JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 28, Number 2, 2021

Pp. 146–165
DOI: 10.1089/cmb.2020.0252

Diagnosis of Autism Spectrum Disorder Based on Functional Brain Networks with Deep Learning

WUTAO YIN,[1] SAKIB MOSTAFA,[2] and FANG-XIANG WU[3]

ABSTRACT

Autism spectrum disorder (ASD) is a neurological and developmental disorder. Traditional diagnosis of ASD is typically performed through the observation of behaviors and interview of a patient. However, these diagnosis methods are time-consuming and can be misleading sometimes. Integrating machine learning algorithms with neuroimages, a diagnosis method, can possibly be established to detect ASD subjects from typical control subjects. In this study, we develop deep learning methods for diagnosis of ASD from functional brain networks constructed with brain functional magnetic resonance imaging (fMRI) data. The entire Autism Brain Imaging Data Exchange 1 (ABIDE 1) data set is utilized to investigate the performance of our proposed methods. First, we construct the brain networks from brain fMRI images and define the raw features based on such brain networks. Second, we employ an autoencoder (AE) to learn the advanced features from the raw features. Third, we train a deep neural network (DNN) with the advanced features, which achieves the classification accuracy of 76.2% and the receiving operating characteristic curve (AUC) of 79.7%. As a comparison, we also apply the same advanced features to train several traditional machine learning algorithms to benchmark the classification performance. Finally, we combine the DNN with the pretrained AE and train it with the raw features, which achieves the classification accuracy of 79.2% and the AUC of 82.4%. These results show that our proposed deep learning methods outperform the state-of-the-art methods.

Keywords: autism spectrum disorder, autoencoder, deep learning, functional brain network, functional magnetic resonance imaging.

1. INTRODUCTION

Autism spectrum disorder (ASD) is a neurological and developmental disorder that begins early in childhood and lasts even one's whole life. ASD affects patients' behavior and social interactions. Patients have shown a wide variety of behavioral abnormalities such as impaired social skills, incoherent speech/behaviors, and attention deficit, which give the term spectrum in ASD naming. According to

[1]Division of Biomedical Engineering, University of Saskatchewan, Saskatoon, Canada.
[2]Department of Computer Science, University of Saskatchewan, Saskatoon, Canada.
[3]Division of Biomedical Engineering, Department of Mechanical Engineering, Department of Computer Science, University of Saskatchewan, Saskatoon, Canada.

Hirvikoski et al. (2016), ASD patients are more likely subjected to premature death than healthy controls, and unfortunately, there are no effective medical treatments for ASD. However, an early diagnosis can help clinicians and caregivers to take preliminary measures and provide necessary preventions to maintain certain level of patients' normal lives. Early diagnosis has been proven to be a no-easy task after many years' extensive research. Traditional questionnaire-based diagnosis methods include Autism Diagnostic Observation Schedule (Lord et al., 1989) and Autism Diagnostic Interview (Lord et al., 1994), where the diagnosis is typically performed through the observation of behaviors and assessment of patients' interview answers. Unfortunately, these diagnosis methods are subjective and inefficient and can be misleading sometimes because there are no specific identifiable behaviors that can be described as ASD in an objective way. Therefore, it is imperative to devise reliable methods that can diagnose ASD more accurately and efficiently in a quantitative or semiquantitative way without purely relying on behavioral questions.

With the advance of neuroimaging technology (Khosla et al., 2019; Martino, 2019) and artificial intelligence (Russell and Norvig, 2020), neuroimages have been widely used to understand brain disorders through functional and/or structural studies. Among all the imaging modalities, magnetic resonance imaging (MRI) is one of the excellent tools to study brain disorder due to its high resolution and noninvasive imaging mechanism. MRI can be used to extract information about physical structures and functional activities of human brains using different imaging configurations and protocols.

Of the different MRI techniques, the resting-state functional MRI (rs-fMRI) provides information about the neural activities of human brains by measuring blood oxygen levels variations, that is, blood oxygen level dependent (BOLD) signal, at regular time steps. The BOLD signal reflects brain activities and hence gives an indirect way to study the brain functionalities and brain networks in a noninvasive way. Instead of studying the raw fMRI images data sets due to its high-dimension but low-samples characteristics, an effective alternative approach is to use the graph-theoretic or network-based theory to extract the hidden features out of the fMRI data and reduce the noise components and compress the raw fMRI data for further artificial intelligence algorithms. A brain network is composed of nodes and edges connecting nodes, where the nodes are the regions of interest (ROIs). In network-based approaches, a brain network is constructed from fMRI data by parcellating a brain cortex into different ROIs. In a weighted functional brain network, ROIs are nodes while their edge weights are typically determined, by the (full or partial) Pearson correlation coefficient (PCC), other mathematical measures such as tangent space (Dadi et al., 2019) or recently developed dynamic time warping (Meszlenyi et al., 2017), between ROI pairs extracted out of blood oxygen level time series signals.

With the ever-increasing amount of brain neuroimaging data, machine learning and artificial intelligence (Murphy, 2013; Russell and Norvig, 2020) enabled algorithms are widely adapted to extract the brain functional connectivity networks and hence detect meaningful biomarkers for ASD diagnosis. Many studies incorporated features extracted from brain functional networks into machine learning pipelines to detect ASD subjects from typical control subjects (Shen et al., 2017; Lundervold, 2019; Zhu et al., 2019; Yin et al., 2020).

Heinsfeld et al. (2017) proposed a transfer learning-based ASD diagnosis method using brain functional networks extracted from fMRI time series by using PCCs from the Autism Brain Imaging Data Exchange 1 (ABIDE 1) data set (Poldrack et al., 2017; Martino, 2019) to detect ASDs. A deep neural network (DNN) classifier was first pretrained using a stacked denoising autoencoder (AE) and the input features fed into the AEs were the upper half of the connectivity matrix constructed by the PCCs of all pairwise ROIs. The authors reported a classification accuracy of 70.0% for the entire ABIDE 1 data set, and it was the state-of-the-art at the publishing time. In Eslami et al. (2019), a classification accuracy of 70.1% was reported for the diagnosis of ASD on the ABIDE 1 data set. Eslami et al. also utilized an AE to pretrain a single layer perceptron, and the trained AE was used as a feature extractor and the perceptron as the classifier. It was reported an average classification accuracy of 63.0% for the individual sites of ABIDE 1. The Riemannian geometry of the functional connectivity was studied in Wong et al. (2018). Using the log-Euclidean and affine-invariant Riemannian matrices in the machine learning algorithms, the authors achieved an accuracy of 71.1% for the entire ABIDE 1 data set.

