problem

In this section, we introduce the data sets used and the classification problems investigated in our study. Two different data sets are used to evaluate the proposed DFL framework. Based on the context of the data sets, two different classification problems are considered.

A. Data Set 1: Hospital Patients With Pneumonia

1) Data Set Description: The first data set contains vital-sign time series data of patients with community acquired pneumonia. To conduct this research, we worked with the National University Hospital (NUH) in Singapore. The pneumonia patient data are recorded as physical medical dockets, which is collected from the Medical Records Office of NUH. A total of 111 patients' physical medical dockets were collected from January 2016 to March 2016. The patient data were manually digitalized from the physical medical dockets for the analysis in this paper.

The recorded clinical data of each patient contain six types of time series measurements, including *SpO2*, *respiration*, *pulse rate*, *systolic blood pressure*, *diastolic blood pressure* and *body temperature*. Furthermore, a set of static patient data are recorded in the data set as well, including *required oxygenation level*, *intravenous infusion*, *monitoring frequency*, *gender* and *age*. We consider each health record as a patient. In total, there are 126 health records (patients) in this study.

The data collection for this study was approved by the local institutional ethics committee (DSRB Reference: 2016/00234).

2) Classification Problem: In this paper, we are interested in predicting the patients' length of stay in the hospital. Specifically, we would like to predict whether or not a patient will stay in the hospital for more than a week by using her health data measured in the first 24 hours from the point of time when she is admitted to the hospital. The problem is modeled as a binary classification problem: Class 0 includes patients who stay in the hospital for less than or equal to 7 days, and Class 1 includes patients who stay in the hospital for more than 7 days. In the pneumonia data set, Class 0 contains 80 patients and Class 1 contains 46 patients.

B. Data Set 2: Electroencephalogram (EEG) Data

1) Data Set Description: The second data set used is a EEG data set extracted from the UCI Machine Learning Repository [28]. It is collected to study the correlation between EEG measurements and genetic predisposition to alcoholism.

Only a subset of the whole data set are used for our study, which contains the EEG time series data from 600 different patients. Each patient's EEG time series data contains a 64-channel measurement for 256 time periods. Each patient is labeled as either alcoholic (Class 0) or non-alcoholic (Class 1). In the EEG data set used in our study, both Class 0 and Class 1 contain 300 patients equally.

2) Classification Problem: The classification problem considered in this study is to classify whether a patient is alcoholic (Class 0) or non-alcoholic (Class 1), based on her EEG measurements.

V. Results Using Data Set 1

We now evaluate the proposed DFL framework using Data Set 1, which contains the vital signs of pneumonia hospital patients. We first discuss some existing feature selection methods that are

used to compare with DFL and then present the classification performance comparison results.

A. Feature Selection Methods

1) Manual Feature Selection (MFS): MFS manually extracts features from the measured patient vital signs. The measured patient data consist of time-series vital sign data and demographic data, e.g., age and gender.

For the time series data, MFS extracts features in the time and frequency domains. We include frequency domain characteristics because certain patterns such as periodicity are captured more easily by frequency domain transforms, i.e., Discrete Wavelet Transform (DWT) and Discrete Fourier Transform (DFT). The time domain features extracted include the following statistics: (i) mean (Mean), (ii) maximum value (Max), (iii) minimum value (Min) and (iv) standard deviation (StdDev), of each vital sign. The frequency domain features extracted include (i) DWT maximum approximation coefficient (MAC-DWT), (ii) DWT maximum detailed coefficient (MDC-DWT), and (iii) DFT largest frequency component (LFC-DFT). In total, seven features for each vital sign measurement are extracted.

In order to capture the fluctuation and trending in the vital signs. We compute the delta change of each original vital sign between every two adjacent measurements. The delta change of each vital sign is also time series and can be understand as the first derivatives of the original vital sign measurements. Similarly, the 7 features in both time and frequency domains are extracted from the delta change of each type of vital sign measurements.

