Spatio-Temporal Autoencoder for Feature Learning in Patient Data with Missing Observations

Yao Jia, Chongyu Zhou, and Mehul Motani
Department of Electrical and Computer Engineering
National University of Singapore Singapore
E-mail: eleyaj@nus.edu.sg, chongyuzhou@u.nus.edu, motani@nus.edu.sg

***Abstract*—Modern patient data tends to be large-scale and multi-dimensional, containing both spatial and temporal features. Learning good spatio-temporal features from large patient data is a challenging task, especially when there are missing observations. In this paper, we propose a spatio-temporal autoencoder (STAE), an unsupervised deep learning scheme, to learn features from large-scale and high-dimensional patient data with missing observations. Through both spatial and temporal encoding, STAE is able to automatically identify patterns and dependencies in the patient data, even with missing values, and learn a compact representation of each patient for better classification. Publicly available electroencephalogram (EEG) data are extracted from the UCI Machine Learning Repository to test and support our findings. Through simulations, we compare STAE with several baseline feature selection methods and demonstrate its effectiveness in the presence of missing data.**

I. Introduction

Modern medicine is being transformed into a paradigm of *precision medicine* [1] where machine learning tools are applied to analyze patient data in order to precisely characterize each individual patient and better inform clinical decision making. It is widely recognized that the performance of machine learning methods heavily depends on the choice of data representation (or features) on which they are applied [2]. However, despite the recent advances of feature extraction techniques in various application areas such as computer vision, pattern recognition and text mining etc., unique challenges exist in learning efficient data representations for analyzing patient data to support clinical decisions.

First, understanding patient data usually requires deep domain knowledge. Therefore, in the literature, a common approach to identify features in patient data is to have some domain experts to designate the patterns to look for in an ad-hoc manner [3]. This kind of supervised definitions of the feature space, though appropriate in some situations, scales poorly and does not generalize well. Therefore, an unsupervised manner of feature selection is preferred to learn the feature from the patient data directly.

Second, with the advance of modern patient monitoring systems and the increasing availability of electronic health record (EHR), modern patient data, such as vital-sign measurements, are stored as multivariate time series with high sampling frequencies [4]. Processing and learning features from such large-scale and high-dimensional patient data is usually difficult or sometimes impossible. Therefore, statistical signal processing techniques are usually applied to encode the patient data into a lower dimension for efficient processing. However, how to encode the patient data without losing useful signatures in the data is a non-trivial problem.

Another challenge in analyzing patient data is due to missing values. Due to various reasons, such as human errors, machine faults, power loss and so on, practical patient data often inevitably carry missing observations [5], [6]. It has been noted that these missing values are sometimes informative which may provide rich information in supervised learning tasks, e.g., time series classification [7]. Therefore, one should not simply ignore these missing values during feature learning. With missing values, feature learning in patient data becomes even more challenging [3], [8].

In this paper, we propose a spatio-temporal autoencoder (STAE) to learn features from large-scale and high-dimensional patient data with missing observations. Through a deep learning approach, STAE attempts to capture signatures in patient data from both spatial and temporal domains. Motivated by denoising autoencoders [9], [10], STAE first capture the spatial structure in the data by encoding the noisy patient data into a smaller space without breaking the temporal correlations. Next, STAE captures the temporal structure in the data using a temporal autoencoder containing a set of Long short-term memory (LSTM) blocks, which are a type of building blocks in recurrent neural networks (RNNs). The proposed STAE scheme is unsupervised in nature. By jointly training the spatial and temporal autoencoders, STAE is able to automatically identify patterns and dependencies in the patient data from both spatial and temporal domains, even with missing values, and learn a compact representation of each patient for better classification.

A publicly available electroencephalogram (EEG) data set is extracted from the UCI Machine Learning Repository to test the performance of the proposed STAE scheme where a classification problem of determining whether a patient is under alcoholic control or not is conducted. By comparing with several baseline feature selection approaches, it is shown that the proposed STAE scheme can achieve better results in the classification problem investigated.