In a DNN-based ASD diagnosis method presented in Kong et al. (2019), the authors extracted features from brain networks and used the F-score to select the dominant features. The features were then fed into the machine learning pipeline as inputs to a deep learning-based classifier. The deep learning-based classifier consisted of two stacked AEs and a softmax layer for classification output. The authors achieved a classification accuracy of 90.4%. However, the data were only from a single site of the ABIDE 1 data set, and hence, it did not reflect the site variance and the fMRI scanner configuration variations and usually this

type of variance was critical for practical clinical applications due to different application scenarios. In some other studies, the higher classification accuracy for diagnosing ASD subjects was also achieved (Yahata et al., 2016; Watanabe and Rees, 2017). These studies also only included certain parts of the ABIDE 1 data set instead of the whole one and they could not generalize well enough to accommodate the site variations when including the entire ABIDE 1 data set. Hence, these types of study were not suitable for practical deployment. According to Arbabshirani et al. (2017), the authors stated that the studies related to the diagnosis of ASD tended to have high accuracy for the data set from a single site and the accuracy declined with the data set from multiple sites.

Therefore, it is imperative to design diagnosis methods with high classification accuracy over the whole ABIDE 1 data set, and it can also generalize well to accommodate different scanners and site configurations with potential to be deployed in real clinical settings. This type of generalizations is highly sought after by hands-on practitioners as the real-world applications are not necessarily well aligned with protocol, and it is very common for medical research and data acquisition that some data fields or items are missing. In Mostafa et al. (2019), a new set of features was proposed for the diagnosis of ASD by turning to graph theories for better hidden feature extractions and compressions. In this article, the spectrum of the Laplacian matrix of brain functional networks was introduced as new high-level features and then combined with three network centralities: assortativity, clustering coefficient, and average degree. Even using traditional machine learning algorithms such as support vector machines (SVM), a classification accuracy of 77.7% was achieved, which is the highest at that time over the entire ABIDE 1 data set (Heinsfeld et al., 2017). This have demonstrated that the features extracted from a graph theory point of view is very effective, and it also serves as a mechanism to reduce the dimension of the typical high-dimension learning problem related to fMRI data processing.

In this study, we are extending the work proposed in Mostafa et al. (2019) and Mostafa et al. (2020) and studying the performance with different configurations and parameterizations. The same 871 subjects from the ABIDE 1 data set are used to conduct this study to accommodate the site variations and scanner configuration differences. The raw features introduced in Mostafa et al. (2019) and Mostafa et al. (2020), that is, the spectrum of Laplacian matrices, assortativity, clustering coefficient, and average degree of brain networks are adapted to be used in this study, and then, an AE is trained to learn the hidden representations from the raw brain network connectivity models. The learned high-level hidden representations are then used to train several machine learning models such as SVM, K-nearest neighbor (KNN), and DNN to benchmark different classifiers for the diagnosis of ASD.

The average performances are comparable to those of the state-of-the-art methods (Yahata et al., 2016; Arbabshirani et al., 2017; Watanabe and Rees, 2017), for example, a DNN with the graph-theoretic-based features can achieve the classification accuracy of 76.2% and the receiver operating characteristic curve (AUC) of 79.7%. As a comparison, we also apply the same feature sets to train several traditional machine learning algorithms to benchmark the classification performance. We further combine the DNN with the pretrained AE and then train the AE-based feature extractor and the classification network with raw features. The mixed encoder and classification network achieve an accuracy of 79.2% and the AUC of 82.4%, and it gives better results than that reported in Mostafa et al. (2019). We also adapt the learnt hidden representations by AE network and DNNs to traditional machine learning algorithms such as SVM, KNN, and subspace discriminant. These shallow models also show improved performance with the best classification accuracy of 74.6% and the AUC of 78.7%.

In summary, the reported graph theory enabled features that are effective in both feature extraction and dimension reduction. The features extracted by resorting to the spectrum of Laplacian matrix and other graph characteristics are capturing the important hidden representation of the human brain functional connectivity networks constructed by using PCCs. It is also demonstrated that these features can also help the traditional machine learning methods, and hence it is, in turn, a proof that the proposed feature construction methods are valid and effective.

2. METHODS

2.1. ABIDE data set

ABIDE 1 data set is a frequently used data set related to the diagnosis of ASD. In this data set, there include rs-fMRI images, T1-weighted images, and phenotypic information of subjects suffering from ASD as well as healthy ones. The data were collected from 17 different research sites around the globe. In ABIDE 1, there are

in total 1112 test subjects. Of the total 1112 subjects, 539 are ASD subjects and 573 are healthy control subjects. Analyzing the image acquisition techniques and phenotypic information provided in Poldrack et al. (2017) and Martino (2019), it is evident that ABIDE 1 covers a wide range of scanners, configuration parameters, age groups, and so on. Because of the heterogeneity and variations of subjects and scanning protocol variations among intersite data, ABIDE 1 is in general a rather complicated data set to deal with. It is typically observed that many machine learning algorithms, shallow or deep models, cannot generalize well across the whole ABIDE I data set. Therefore, a machine learning method that can perform well across the whole ABIDE 1 may have the potential to deal with the variations of different scanner configurations and phenotypic differences and may be suitable to generalize well in the real-world clinical settings.

To be consistent and fair, we compare the proposed methods in this article with other publications (Yahata et al., 2016; Arbabshirani et al., 2017; Heinsfeld et al., 2017; Watanabe and Rees, 2017; Mostafa et al., 2019; Mostafa et al., 2020) under the same structure and similar setups. Hence, we have experimented with the same 871 subjects, among which 403 are ASD subjects and 468 are typical control subjects. And this more or less aligns with other state-of-the-art publications.

2.2. Data preprocessing and brain networks

Instead of creating brain functional networks from the raw rs-fMRI images, it is necessary to preprocess fMRI raw data to remove motion artifacts and perform other denoising procedures. The preprocessing pipeline is fairly straightforward and standard. We apply the AFNI (Analysis of Functional NeuroImages) (Cox, 1996) and FSL (FMRIB's Software Library) (Jenkinson et al., 2012) software packages to both rs-fMRI and T1-weighted images. The purpose of using T1-weighted images is to register rs-fMRI images to the normalized Montreal Neurological Institute (MNI) space/template. The same preprocessing steps as in Mostafa et al. (2019) and Mostafa et al. (2020) are used to remove the noises from rs-fMRI data and hence get prepared for subsequent ROIs extraction steps.