In MFS, we also use demographic data and static health data as relevant features. The static health data include the total amount of intravenous infusion (an integer value), nurse monitoring frequency (an integer value) and whether oxygenation was required in the first 24 hours (a binary value). The demographic data include gender and age.

Overall, 14 features are extracted for each type of vital sign measurements, i.e. 7 features for the original measurements and 7 features for the delta change. As introduced in Section IV-A, there are 6 types of vital sign measurements recorded in the pneumonia data set. Therefore, there are 84 features extracted from all time series data in total. Adding in the 5 features extracted from the static patient data, in the end, MFS extracts 89 features from the pneumonia data set.

2) Raw Feature Selection (RFS): In RFS, we want to use the raw measured data directly as the feature set. For Data Set 1, the raw measurements are irregular and thus, we need to interpolate the measured values in order to yield a regular structured data set. As such, we interpolate the measured data to a resolution of one minute using the various algorithms described in Section III-A and use the first twenty-four hours of measured data as the feature set. Therefore, RFS extracts 1440 features for each vital sign. As six vital signs are measured, RFS extracts a feature vector of size 8640 for each patient from Data Set 1.

3) Feature Extraction by PCA: Principal component analysis (PCA) [29] is a popular method for extracting features from high dimensional data. In particular, as a statistical procedure, PCA converts the realization of some probably correlated variables into principal components which are linearly uncorrelated to one another using orthogonal transformation. The amount of principal components is usually less than or equal to the amount of original variables. In this paper, we apply PCA to the interpolated data to generate features set with 100 principal components as a benchmark feature set to compare with DFL.

4) Feature Set Computed From DFL: DFL aims to extract features based on a constructed SDAE as shown in Fig. 3. The feature set obtained depends on the number of layers in the SDAE and the number of hidden units in the DAE at each layer. As the hidden layers in the SDAE can have arbitrary sizes, in this paper, we define a finite set U to indicate the possible numbers of hidden units that can be added into a hidden layer of the SDAE. For the pneumonia data set, we define $U = \{8640, 8192, 4096, 2048, 1024, 512, 256, 128\}$. After conducing some preliminary experiments, we find the SDAE with 3 hidden layers can give relatively good results. Therefore, we set the number of the hidden layers in the SDAE to be $m = 3$. The standard deviation of the noise level $\boldsymbol{\sigma}$ for training the SDAE is set to be 0.1 in this experiment.

Motivated by the settings in [18], [21], a simple way to compose a SDAE is to let different hidden layers have the same size. Therefore, we first construct a *uniform SDAE* (USDAE) for extracting features, which have the same number of hidden units in each hidden layer. Based on the defined U, by solving the optimization problem in (4) with an additional constraint that all hidden layers have the same size, we find out that the USDAE with 512 hidden nodes in each hidden layer can generate the best feature set. Therefore, the resulting feature set from USDAE has a size of 512 for each patient.

Finally, we try to optimize the structure of the constructed SDAE in order to further improve the generated features. By solving the optimization problem in (5a)-(5b), we construct an optimized stacked denoising autoencoder, denoted as OSDAE. For the pneumonia data set, based on the defined U, the obtained OSDAE have 3 layers with 2048, 512 and 256 hidden units in each of its hidden layers, respectively. By passing the interpolated raw data into the constructed OSDAE, the resulting feature set has a size of 256 for each patient.

B. Classification Results Using Data Set 1

In Tables I and II, we compare the performance of classification algorithms using the feature sets determined by the various feature selection methods described in Section V-A. Fig. 5 shows the work flow of the various feature selection systems tested. The raw data are passed through an interpolation block (if necessary), a feature learning block and a classification block. There are different sub-blocks inside each block indicating the different interpolation, feature learning and classification methods considered in this paper. In the experiment results shown later, we name each experiment case based on the sub-blocks it passes through. For example, D1-S-OSDAE-SVM indicates the experiment case of passing the raw pneumonia through spline interpolation, OSDAE for feature learning and SVM for classification.