The rest of this paper is organized as follows. Section II reviews some related work in patient data analysis and feature selection algorithms. In Section III, we introduce the proposed STAE scheme. Section IV introduces the details of the data set

978-1-5090-3050-7/17/$31.00 ©2017 IEEE

used and the classification problem investigated in this paper. The experiment results are presented in Section V where the performance of the proposed STAE feature learning scheme is evaluated. Finally, Section VI concludes the paper.

II. Related Work

Supervised feature selection methods have been extensively explored in analyzing patient data. In particular, a hidden Markov model (HMM) has been used in [11] to learn the patterns of the patients from a set of Asthma patient data. A nonnegative matrix factorization (NMF) based algorithm is proposed in [12] for mining temporal signatures from a set of patient data for event detection applications. Furthermore, a compressive sensing (CS) based method is adopted in [13] to detect abnormal events in ECG signals. However, such supervised feature selection methods assume prior knowledge in the structure of the data model, which normally come from domain experts. As discussed above, these kind of supervised feature selection methods are labor intensive and hard to scale.

As a result, unsupervised feature learning methods which automatically learn features from patient data without domain knowledge and human labeling have drawn huge research interests recently. Plenty of threshold-based feature metrics, such as [14], [15], have been proposed for classifying patient data. We note that such ad-hoc feature selection methods are not generic and may only be suitable for some specific problems. Generic feature learning algorithms based on deep learning are proposed in [5], [16], [17]. A stacked denoising autoencoder-based method is proposed in [5] for predictive diagnosis problems. A similar algorithm has been proposed in [16] for disease classification. Instead of autoencoders, a deep belief network-based algorithm has been proposed in [17] to learn features from patient data for clinical decision making. We note that the methods proposed in [5], [16], [17] only focus on extracting spatial features and fail to capture the temporal features. Furthermore, the methods proposed in [5], [16], [17] cannot deal with missing values. This is where the proposed STAE scheme tries to fill the gap.

III. Methodology

The overall structure of the proposed STAE scheme is presented in Fig. 1, which consists a temporal autoencoder nested into a spatial autoencoder. Let $\boldsymbol{X} = \{X_t\}_{t=1}^{T}$ be the raw data of a patient, which are multi-dimensional time series containing T time slots. $X_t \in \mathbb{R}^d$ is a vector of measurements at time slot t, where d is the dimension of the measurement vector. For example, if X_t is the set of vital-sign measurements at time t, d will be the number of vital-signs types measured. In the STAE scheme, the raw patient data $\boldsymbol{X}$ go through two stages of encoding. First, $\boldsymbol{X}$ go though a spatial encoder E to get a compact representation $\boldsymbol{Y} = \{Y_t\}_{t=1}^{T}$. We note that $\boldsymbol{Y}$ only captures the spatial features $\boldsymbol{X}$. In order to extract the temporal features in $\boldsymbol{X}$, we further encode $\boldsymbol{Y}$ using a set of LSTMs to get a more compact representation $\boldsymbol{h}_T$. $\boldsymbol{h}_T$ will be used as the feature set for classification tasks. In the following, we describe the details of the spatial autoencoder and the temporal autoencoder modules, respectively.

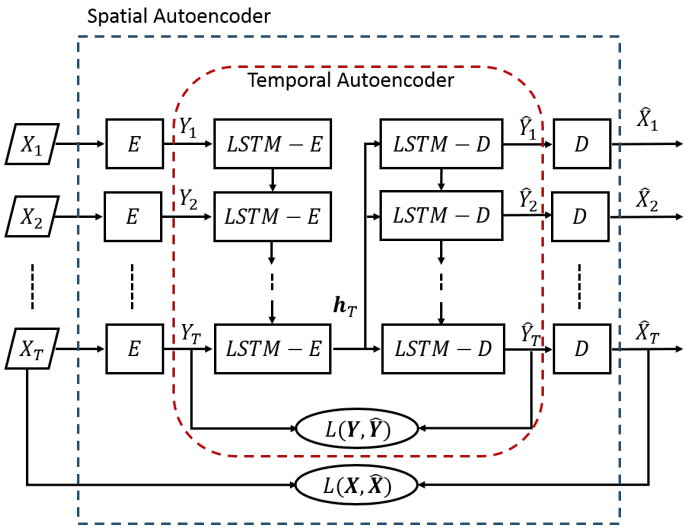

Fig. 1: Spatio-temporal autoencoder.