The 264 ROI-based parcellation scheme reported in Power et al. (2011) was used to parcellate the brain region into 264 ROIs for fMRI time series extraction. Specifically, a brain cortex was divided into 264 ROIs, and each ROI is referred as a node of the network, and then, the BOLD time series from fMRI for each ROI is hence constructed. A brain functional network is composed of nodes and edges. Brain connectivity network approaches to understand the brain from functional networks point of view require an anatomical or data-driven functional atlas to create connections between ROIs and to construct a brain functional network for further study. Even though no single functional/anatomical atlas has dominated, an underlying assumption is that such a valid functional atlas exists (Logothetis, 2008; Glover, 2011). Under this framework, the nodes are naturally the individual ROI region itself and the edge weights of the brain network are defined by PCCs between the time series of pair-wise ROIs. The PCC, r_{xy}, of two time series, x and y, is calculated as follows:

$$r_{xy} = \frac{\sum_{b=1}^{s}(x_b - \bar{x})(y_b - \bar{y})}{\sqrt{\sum_{b=1}^{s}(x_b - \bar{x})^2}\sqrt{\sum_{b=1}^{s}(y_b - \bar{y})^2}}, \tag{1}$$

where s is the length of time series, x_b and y_b are the $b^{\text{-th}}$ components of vectors x and y, respectively, $\bar{x}$ and $\bar{y}$ are the means of vectors x and y, respectively. The PCC ranges from -1 to $+1$, where a positive PCC indicates the similarity between the activation patterns of a particular ROI pair, and a negative PCC indicates the dissimilarity between the activation patterns of a particular ROI pair. Figure 1 gives a simplified diagram on how functional network is constructed from fMRI data. The raw fMRI data are first converted into time series of per each ROI, and then, PCCs on ROI pairs are constructed and visualized as a connectivity matrix. Alteration in the brain functional network may lead to biomarker discovery for predicting brain disorders. fMRI may help researchers and clinicians detect brain functional abnormalities that cannot be found with other imaging techniques/modalities, especially when the changes caused by brain disorders such as ASD are minor and no significant structural changes are shown at most of the disease progression. In combination with deep learning models, functional networks constructed from fMRI data are becoming prevalent in many brain studies and even clinical trials.

2.3. Features extraction from a graph-theoretic perspective

To create a matrix representation of a brain functional network as shown in Figure 1, for the simplicity of feature extraction, a 264×264 connectivity matrix directly out of the anatomical parcellation atlas (Powell

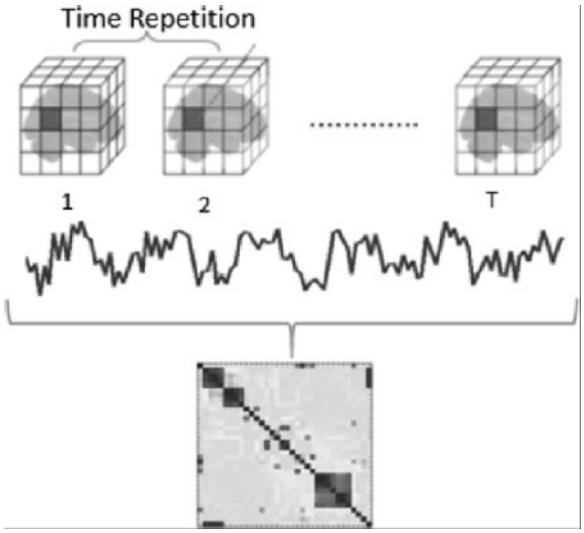

FIG. 1. PCC-based functional connectivity network. PCC, Pearson correlation coefficient.

et al., 2011) was defined for each functional network without further dimension reduction method being used. The elements of the connectivity matrix are the corresponding edge weights denoted by the PCC parameters. Similar to Mostafa et al. (2019) and Mostafa et al. (2020), we apply different threshold $T > 0$ to filter noisy elements in the connectivity matrix, $CM = \left(cm_{i,j}\right)_{n\times n}$, as the fMRI time series are noisy and limited by relatively low resolution compared with structural MRI and easily contaminated by random perturbations such as head movements during the fMRI scan. Each element $cm_{i,j}$ is denoted by the PCC between the *i*th and *j*th ROI. Thus, the adjacency matrix $A = \left(a_{i,j}\right)_{n\times n}$ is constructed out of the connectivity matrix by applying the threshold T to the absolute value of PCC in the connectivity matrix as follows:

$$a_{i,j} = \begin{cases} 1, & \text{if } cm_{i,j} \geq T \\ -1, & \text{if } cm_{i,j} \leq -T \\ 0, & if\ i = j \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

The above thresholding method reduces the random disturbance due to the inherent system noise by fRMI itself. To some extent, this type of threshold operation reduces the raw feature space dimension and also increases the stability of the downstream machine learning algorithms. The graph theory, also called the graph Laplacian, is a matrix representation of a graph, and it can be used to construct many useful properties from a graph theory point of view. One of the many import characteristics is that it can also be used to construct low-dimensional embeddings. This low-dimension embeddings can be very useful for machine learning applications such as dimension reduction in our framework. Here, we consider the spectral techniques to perform dimensionality reduction. This technique relies on the basic assumption that the data lie in a low-dimensional manifold in a high-dimensional space (Belkin, 2003). The Laplacian matrix of an undirected graph $G = (V, E)$ (V denotes the vertex and E denotes the edges) from the adjacency matrix can be calculated as follows:

$$L(G) = D(G) - A(G), \tag{3}$$

where $A(G)$ is the adjacency matrix and $D(G)$ is the degree matrix, $D = \left(d_{i,j}\right)_{n\times n}$ calculated as follows:

$$d_{i,j} = \begin{cases} \sum_{k=1}^{n} a_{i,k}, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases}. \tag{4}$$

From Equation (3), we can see that the Laplacian matrix is the difference between the degree matrix $D(G)$ and the adjacency matrix $A(G)$. In this article, after creating the matrix representation of the brain network and accordingly constructing its Laplacian matrix representation, we use the spectrum of the Laplacian matrix as a portion of raw features to reduce the dimensionality of the raw connectivity matrix

$CM = \left(cm_{i,j}\right)_{n\times n}$, where the spectrum of a Laplacian matrix is its all eigenvalues. An eigenvalue λ of a matrix M can be obtained by solving its following characteristic equation:

$$P(\lambda) = \det(M - \lambda I) = 0, \tag{5}$$

where I is an identity matrix. Other than the spectrum of the Laplacian matrix, topological centralities, that is, assortativity, clustering coefficient, and average degree are also calculated as some other raw features for machine learning algorithms. The assortativity is a preference for a network's node to attach to others that are similar under certain metric. To calculate the assortativity and the clustering coefficient, the adjacency matrix A is first transformed to $\bar{A} = \left(\bar{a}_{i,j}\right)_{n\times n}$ as follows:

$$\bar{a}_{i,j} = \begin{cases} 1, & \text{if } a_{i,j} > 0 \\ 0, & \text{otherwise} \end{cases}. \tag{6}$$