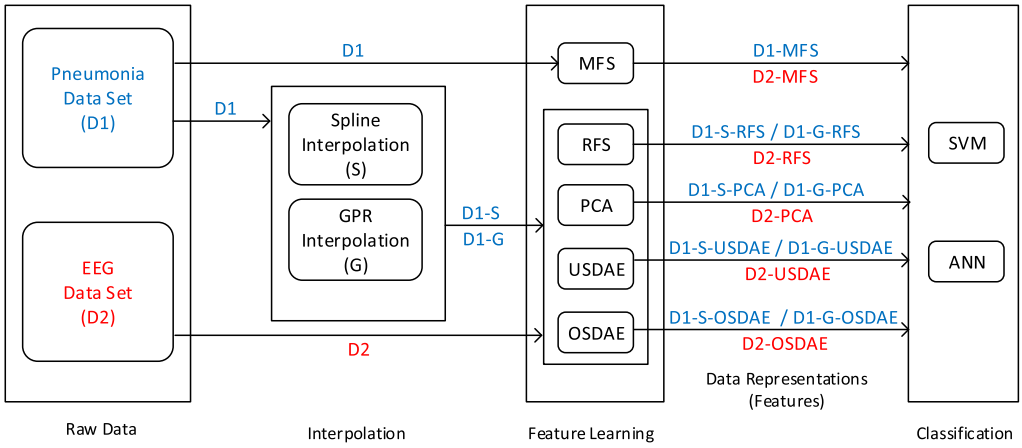

Fig. 5. Workflow of the various feature selection experiments. MFS/RFS stands for manual/raw feature selection. PCA stands for principal component analysis. USDAE/OSDAE stands for uniform/optimized stacked denoising autoencoder. SVM and ANN are the support vector machine and artificial neural network classifiers, respectively.

TABLE I
SVM CLASSIFICATION RESULTS USING PNEUMONIA DATA

Feature Set	Accuracy	AUC-ROC	F1-score
D1-MFS-SVM	0.534 (±0.057)	0.617 (±0.013)	0.535 (±0.025)
D1-S-RFS-SVM	0.531 (±0.074)	0.635 (±0.017)	0.514 (±0.033)
D1-S-PCA-SVM	0.563 (±0.006)	0.571 (±0.007)	0.520 (±0.009)
D1-S-USDAE-SVM	0.635 (±0.006)	0.711 (±0.006)	0.621 (±0.002)
D1-S-OSDAE-SVM	**0.651** (±0.007)	**0.723** (±0.006)	**0.623** (±0.003)
D1-G-RFS-SVM	0.681 (±0.089)	0.796 (±0.012)	0.681 (±0.022)
D1-G-PCA-SVM	0.643 (±0.005)	0.712 (±0.003)	0.632 (±0.002)
D1-G-USDAE-SVM	0.698 (±0.007)	0.742 (±0.008)	0.683 (±0.010)
D1-G-OSDAE-SVM	**0.714** (±0.002)	**0.810** (±0.001)	**0.702** (±0.002)

TABLE II
ANN CLASSIFICATION RESULTS USING PNEUMONIA DATA

Feature Set	Accuracy	AUC-ROC	F1-score
D1-MFS-ANN	0.619 (±0.079)	0.685 (±0.011)	0.574 (±0.062)
D1-S-RFS-ANN	0.688 (±0.009)	0.715 (±0.054)	0.675 (±0.055)
D1-S-PCA-ANN	0.667 (±0.007)	0.671 (±0.003)	0.652 (±0.002)
D1-S-USDAE-ANN	0.677 (±0.007)	0.701 (±0.008)	0.661 (±0.003)
D1-S-OSDAE-ANN	**0.703** (±0.006)	**0.736** (±0.006)	**0.681** (±0.002)
D1-G-RFS-ANN	0.718 (±0.009)	0.795 (±0.012)	0.709 (±0.068)
D1-G-PCA-ANN	0.683 (±0.008)	0.704 (±0.008)	0.675 (±0.010)
D1-G-USDAE-ANN	0.695 (±0.007)	0.712 (±0.006)	0.683 (±0.002)
D1-G-OSDAE-ANN	**0.816** (±0.007)	**0.856** (±0.006)	**0.801** (±0.001)