A. Spatial Autoencoder

The spatial autoencoder in STAE is a classic encoder-decoder architecture [9], [10] based on multi-layer neural networks. The encoder E consists of a multi-layer neural network, where each hidden layer of the network is trained to produce a higher-level representation of the input data. This is done by optimizing a local unsupervised criterion based on the data it receives from the previous layer. Therefore, every layer produces a representation of the input pattern that is more abstract than the previous layer. The decoder D mirrors the structure of the encoder.

Through the encoder E, the input signal $\boldsymbol{X} \in \mathbb{R}^{d \times T}$ is mapped to a hidden representation $\boldsymbol{Y} \in \mathbb{R}^{d' \times T}$ through a deterministic mapping:

$$\boldsymbol{Y} = f_{\boldsymbol{\theta}^{se}}(\boldsymbol{X}) = s(\mathbf{W}\mathbf{X} + \mathbf{b}), \tag{1}$$

parameterized by $\boldsymbol{\theta}^{se} = (\boldsymbol{W}, \boldsymbol{b})$ where $\boldsymbol{W}$ is a weight coefficient matrix, and $\boldsymbol{b}$ is a bias vector. $s(\cdot)$ is a non-linear transformation, which is also referred to as an activation function. In this paper, we use sigmoid function as the activation function in the spatial autoencoder. The hidden representation $\boldsymbol{Y}$ is then passed into the temporal (LSTM) autoencoder for further processing (see Section III-B). Let $\hat{\boldsymbol{Y}} \in \mathbb{R}^{d' \times T}$ be the output signal from the temporal autoencoder which have the same size as $\boldsymbol{Y}$. Then, $\hat{\boldsymbol{Y}}$ is mapped back to a reconstructed signal $\hat{\boldsymbol{X}} \in \mathbb{R}^{d \times T}$ through a decoder:

$$\hat{\boldsymbol{X}} = g_{\boldsymbol{\theta}^{se'}}(\hat{\boldsymbol{Y}}) = s(\mathbf{W}'\hat{\boldsymbol{Y}} + \mathbf{b}'), \tag{2}$$

parameterized by $\boldsymbol{\theta}^{se'} = (\mathbf{W}', \mathbf{b}')$. The parameters $\boldsymbol{\theta}^{se}$ and $\boldsymbol{\theta}^{se'}$ are trained to minimize the average reconstruction error over a training set, which is measured by some loss function $L(\boldsymbol{X}, \hat{\boldsymbol{X}})$. We explored the use of both the mean squared error

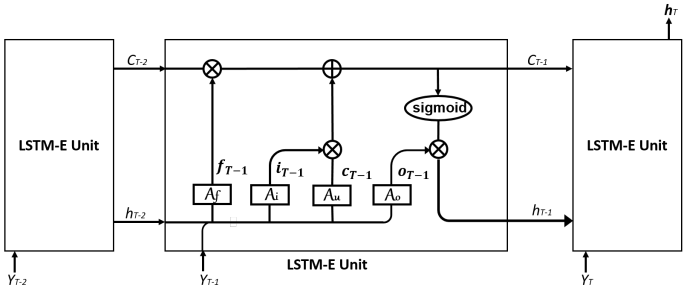

Fig. 2: LSTM-E Unit.