The assortativity is then calculated as shown in Equation (7) (Mijalkov et al., 2017):

$$r = \frac{l^{-1}\sum_{i,j\in L} k_i k_j - \left[l^{-1}\sum_{i,j\in L}\frac{1}{2}\left(k_i + k_j\right)\right]^2}{l^{-1}\sum_{i,j\in L}\frac{1}{2}\left(k_i^2 + k_j^2\right) - \left[l^{-1}\sum_{i,j\in L}\frac{1}{2}\left(k_i + k_j\right)\right]^2}, \tag{7}$$

where k_i and k_j are the respective degrees of nodes i and j, and l is the number of edges in the graph with the adjacent matrix $\bar{A}$. A positive assortativity coefficient indicates that nodes tend to link to other nodes with similar degree/strength (Mijalkov et al., 2017). The assortativity coefficient is analogous to the PCC that assesses the association between two continuous variables. The assortativity coefficient measures the correlation between every pair of nodes that are connected. It is a number between −1 and 1, just very similar with correlation coefficients. A large positive value indicates that connected nodes tend to share similar characteristics; a large negative value means that connected nodes tend to possess very different properties, and a value close to 0 means no strong association between connected nodes.

Clustering coefficient is a measure of the degree that the nodes in a graph tend to cluster together. The clustering coefficient quantifies the abundance of connected triangles in a network and is a major descriptive statistics of networks (Masuda et al., 2018). To calculate the clustering coefficient, at first, we compute the number of triangles associated with each node (denoted by β_G) as follows:

$$\beta_G = \boldsymbol{diag}(\bar{A}\times U(\bar{A})\times\bar{A}), \tag{8}$$

where $\boldsymbol{diag}(X)$ is the operator that takes the diagonal elements of a matrix X, and $U\left(\bar{\mathbf{A}}\right)$ is the upper triangular matrix of $\bar{A}$. The clustering coefficient C is calculated as follows:

$$C = \frac{1}{f}\left(\sum_{i\in V} 2\times\left(\frac{\beta_G(i)}{d_i\times(d_i-1)}\right)\right), \tag{9}$$

where f is the total number of nodes in the network with the adjacent matrix $\bar{A}$ and d_i is the degree of node i.

Average degree is simply the average number of edges per node in the graph. The average degree of a graph $G = (V, E)$ is another measure of how many edges are in set E compared with number of vertices in set V. The average degree (denoted by Q) of a network is calculated directly from the adjacency matrix $\bar{A} = \left(\bar{a}_{i,j}\right)_{n\times n}$ as follows:

$$Q = \frac{2}{f}\times\sum_{i=1}^{f}\sum_{j=1}^{f}\bar{a}_{i,j}, \tag{10}$$

where f is the total number of nodes in the network with the adjacent matrix $\bar{A}$.

2.4. Feature transformation

The feature normalization across samples or data sets is an important step for machine learning algorithms as the raw feature values may have very diverse numerical ranges due to different variations. This wide range of the original features may bias the results. Therefore, it is common to normalize the features and bring them into comparable levels before feeding them to machine learning pipelines. Because the size

of the Laplacian matrix is 264×264 as each row/column represents one ROI, there are 264 eigenvalues for each Laplacian matrix. The eigenvalues are sorted in ascending order for each subject ranging from 0 to $+\infty$. However, when the eigenvalues are normalized by the maximum eigenvalue of each subject, the contribution of each eigenvalue is measured over all the subjects. The contribution of an eigenvalue for a particular subject is ignored in this scenario. Here, in this article, we normalized the eigenvalues for a subject rather than normalizing each eigenvalue over all subjects using the equation as follows:

$$z' = \frac{z - \min(z)}{\max(z) - \min(z)}, \tag{11}$$

where z is an original value and is the normalized value. After normalization, the maximum value and the minimum value in the spectrum of each Laplacian matrix are 1 and 0, respectively. These two features are excluded from the raw feature set as they are now constants. The assortativity, clustering coefficient, and average degree are normalized when calculated by Equations (7), (9), and (10), respectively.

2.5. AE and representation learning

An AE is a type of artificial neural network used to learn hidden representations of the data in an unsupervised manner (Schmidhuber, 2015). The goal of AEs is to minimize the reconstruction error between the input and the output and learn important hidden representation shown in the data. Typically, an AE can learn representations of the data and can be used to reduce the dimensionality of the input data (Baldi, 2012). The latent representation learned by AEs outperforms those handcrafted and predefined features in many applications. Hence, it is a very efficient tool to perform feature construction and dimension reduction, and it helps alleviate the requirement of feature engineering by experts in an unsupervised manner.

Figure 2 shows the architecture of a typical AE. There are three main components of the AE: encoder, latent representation, and decoder. The encoder compresses the input data and learns a latent representation, which is an equivalent but compressed representation of the input data. The decoder aims to reconstruct the input data from the latent representation. There are a total of three hidden layers, one input layer, and one output layer in the proposed AE architecture. In the consecutive layers of the encoder, the number of

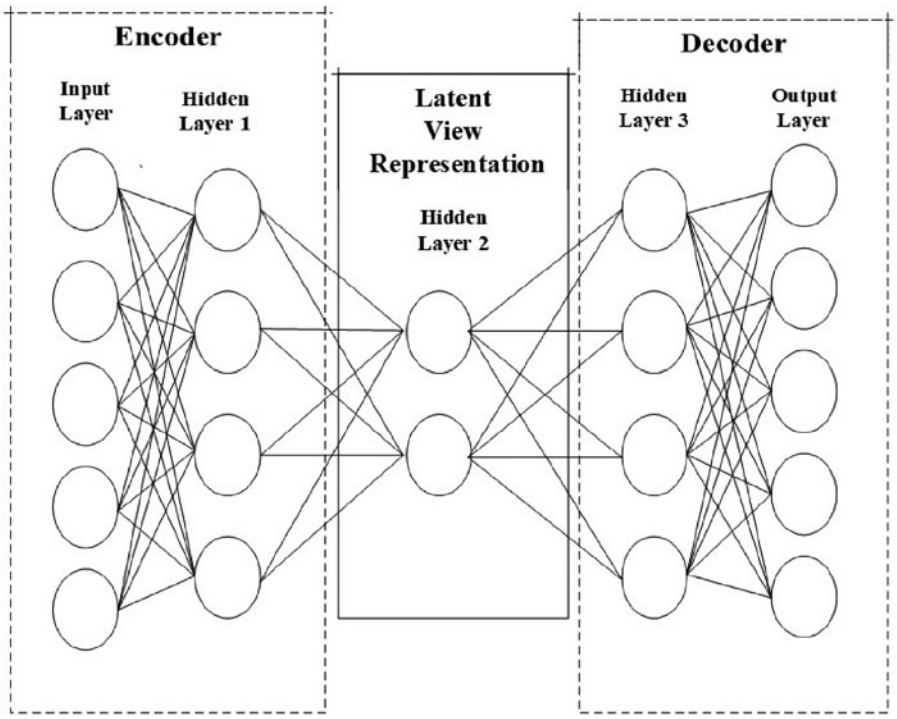

FIG. 2. A simplified architecture of a symmetric autoencoder.