As introduced in Section IV-A, the data samples from two classes are imbalanced. Class 0 has 80 patients and Class 1 has 46 patients. In order to construct a balanced training set and test set to prevent over-fitting, we choose the training and test sets in the following way. During each experiment run, we randomly select 30 patients from both Class 0 and Class 1 respectively to construct the training set. Then, the test set is constructed by randomly selecting 16 patients from the remaining Class 0 patients together with all remaining patients from Class 1. The classifiers are trained with the constructed training set and their classification performances are evaluated on the constructed test set. The results presented in this section is the average of 100 experiment runs with different randomly constructed training and test sets during each run.

We use three metrics to evaluate the classification results, i.e., classification accuracy, AUC-ROC and F1-Score. Accuracy is the most intuitive metric since it indicates the probability of the classification result being correct. The Receiver Operating Characteristic (ROC) curve captures the relationship between true and false positive rates. Therefore, the area-under-curve (AUC) of the ROC curve, i.e., AUC-ROC, is commonly used to measure the performance of a classifier. The F1-score computes the harmonic mean of the precision and recall, which is another popular metric to measure the performance of a binary classifier. We refer the reader to [30] for more information.

Tables I and II present the classification results using the various feature sets and classifiers. The results using SVM are summarized in in Table I. We observe that the features extracted by DFL, especially OSDAE, have the highest classification accuracy. Specifically, with GPR interpolation and SVM, the experiment case D1-G-OSDAE-SVM can classify patients' lengths of stay in the hospital with 71.4% accuracy, which is 18% higher than the classification results using the feature set selected by MFS. The deep learning based feature learning methods, i.e., USDAE and OSDAE, generally perform better than the other feature extraction methods such as RFS and PCA. Furthermore, by comparing OSDAE and USDAE, it can be seen that OSDAE generally performs better than USDAE. This shows the advantage of OSDAE for being able to optimize the structure of the constructed SDAE.

We note that GPR interpolation generally performs better than spline interpolation. This shows spline interpolation is unstable when the number of training samples is small; whereas GPR interpolation is generally stable for varying sample sizes.

Similar results can be seen from the AUC-ROC and F1-score comparisons in Table I. The highest AUC-ROC and F1-score are achieved by D1-G-OSDAE-SVM, which is the highest among all experiment cases.

The classification results with ANN classifier are shown in Table II. Similarly, D1-G-OSDAE-ANN achieves the best results in all performance metrics. With the ANN classifier and GPR interpolation, OSDAE can achieve a classification accuracy of 81.6%, which is around 20% higher than MFS and more than 10% higher than the other experiment cases. The

AUC-ROC and F1-score achieved by D1-G-OSDAE-ANN are also the highest among all experiment cases, which shows the advantage of OSDAE.

VI. Results Using Data Set 2

In this section, we evaluate the performance of DFL using Data Set 2, which is a larger data set consisting of EEG measurements from 600 participants.

A. Feature Selection Methods

1) Manual Feature Selection (MFS): For the EEG data set, MFS extracts features from both time and frequency domains which are similar to the features extracted in the Data Set 1 (pneumonia data set). We compute statistics including Mean, Max, Min and StdDev of the time series data for each EEG channel. In the frequency domain, we compute the DWT and DFT, and select MAC-DWT, MDC-DWT, and LFC-DFT as the feature set for each EEG channel. As before, for the delta change of the EEG time series for each channel, we compute the same time and frequency domain metrics.

In total, MFS extracts 896 features from EEG data set, which consist of 4 statistical features and 3 frequency features from both the original and the delta change of EEG time series for all 64 channels.