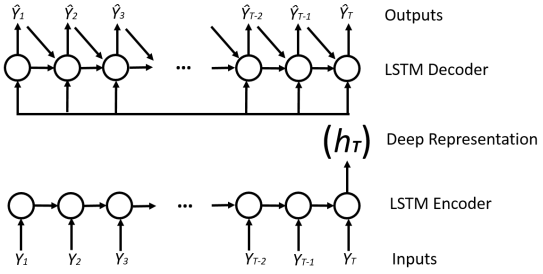

Fig. 3: LSTM Encoder-Decoder.

(MSE) and the cross-entropy as the loss function for training the parameters.

By minimizing the reconstruction error between $\boldsymbol{X}$ and $\hat{\boldsymbol{X}}$, the spatial encoder-decoder E and D also minimize the reconstruction error between $\boldsymbol{X}$ and $\boldsymbol{Y}$ [10], [18]. Therefore, $\boldsymbol{Y}$ can be treated as a compact representation of the raw data $\boldsymbol{X}$ that captures the coordinates along the main factors of variation in the data. As the structure of the spatial autoencoder $E-D$ is only capable to capture the spatial correlations among the data, another layer of autoencoder need to be constructed to capture the temporal correlations in $\boldsymbol{X}$. In Section III-B, we introduce a temporal autoencoder based on LSTM.

B. Temporal (LSTM) Autoencoder

After spatial encoding, the encoded data $\boldsymbol{Y}$ are passed into a temporal autoencoder to extract the temporal features. In STAE, the temporal autoencoder is constructed by a set of LSTM cells, which is a special type of RNN.

RNN is a sequence dependent neural network as it considers the current inputs as well as the decision made in the previous time step. It is effective in learning features from time series data with temporal information. LSTM is a special RNN which has long-short-term memory units in the hidden layer. It is specifically designed for long sequences. LSTM units have an ability to optionally keep particular information from the previous state, and to choose certain information on the current state to be updated.

In the STAE scheme, the temporal information of the data in each time slot is taken care by one LSTM-based temporal encoder, which is referred to as an LSTM-E unit. Therefore, T LSTM-E cells are used in the STAE scheme. Here, we simply describe the architecture of the LSTM-E units used which is based on the standard LSTM unit designed by Hochreiter and Schmidhuber [19] with a minor modification. Fig. 2 shows the architecture of one LSTM-E unit. Each LSTM-E unit consist of four gates, namely, forget gate, input gate, update gate and output gate, which are denoted as A_f, A_i, A_u and A_o, respectively. Each gate takes in both the current input data Y_t and the output signals from previous LSTM-E unit, i.e., h_{t-1} and C_{t-1}. The gates of each LSTM-E cell $t \in \{1, 2, ..., T\}$ operate as follows:

$$f_t = A_f(W_f[h_{t-1}, Y_t] + b_f) \tag{3}$$
$$i_t = A_i(W_i[h_{t-1}, Y_t] + b_i) \tag{4}$$
$$c_t = A_u(W_c[h_{t-1}, Y_t] + b_c) \tag{5}$$
$$C_t = c_{t-1} * f_t + i_t * c_t \tag{6}$$
$$o_t = A_o(W_o[h_{t-1}, Y_t] + b_o) \tag{7}$$
$$h_t = o_t * \text{sigmoid}(C_t). \tag{8}$$

In (3)-(8), f_t, i_t, c_t and o_t are the output signals from the four gates, respectively. Then, the combined signals, C_t and h_t, are the ones which will be passed to the next LSTM-E cell corresponding to time slot $t+1$. The vector $\boldsymbol{\theta}^{te} = \{W_f, b_f, W_i, b_i, W_c, b_c, W_o, b_o\}$ are the parameters for each LSTM-E unit. In this paper, we assume that the LSTM-E units at different time slots share the same set of parameters.

In order the train the temporal autoencoder, i.e., the LSTM-E units, a set of LSTM-D units are included, which have mirrored structure with the LSTM-E units, as shown in Fig. 3. We assume that all the LSTM-D units share the same set of parameters, denoted by $\boldsymbol{\theta}^{te'}$.