neurons decreases from 267 (input layer) to 200 (hidden layer 1). Then, the information learned by the encoder is projected into the latent space representation through 10 neurons (hidden layer 2). Then, in the decoder, the number of neurons increases from 200 (hidden layer 3) to 267 (output layer) to guarantee the symmetric structure the AE. Hence, the decoder reconstructs the data from the latent representation in a symmetric way. In a word, AEs compress the input into a lower dimension representation and then reconstruct an output from this representation as close to the input as possible. Therefore, the AE is trying to recreate x' from the input data x by minimizing the reconstruction error, $L(x, x')$, where L is the mean squared error between the original input and the reconstructed output in general. Rather than using the AE to replicate the input information to the output, the latent representation can be used as the features to train machine learning algorithms for various tasks. The latent representation learned by AE networks is a high-level abstraction of the input data.

2.6. AE-based feature extraction

When using the AE network as a feature extractor, the latent representation of the data is considered the high-level features. These high-level features are then used as an input of different classifiers for ASD diagnosis (Hosseini-Asl et al., 2016; Guo et al., 2017). In the case of a feature extraction, the AE is very prevalent because it can learn the nonlinear interactions between the raw and the hidden high-level features through the nonlinear activation functions. It is common that transformation, wrapping, and embedding are some of the common feature selection methods. However, these methods can only find the linear relation among the features. The kernel method can find some simple nonlinear relationship of the features, but the learning of the model greatly depends on the kernel itself and the model capacity is small (Han et al., 2018). The efficacy of a machine learning algorithm greatly depends on feature construction and selections. The more discriminant the input features, the better machine learning algorithms can perform on a given data set. Unnecessary and redundant features can make models complicated and easily get overfitted, and reduce the efficacy of machine learning algorithms. An AE-based feature selection scheme works better than the traditional feature selection algorithms in many applications and can abstract the input features automatically (Han et al., 2018).

As AEs in general can discover the high-level latent representation of the input features and compress the feature space, it is very effective for high-dimensional low-sample data sets such fMRI data sets. It is rather expensive to acquire fMRI data under real clinical environment and especially many research institutes and hospitals are under strong regulations for data acquisition and research usage. fMRI data inherently are of very high dimension, but the samples are relatively scarce. AE-based methods naturally can be suitable for this task and can enable data compression and feature selection. It has been used by many researchers for fMRI data feature compression and selection under the frameworks of brain network models (Baldi, 2012; Abraham, 2017). The features selected by AEs are effective and they can be used by traditional machine learning algorithms for better performance as well. Figure 3 gives the proposed AE-based deep learning model for classification for fMRI data. The features constructed in Section 2.3 are compressed by an encoder to extract the abstract latent representations and then fed into a classification network followed right after the encoder. Despite from using the AE as a feature extractor, a neural network-based classifier is developed with an AE. The architecture of our proposed neural network is shown in Figure 3.

In the proposed AE-based DNN classification model, the first two hidden layers are the same as the encoder and latent representation of the AE in Figure 2. The encoder is first trained by the labeled training data, and the network parameters are adapted as part of the classification network. In other words, the pretraining of an AE network is needed to learn the latent features. However, after the latent representation instead of the decoder, we have used two new layers (hidden layer $3'$ and hidden layer $4'$). Finally, the probability of data belonging to a particular class is predicted at the output layer. The output probability is used as the criterion for classifications, and then, a binary decision is made according to the probabilities of classes. At the end, the ASD diagnosis results are given based on the prediction model built.

2.7. Preprocessing pipeline and connectivity models

fMRI has the potential to reveal functional biomarkers of neuropsychiatric disorders, but the data have high dimension and are noisy by itself. However, extracting such biomarkers is extremely difficult for ASD due to complex neuropathologies (Abraham, 2017). Large multisite data sets increase sample sizes to reduce this complexity but comes with the cost of uncontrolled heterogeneity. This heterogeneity raises new challenges in realistic diagnostic scenarios.

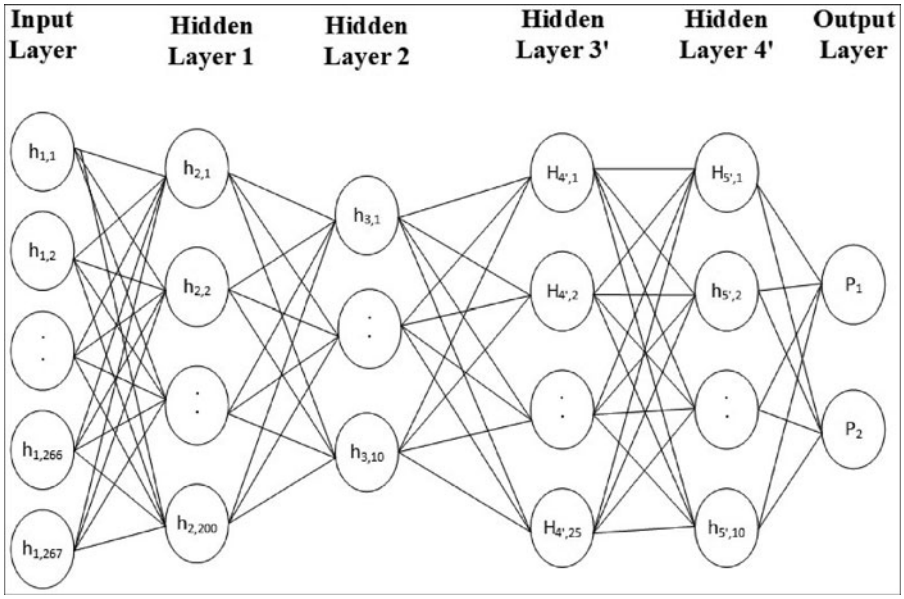

FIG. 3. Proposed autoencoder-based classifier for ASD diagnosis. ASD, autism spectrum disorder.

Different preprocessing and brain functional network construction normally give different outputs and pose a serious challenge to the performance of machine learning algorithms. Different preprocessing and functional connectivity model may lead to significantly different results, and some machine learning algorithms cannot fit well into different preprocessing pipelines and functional network construction methods. In this study, we demonstrate the feasibility of intersite classification of ASD with ABIDE I data set. However, in the previous sections, we did not discuss the impact of the preprocessing and functional network extraction pipeline on the performance of algorithms. In other words, we also need to investigate whether the preprocessing pipelines and functional connectivity construction methods matter and whether different pipelines provide the consistent performance under the proposed AE-based classifiers.