2) Raw Feature Selection (RFS): As the EEG data set is structured where the data from each patient have the same size, RFS just uses the original raw data as features. Therefore, for the EEG data set, RFS extracts feature vectors of size 16384 (256 time periods $\times$ 64 channels).

3) Feature Set From PCA: For the EEG data set, we use PCA to generate feature set with 100 principal components.

4) Feature Set Computed From DFL: Similar to the Pneumonia data set, we also construct USDAE and OSDAE to generate feature sets. The standard deviation of the noise level σ for training the SDAE is set to be 0.1 in this experiment. For the EEG data, we define the set of possible numbers of hidden units to be $U = \{16384, 8192, 4096, 2048, 1024, 512\}$. Based on the defined U, by solving the optimization problem in (4) with an additional constraint that all hidden layers have the same size, we find out that the USDAE with 3 hidden layers and 4096 hidden units in each hidden layer can generate the best feature set. By solving the optimization problem in (5a)-(5b), DFL obtains the OSDAE for the EEG data set which has 3 hidden layers with 8192, 4096 and 4096 hidden units, respectively. Therefore, USDAE and OSDAE both extract feature vectors of size 4096.

B. Classification Results Using Data Set 2

In this section, we evaluate the classification performance using the EEG data set following the work flow in Fig. 5. The EEG data set is evaluated via 5-fold cross-validations. The results presented in this section is the average of 100 experiment runs with different randomly constructed training and test sets during each run.

In Tables III and IV, we compare the performance of classification algorithms using the feature sets determined by the various feature selection methods depicted in Fig. 5. It can be seen that OSDAE achieves the highest score in all three performance metrics for both SVM and ANN classifiers. We note that there are existing works in the literature investigating feature selection and classification algorithms using the same EEG data set as used in this paper. To the best of our knowledge, [31] and [32] report the highest achieved classification accuracy of 94.5% and 95.8%, respectively. It can be seen that OSDAE beats both of these methods in terms of classification accuracy using both SVM and ANN classifiers, which further highlights the advantages of DFL.

TABLE III
SVM Classification Using EEG Data

Feature Set	Accuracy	AUC-ROC	F1-score
D2-MFS-SVM	0.942 ($\pm$0.001)	0.969 ($\pm$0.000)	0.967 ($\pm$0.000)
D2-RFS-SVM	0.808 ($\pm$0.019)	0.884 ($\pm$0.005)	0.833 ($\pm$0.001)
D2-PCA-SVM	0.777 ($\pm$0.001)	0.880 ($\pm$0.001)	0.803 ($\pm$0.001)
D2-USDAE-SVM	0.753 ($\pm$0.001)	0.810 ($\pm$0.000)	0.730 ($\pm$0.001)
D2-OSDAE-SVM	**0.988** ($\pm$0.001)	**0.999** ($\pm$0.000)	**0.967** ($\pm$0.001)

TABLE IV
ANN Classification Results Using EEG Data

Feature Set	Accuracy	AUC-ROC	F1-score
D2-MFS-ANN	0.919 ($\pm$0.014)	0.973 ($\pm$0.013)	0.921 ($\pm$0.010)
D2-RFS-ANN	0.753 ($\pm$0.025)	0.836 ($\pm$0.014)	0.754 ($\pm$0.024)
D2-PCA-ANN	0.793 ($\pm$0.001)	0.883 ($\pm$0.001)	0.812 ($\pm$0.004)
D2-USDAE-ANN	0.747 ($\pm$0.002)	0.810 ($\pm$0.001)	0.761 ($\pm$0.000)
D2-OSDAE-ANN	**0.988** ($\pm$0.001)	**0.999** ($\pm$0.001)	**0.988** ($\pm$0.002)

An interesting observation is that, by comparing RFS with USDAE, it can be seen that USDAE is not always better than RFS. In Table III, RFS performs better than PCA and USDA with SVM classifiers. With ANN classifiers, RFS also performs better than USDAE. This shows the evidence that the structure of the SDAE affects the performance of the generated feature set. Without carefully designing the structure of the SDAE, using an SDAE to generate feature set could perform worse than simply using the raw data as features. This shows the importance of the proposed DFL feature learning framework for being able to optimize the structure of the SDAE when generating features. With an optimized SDAE structure, i.e., OSDAE, the generated feature set is the most efficient among all experiment cases.