The training process of the temporal autoencoder works as follows. The encoded signal $\boldsymbol{Y} \in \mathbf{R}^{d' \times T}$ is first passed into the temporal encoder, i.e., the LSTM-E units, where the temporal variation in $\boldsymbol{Y}$ is captured. As a result, the output signal from the Tth LSTM-E units, i.e., $\boldsymbol{h}_T$, contains the temporal information in $\boldsymbol{Y}$ for all time slots. We note that $\boldsymbol{h}_T$ has a dimension of $d' \times T$. $\boldsymbol{h}_T$ is then passed into the temporal decoder, i.e., the LSTM-D units, for decoding in order to reconstruct $\boldsymbol{Y}$. Let $\hat{\boldsymbol{Y}} \in \mathbf{R}^{d' \times T}$ be the reconstructed signal from the temporal decoder based on $\boldsymbol{h}_T$. The parameters of the temporal autoencoders, i.e., $\boldsymbol{\theta}^{te}$ and $\boldsymbol{\theta}^{te'}$ are trained by minimizing the reconstruction error according to a loss function $L(\boldsymbol{Y}, \hat{\boldsymbol{Y}})$. Similar to the spatial autoencoder, in this paper, we use MSE as the loss function for training the parameters in the temporal autoencoder.

It can be seen that the temporal autoencoder can be trained in a unsupervised manner. After the parameters are trained, the encoded signal $\boldsymbol{h}_T$ can be treated as a deep representation of the input sequence $\boldsymbol{Y}$ that captures the temporal correlations in $\boldsymbol{Y}$ [20] [21]. As introduced in Section III-A, $\boldsymbol{Y}$ only does spatial encoding in $\boldsymbol{X}$ and does not break the temporal structure of $\boldsymbol{X}$. Therefore, $\boldsymbol{h}_T$ captures the temporal correlations

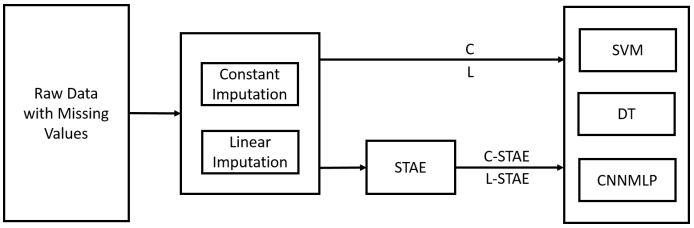

Fig. 4: Illustration of System Workflow.

in $\boldsymbol{X}$ as well. Furthermore, as $\boldsymbol{h}_T$ is encoded based on $\boldsymbol{Y}$, $\boldsymbol{h}_T$ also captures the spatial correlations in $\boldsymbol{X}$. As a result, $\boldsymbol{h}_T$ is a spatio-temporal deep representation of $\boldsymbol{X}$. In the experiments in Section V, $\boldsymbol{h}_T$ will be used as the features for the considered classification tasks.

IV. Data Set and Problem Description

A. Electroencephalogram (EEG) Data Set

1) Data Set Description: The data set used in this paper is a EEG data set extracted from the UCI Machine Learning Repository [22]. This data set arises from a large study to examine EEG correlates of genetic predisposition to alcoholism.

Only a subset of the whole data set are used for our study, which contains 600 EEG time series data. Each piece of EEG time series contains a 64-channel measurement for 256 time periods. Therefore, each EEG time series has a dimension of 64×256. Each piece of EEG time series data is labeled as alcoholic (Class 0) or non-alcoholic (Class 1). Each class has 300 instances.

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 his EEG measurements.

As there are no missing observations in the data set, we randomly removed 30% of the measurements from the EEG data set and marked them as missing observations. Hereinafter, we referred to the original EEG data set without missing observations as EEG_ORIGINAL and the pre-processed EEG data with 30% missing observations as EEG.