In this section, we summarize some frequently used preprocessing pipelines and connectivity extraction methods (Dadi et al., 2019; Yao et al., 2019). We try to create a comparison strategy to study the impact of different preprocessing methods and functional connectivity model construction methods. The corresponding simulation results are presented in the results as well. Figure 4 gives a comparison flowchart for different preprocessing and network construction methods. In the first step of the pipeline, ROIs are estimated from the training set either through atlas methods such as Harvard Oxford (Desikan et al., 2006) and MODL (massive online dictionary learning) (Mensch et al., 2018) or data-driven methods such as dictionary learning (Mensch et al., 2016), canonical independent component analysis (GroupICA or CanICA) (Varoquaux et al., 2010). The second step consists of extracting time series signals of interest from all the participants, and the extracted time series of all the ROIs are turned into connectivity features via

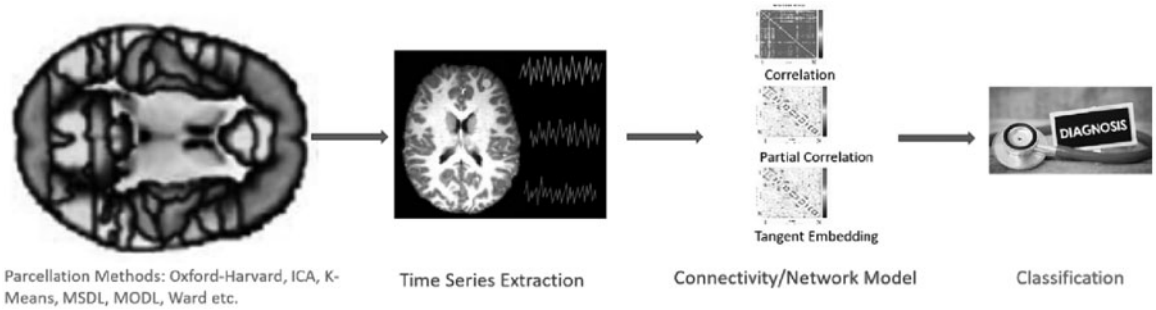

FIG. 4. Different preprocessing and connectivity model for ASD classifications.

covariance estimation at the third step. In combination with other constructed features or transformation, these features are fed into a supervised learning algorithm for classification and yield an accuracy score. As a result, an instance of machine learning pipeline is formed for intersite prediction.

2.8. Comparison and data preparation

In this study, we follow the footsteps of Dadi et al. (2019) and utilize the preprocessing and connectivity/network estimation pipeline. The resulting time series and scripts from Dadi et al. (2019) are used to produce different types of parcellation and covariance estimation (connectivity model) combination for the comparison as stated in Section 2.7.1. The data and scripts in Dadi et al. (2019) are used to prepare the preprocessed data and connectivity matrix, and then, the processed data are utilized in MATLAB for further testing different machine learning pipelines, as shown in Figure 4.

3. RESULTS AND DISCUSSION

In this section, we evaluate the proposed AE-based classifier by comparing to some well-know machine learning algorithms such as SVM and KNN. Then, experiments and discussions on different preprocessing and covariance parameterization methods are presented.

3.1. Experimental setup

We implement an AE-based feature extraction pipeline. As abovementioned, the decoder reconstructs the input data at the output from the latent representation. Therefore, if the reconstruction error is small enough under some measures such as mean square errors, it indicates that the latent variables have learned the salient features of the input data. Thus, the latent variables can produce discriminative and salient representation (advanced features) of the input data. In general, AE network projects the input data from a high-dimension space to a relatively low-dimension latent space and preserve the discriminative features of the original space. These features learnt by the AE network now can be used in machine learning algorithms for the purpose of classification between ASD and healthy subjects. To evaluate the performance of the AE-based feature extractor, the entire data set is divided into two sets of subjects: 80% training data (697 subjects) and 20% testing data (174 subjects). We perform 10-fold cross-validation (CV) by randomly shuffling and splitting each data set into 10-folds. The average accuracy was reported as the overall accuracy on ABIDE data set.

3.2. Comparison study with classic machine learning methods

The AE network is trained by using only the training data split. After completing the training, the AE is used to learn the features from the testing data. Then, the extracted features of the training data are used to train all the machine learning algorithms available in the classification learner toolbox in MATLAB. SVM and KNN with different kernels, and subspace discriminant of the ensemble method normally give better and consistent results consistently. The high-level representations learned by the AE networks are effective and capture the inherent characteristics in this sense as it can improve the overall performance for different machine learning algorithms. Therefore, we include the results of only the aforementioned machine learning algorithms. The illustration of the framework is shown in Figure 5.

The 10-fold CV was conducted with random shuffling to ensure that the results are not biased by the data split itself. Meanwhile, we carry out the experiments by adjusting the threshold value T from 0 to 0.6, with all the edges with positive or negative PCCs. The accuracy and the AUC are the metrics that are used to evaluate the prediction performance and standard deviation to evaluate the stability. The results of the experiments are shown in Table 1.

From Table 1, we can observe that the better classification results are achieved when a threshold is applied to the connectivity matrix. It is concluded that the threshold applied can reduce the noise in the fMRI time series and hence improve the random disturbance caused by fMRI scan. This is a common approach that is adapted by many researchers, and it has been proven to be simple but effective. Both ACC and AUC are the highest for the threshold value of $T = 0.4$. This might be explained by the fact that fMRI data itself are noisy and easily contaminated by head motions. With a threshold being applied to correlation