VII. Conclusion

In this paper, we present a novel scheme to learn an efficient and compact representation of patient health data useful for decision making and diagnosis. The scheme, called DFL, adopts a deep learning approach to construct a feature learning block, built upon the idea of stacked denoising autoencoders, that is able to learn features from patient health data in an unsupervised way. Furthermore, DFL optimizes the structure and parameters of the underlying feature learning block to ensure the utility of the generated feature set. We demonstrate the effectiveness of DFL by applying it to two different classification problems using two real-world patient data sets.

In future work, we plan to investigate the effects of the noise level during training on the performance of the constructed SDAE. Furthermore, we would like to extend the exploration of

efficient feature learning methods to other graph-based models beyond autoencoders, such as dynamic Bayesian networks and deep graphical models.

REFERENCES

[1] D. B. Neill, "Using artificial intelligence to improve hospital inpatient care," *IEEE Intell. Syst.*, vol. 28, no. 2, pp. 92–95, Mar. 2013.

[2] F. Rincon *et al.*, "Impact of delayed transfer of critically ill stroke patients from the Emergency Department to the Neuro-ICU," *Neurocritical Care*, vol. 13, no. 1, pp. 75–81, Aug. 2010.

[3] D. B. Chalfin, S. Trzeciak, A. Likourezos, B. M. Baumann, and R. P. Dellinger, "Impact of delayed transfer of critically ill patients from the emergency department to the intensive care unit," *Crit. Care Med.*, vol. 35, no. 6, pp. 1477–1483, Jun. 2007.

[4] R. Caccialanza *et al.*, "Nutritional parameters associated with prolonged hospital stay among ambulatory adult patients," *Can. Med. Assoc. J.*, vol. 182, no. 17, pp. 1843–1849, Nov. 2010.

[5] F. C. Ryckman, P. A. Yelton, A. M. Anneken, P. E. Kiessling, P. J. Schoettker, and U. R. Kotagal, "Redesigning intensive care unit flow using variability management to improve access and safety," *Joint Commission J. Qual. Patient Saf.*, vol. 35, no. 11, pp. 535–543, Nov. 2009.

[6] P. B. Jensen, L. J. Jensen, and S. Brunak, "Mining electronic health records: Towards better research applications and clinical care," *Nature Rev. Genetics*, vol. 13, no. 6, pp. 395–405, Jun. 2012.

[7] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[8] A. Simpson *et al.*, "Beyond atopy: Multiple patterns of sensitization in relation to asthma in a birth cohort study," *Amer. J. Respiratory Crit. Care Med.*, vol. 181, pp. 1200–1206, Feb. 2010.

[9] F. Wang, N. Lee, J. Hu, J. Sun, S. Ebadollahi, and A. F. Laine, "A framework for mining signatures from event sequences and its applications in healthcare data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 2, pp. 272–285, Feb. 2013.

[10] H. Mamaghanian, N. Khaled, D. Atienza, and P. Vandergheynst, "Compressed sensing for real-time energy-efficient ECG compression on wireless body sensor nodes," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 9, pp. 2456–2466, Sep. 2011.

[11] S. H. Huang, P. LePendu, S. V. Iyer, M. Tai-Seale, D. Carrell, and N. H. Shah, "Toward personalizing treatment for depression: Predicting diagnosis and severity," *J. Amer. Med. Inform. Assoc.*, vol. 21, no. 6, pp. 1069–1075, Nov. 2014.