B. Missing Observations Marking

In this paper, we consider two types of marks for the missing observations, namely *constant imputation* and *linear imputation*. For constant imputation, we simply mark the missing observations as constant outliers. For linear imputation, we try to impute the missing observations using piecewise linear regression techniques.

V. Performance Evaluation

Fig. 4 shows the workflow for evaluating the performance of the proposed STAE scheme. Four feature sets are extracted from the EEG data for comparison. The first and second feature sets are obtained by imputing the missing values in the EEG data via constant imputation (denoted by "C") and linear imputation (denoted by "L"), respectively. The third and fourth feature sets are obtained by passing the imputed data through the proposed STAE scheme, denoted by "C-STAE" and "L-STAE", respectively.

TABLE I: SVM classification with EEG data set

Dataset and Classifier	AUC-ROC	Cross Validation Accuracy
EEG_ORIGINAL	0.8840	0.7950
C-SVM	0.5762	0.5370
L-SVM	0.7275	0.6700
C-STAE-SVM	0.7830	0.7070
L-STAE-SVM	0.7900	0.7230

TABLE II: DT classification with EEG data set

Dataset and Classifier	AUC-ROC	Cross Validation Accuracy
EEG_ORIGINAL	0.6820	0.6820
C-DT	0.5405	0.5105
L-DT	0.5842	0.5650
C-STAE-DT	0.6465	0.6380
L-STAE-DT	0.6136	0.6060

TABLE III: CNNMLP classification with EEG data set

Dataset and Classifier	AUC-ROC	Cross Validation Accuracy
EEG_ORIGINAL	0.9584	0.9009
C-CNNMLP	0.5084	0.5350
L-CNNMLP	0.6272	0.6330
C-STAE-CNNMLP	0.8006	0.6616
L-STAE-CNNMLP	0.8052	0.6882

A. Classification Methods

Three classifiers are used to evaluate the performance of the feature sets: support vector machines (SVMs) [23], decision trees (DTs) [24] and convolutional neural networks (CNNs) [25]. Specifically, for the SVM classifier, C-support Vector Classification (SVC) and linear kernel are used. The CNN used in our study contains one convolutional layer with number of filters equal to 150 and kernel size equal to 2. One pooling layer with pool size equal to 3 is added after the convolutional layer. The outputs from the CNN are fully connected with a multi-layer perceptron network which consists of one hidden layer with 50 neurons (we refer to this classifier as CNNMLP).

5-fold cross validation is used to evaluate the classification performance. The AUC-ROC and cross validation accuracies are shown in Tables I, II and III. The results presented are averaged over 10 experimental runs.

B. Classification Results for EEG Data Set

Tables I, II and III show the classification results for the EEG data set (with 30% missing observations). We label each testing case by the feature set followed by the classifier. For example, in Table I, "C-SVM" indicates the testing case to be

the feature set "C" (constant imputation) with classifier SVM. For comparison purposes, the results for EEG_ORIGINAL (original EEG data set with no missing observations) with each classifier are also shown.

It can be seen from Table I that, using the SVM classifier, by comparing EEG_ORIGINAL with C-SVM and L-SVM, the AUC-ROC score drops by around 40% and 20%, respectively. The cross validation accuracy also drops by around 30% and 15%, respectively. Therefore, it can be seen that the 30% missing observations can greatly affect the performance of the SVM classifier. With STAE, as part of the spatio-temporal variations in the data are recovered, we can improve both the AUC-ROC score and the cross validation accuracy by some margin. The results from C-STAE-SVM and L-STAE-SVM, though still being lower than EEG_ORIGINAL, are much better than C-SVM and L-SVM in terms of both AUC-ROC score and the cross validation accuracy. Similar observations can be seen from Table II and Table III for the DT and CNNMLP classifiers as well.