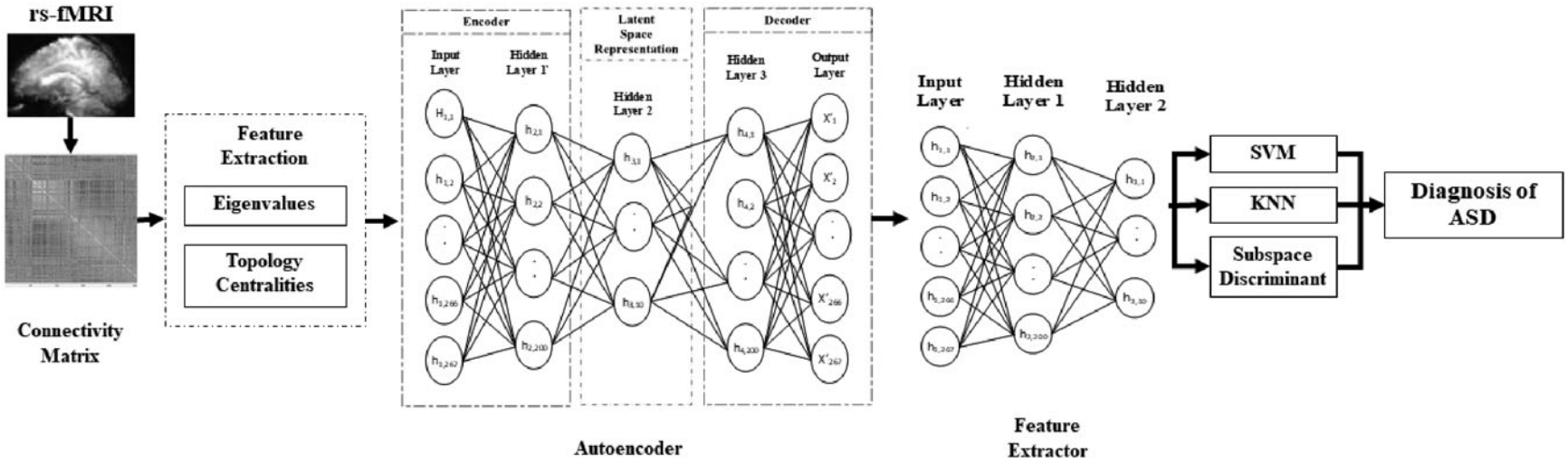

FIG. 5. Illustration of the main steps with autoencoder as feature extractor. KNN, K-nearest neighbor; SVM, support vector machines.

Table 1. Performance Analysis of the Autoencoder-Based Feature Selector

Thresholding condition	*Linear SVM, % (stdv)*		*Medium Gaussian SVM, % (stdv)*		*Coarse Gaussian SVM, % (stdv)*		*Medium KNN, % (stdv)*		*Cosine KNN, % (stdv)*		*Weighted KNN, % (stdv)*		*Subspace discriminant, % (stdv)*	
	ACC	*AUC*	*ACC*	*AUC*	*ACC*	*AUC*	*ACC*	*AUC*	*ACC*	*AUC*	*ACC*	*AUC*	*ACC*	*AUC*
All edges with positive PCC	57.2 (2.0)	57.4 (3.7)	62.2 (1.8)	70.2 (2.1)	54.7 (1.9)	57.5 (1.5)	64.9 (3.2)	70.5 (3.0)	64.8 (1.2)	70.8 (2.0)	68.1 (3.1)	74.8 (2.5)	59.9 (1.4)	58.2 (2.0)
All edges with negative PCC	61.2 (2.1)	65.3 (1.9)	64.6 (1.7)	71.3 (1.2)	59.7 (2.7)	62.0 (2.5)	65.4 (1.3)	70.5 (1.1)	65.8 (3.1)	70.4 (2.0)	65.5 (1.7)	73.1 (1.3)	61.3 (2.4)	65.7 (1.8)
$T=0$	61.3 (3.6)	66.4 (4.0)	67.5 (3.0)	72.2 (1.6)	57.3 (5.4)	62.2 (5.3)	63.6 (2.7)	68.7 (4.6)	63.3 (2.8)	68.1 (4.3)	65.3 (1.7)	71.0 (3.4)	61.0 (3.0)	64.6 (3.3)
$T=0.1$	63.4 (2.1)	67.2 (1.1)	67.4 (2.1)	70.9 (0.9)	59.6 (0.6)	64.7 (0.9)	65.2 (1.6)	71.8 (1.8)	67.2 (2.1)	74.1 (1.0)	65.5 (1.6)	73.4 (1.6)	59.9 (1.5)	65.6 (1.7)
$T=0.2$	66.7 (1.7)	71.0 (1.2)	69.3 (2.2)	73.3 (0.7)	67.6 (3.2)	71.6 (3.0)	66.1 (1.8)	72.9 (1.2)	68.5 (2.5)	74.7 (2.9)	66.6 (2.5)	74.3 (1.3)	68.9 (2.6)	72.2 (1.6)
$T=0.3$	63.8 (1.1)	69.1 (0.7)	64.6 (2.9)	69.5 (2.6)	61.2 (3.8)	63.8 (3.2)	66.2 (3.0)	72.2 (2.6)	65.2 (2.2)	71.3 (1.8)	65.4 (2.9)	72.5 (3.2)	65.2 (2.9)	67.3 (1.6)
$T=0.4$	**68.0 (1.2)**	**71.4 (1.8)**	**72.2 (0.7)**	**78.0 (1.1)**	**66.9 (3.6)**	**70.9 (3.1)**	**72.4 (2.1)**	**77.7 (1.9)**	**72.6 (1.3)**	**79.0 (1.1)**	**72.4 (2.1)**	**77.6 (1.4)**	**69.6 (2.6)**	**72.1 (1.7)**
$T=0.5$	58.9 (5.6)	61.3 (8.0)	69.9 (2.1)	73.1 (1.5)	55.5 (3.5)	61.8 (2.8)	65.3 (1.8)	70.9 (1.4)	64.7 (2.2)	69.1 (2.3)	69.3 (1.7)	77.2 (2.0)	60.1 (2.3)	61.9 (4.8)
$T=0.6$	54.3 (1.6)	55.1 (2.3)	62.9 (2.3)	62.6 (4.6)	56.0 (2.6)	58.1 (1.1)	64.5 (3.2)	69.3 (1.3)	60.9 (3.9)	63.6 (1.4)	63.7 (1.8)	67.9 (1.7)	54.9 (1.6)	57.7 (1.1)

ACC; AUC, receiving operating characteristic curve; KNN, K-nearest neighbor; PCC, Pearson correlation coefficient; stdv, standard deviation; SVM, support vector machines.

parameterization, it can be considered as a high-pass filter to eliminate the minor perturbations caused by different noise sources.

This observation is consistent with the study of Eslami et al. (2019), and the correlations with a threshold in general yield better performance. However, a significant amount of information is lost when applying a larger threshold. It is evident from the results of thresholds $T = 0.5$ and $T = 0.6$. Also, the results are comparatively better when the information of both positive edges (edges with positive PCC) and negative edges (edges with negative PCC) are combined. This is also very interesting and coincides with the instincts. The negative edges also have useful information, and it can be viewed as less connected or possible links between two distinct ROIs. It seems to suggest that it is not wise to directly ignore the negative correlations between ROI time series as some previous work has done. This seems to be different from the assumptions that many other publications used as many researchers directly ignore the negative correlations without considerations.