[12] S. Lyalina, B. Percha, P. LePendu, S. V. Iyer, R. B. Altman, and N. H. Shah, "Identifying phenotypic signatures of neuropsychiatric disorders from electronic medical records," *J. Amer. Med. Inform. Assoc.*, vol. 20, no. e2, pp. e297–e305, Dec. 2013.

[13] X. Wang *et al.*, "Unsupervised learning of disease progression models," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, 2014, pp. 85–94.

[14] R. Pivovarov, D. J. Albers, J. L. Sepulveda, and N. Elhadad, "Identifying and mitigating biases in EHR laboratory tests," *J. Biomed. Inform.*, vol. 51, pp. 24–34, 2014.

[15] N. G. Weiskopf and C. Weng, "Methods and dimensions of electronic health record data quality assessment: Enabling reuse for clinical research," *J. Amer. Med. Inform. Assoc.*, vol. 20, no. 1, pp. 144–151, Jan. 2013.

[16] Y. Mao, W. Chen, Y. Chen, C. Lu, M. Kollef, and T. Bailey, "An integrated data mining approach to real-time clinical monitoring and deterioration warning," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 1140–1148.

[17] J. Yao, C. Zhou, and M. Motani, "Spatio-temporal autoencoder for feature learning in patient data with missing observations," in *Proc. IEEE Int. Conf. Bioinf. Biomed.*, 2017, pp. 886–890.

[18] R. Miotto, L. Li, B. A. Kidd, and J. T. Dudley, "Deep patient: An unsupervised representation to predict the future of patients from the electronic health records," *Scientific Rep.*, vol. 6, May 2016, Art. no. 26094.

[19] Z. Liang, G. Zhang, J. X. Huang, and Q. V. Hu, "Deep learning for healthcare decision making with EMRS," in *Proc. IEEE Int. Conf. Bioinf. Biomed.*, Nov. 2014, pp. 556–559.

[20] C. Zhou, J. Yao, M. Motani, and J. Chew, "Learning deep representations from heterogeneous patient data for predictive diagnosis," in *Proc. ACM Int. Conf. Bioinf., Comput. Biol., Health Inf.*, 2017, pp. 115–123.

[21] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, Dec. 2010.

[22] O. C. Zienkiewicz, "Splines and variational methods," *Int. J. Numer. Methods Eng.*, vol. 10, no. 2, pp. 487–487, 1976.

[23] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2005.

[24] B. K. Beaulieu-Jones and C. S. Greene, "Semi-supervised learning of the electronic health record for phenotype stratification," *J. Biomed. Inform.*, vol. 64, pp. 168–178, 2016.

[25] B. K. Beaulieu-Jones and J. H. Moore, "Missing data imputation in the electronic health record using deeply learned autoencoders," *Pacific Symp. Biocomput.*, vol. 22, pp. 207–218, 2017.

[26] O. Shamir, S. Sabato, and N. Tishby, "Learning and generalization with the information bottleneck," *Theor. Comput. Sci.*, vol. 411, no. 29, pp. 2696–2711, 2010.

[27] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, May 2011.

[28] UCI EEG Data. 2017. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/eeg+database

[29] P. Baldi and K. Hornik, "Neural networks and principal component analysis: Learning from examples without local minima," *Neural Netw.*, vol. 2, no. 1, pp. 53–58, 1989.

[30] D. M. W. Powers, "Evaluation: From precision, recall and f-measure to ROC, informedness, markedness and correlation," *Int. J. Mach. Learn. Technol.*, vol. 2, no. 1, pp. 37–63, 2011.

[31] S. Wang, Y. Li, P. Wen, and D. Lai, "Data selection in EEG signals classification," *Australas. Physical Eng. Sci. Med.*, vol. 39, no. 1, pp. 157–165, Mar. 2016.

[32] G. Zhu, Y. Li, P. P. Wen, and S. Wang, "Analysis of alcoholic EEG signals based on horizontal visibility graph entropy," *Brain Inform.*, vol. 1, no. 1, pp. 19–25, Dec. 2014.