The results presented in Tables I to III have shown that the proposed STAE scheme is more effective in extracting spatio-temporal features from the patient data compared to the baseline schemes. Therefore, the AUC-ROC scores and the cross-validation accuracies achieved by STAE are much higher than those achieved by other baseline feature selection methods. One interesting observation is that, with STAE, linear imputation is not always better than constant imputation. It can be seen that with SVM and CNN, the performance of L-STAE is better than C-STAE. However, the opposite is observed with DT classifier. As the imputation methods can potentially affect the performance of STAE, the performance of STAE can be further improved by automatically determining the imputation methods given a classifier. We will consider such improvements in future work.

VI. Conclusion

In this paper, we propose STAE, an unsupervised deep learning scheme, to learn features from large-scale and high-dimensional patient data with missing observations. Through both spatial and temporal encoding, STAE is able to automatically identify patterns and dependencies in the patient data, even with missing values, and learn a compact representation of each patient for better classification. To test and verify the performance of STAE, we extract an EEG data set from the publicly available UCI ML Repository [22]. Via extensive experiments and by comparing with several baseline feature selection methods, we demonstrate the effectiveness of STAE in extracting spatio-temporal features in the patient data even with missing observations.

VII. Acknowledgments

This research was supported in part by the Singapore Ministry of Education under grant R-263-000-B61-112. We would like to thank our collaborators from the National University Hospital in Singapore for giving constructive feedback.

References

[1] F. S. Collins and H. Varmus, "A new initiative on precision medicine," *New England Journal of Medicine*, vol. 372, no. 9, pp. 793–795, 2015.

[2] 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.

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

[4] 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 KDD*, 2012, pp. 1140–1148.

[5] C. Zhou, J. Yao, M. Motani, and J. Chew, "Learning deep representations from heterogeneous patient data for predictive diagnosis," in *Proc. ACM BCB*, ser. BCB '17, 2017.

[6] R. Pivovarov, D. J. Albers, J. L. Sepulveda, and N. Elhadad, "Identifying and mitigating biases in EHR laboratory tests," *Journal of Biomedical Informatics*, vol. 51, pp. 24 – 34, 2014.

[7] D. B. RUBIN, "Inference and missing data," *Biometrika*, vol. 63, no. 3, pp. 581–592, 1976.

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

[9] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. ICML*, 2008, pp. 1096–1103.

[10] 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.

[11] A. Simpson et al., "Beyond atopy: Multiple patterns of sensitization in relation to asthma in a birth cohort study," *American Journal of Respiratory and Critical Care Medicine*, vol. 181, pp. 1200–1206, February 2010.

[12] F. Wang et al., "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.

[13] 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, Sept 2011.

[14] Q. Cheng, J. Shang, J. Juen, J. Han, and B. Schatz, "Mining discriminative patterns to predict health status for cardiopulmonary patients," in *Proc. ACM BCB*, 2016, pp. 41–49.

[15] Y. Sha, J. Venugopalan, and M. D. Wang, "A novel temporal similarity measure for patients based on irregularly measured data in electronic health records," in *Proc. ACM BCB*, 2016, pp. 337–344.

[16] R. Miotto et al., "Deep patient: An unsupervised representation to predict the future of patients from the electronic health records," *Scientific Reports*, vol. 6, pp. 26 094 EP –, May 2016.

[17] Z. Liang, G. Zhang, J. X. Huang, and Q. V. Hu, "Deep learning for healthcare decision making with emrs," in *Proc. IEEE BIBM*, Nov 2014, pp. 556–559.

[18] A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Comput.*, vol. 7, no. 6, pp. 1129–1159, Nov. 1995.

[19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[20] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. NIPS*, 2014, pp. 3104–3112.

[21] K. Cho et al., "On the properties of neural machine translation: Encoder-decoder approaches," in *Proc. SSST*, 2014.

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

[23] 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.

[24] P. E. Utgoff, "Incremental induction of decision trees," *Mach. Learn.*, vol. 4, no. 2, pp. 161–186, Nov. 1989.

[25] B. Zhao et al., "Convolutional neural networks for time series classification," *Journal of Systems Engineering and Electronics*, vol. 28, no. 1, pp. 162–169, Feb 2017.