3.3. Comparison study with DNN methods

To develop the DNN-based classifier, we first pretrain the AE in a similar way as the previous method. Then, we train the DNN as a classifier with the hidden representation from the well-trained AE (as shown in Fig. 3). The first two hidden layers of the deep neural are pretrained with the weights and biases of the first two hidden layers of the AE, and then, a classification network with softmax output layers is connected to the trained first two layers of the AEs. A 10-fold CV is used to evaluate the performance of the proposed DNN classifier. In the training process, the DNN classifier is trained on 784 subjects, and the performance of the model is evaluated on the testing set of 87 subjects. Finally, the average accuracy over all the 10-folds is averaged as the accuracy of the model so as to avoid data bias for a particular experimental run. We have repeated 10 times the process of training the AE and 10-fold CV of the DNN. Each time the subjects are selected randomly, that is, the test and training data split is random. The process is depicted in Figure 6.

To illustrate the effectiveness of pretraining, we also develop a DNN classifier without the pretraining. Other than pretraining, everything else is kept the same for both DNN classifiers. We repeat the 10-fold CV 10 times for both the DNN classifiers. The comparison of the performance of the DNN with and without the pretraining is shown later.

3.3.1. DNN with/without pretrainings. Here, we show the DNN classifier without pretrainings, that is, the AE training parameters are not directly utilized in the overall classification network. This generally gives the worse result than the DNNs with pretrainings due to the limited samples in the fMRI data set, and most of the time overfitting is becoming common. Table 2 gives the performance comparison of both the DNN classifiers.

From Table 2, we can observe that the DNN without pretraining can achieve a classification accuracy of only 76.2% when the threshold is $T = 0.2$. It is a little surprising that the threshold to give the best classification accuracy is not consistent with that ($T = 0.4$) of the traditional machine learning algorithm, as shown in Section 3.2. The accuracy increases to 79.2% when pretraining is applied to the DNN classifier. This classification accuracy improvement is consistent for every thresholding condition. So we can safely conclude that the pretraining is very useful to boost the classification accuracy with litter overhead incurred. From the standard deviation in Table 2, the DNN with pretraining is more stable (aka, the accuracies of all the runs are showing relatively smaller standard variations) and consistent.

In both cases, that is, with/without pretraining, there is a consistent increase of accuracy and AUC after applying some thresholds, rather than using all the PCCs/edges in the brain function network. For the DNN-based classifiers, the accuracy is reaching its maximum at the threshold of $T = 0.2$ as highlighted in bold in Table 2. In conclusion, even without using the pretraining techniques, the AEs still improve the performance over the perception-based DNN classification methods reported in other work and show consistent behavior for different thresholds. The pretraining or even transfer learning may even further improve AE-based methods and reduce the requirements of training samples. The thresholds applied to filter out the random disturbance shown in fMRI turn to be very effective, and there exists an optimal value that can delineate appropriate noise level but not the useful brain activation patterns/activities that BOLD signal can reflect.

3.3.2. Traditional machine learning algorithms with pretrained AE. Both ASD and healthy subjects are used to balance the training data set when training the AE. Even though the latent space representation of AE creates discriminative and salient features of the input data, the difference in the

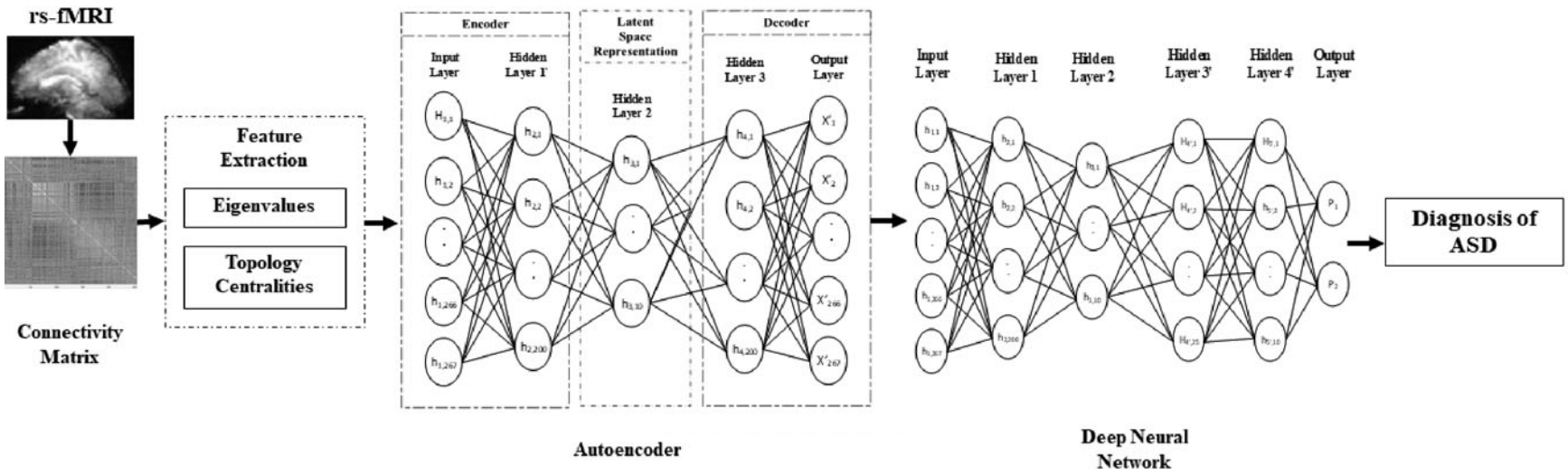

FIG. 6. Illustration of the main steps of neural network-based classifiers.

Table 2. Performance Comparison of the Deep Neural Network Classifier With and Without Pretraining

	ACC, % (stdv)		*AUC, % (stdv)*	
Thresholding condition	*Without pretraining*	*With pretraining*	*Without pretraining*	*With pretraining*
All edges with positive PCC	69.1 (7.4)	74.4 (1.4)	70.9 (10.3)	77.3 (1.3)
All edges with negative PCC	69.5 (0.6)	73.6 (1.1)	72.4 (1.0)	76.9 (0.9)
$T=0$	69.4 (5.2)	73.8 (1.7)	70.8 (7.0)	75.2 (2.1)
$T=0.1$	70.3 (3.4)	74.3 (1.3)	72.1 (4.1)	76.7 (0.9)
$T=0.2$	**76.2 (4.1)**	**79.2 (0.8)**	**79.7 (0.7)**	**82.4 (0.8)**
$T=0.3$	72.9 (1.0)	76.7 (1.1)	76.6 (1.4)	79.8 (1.2)
$T=0.4$	74.6 (1.1)	77.1 (1.8)	77.7 (0.9)	80.9 (1.4)
$T=0.5$	75.5 (1.2)	77.5 (1.1)	80.4 (1.4)	82.2 (0.9)
$T=0.6$	75.8 (1.7)	77.3 (1.5)	80.7 (1.9)	81.7 (3.4)

features between different classes is still not satisfactory (see Table 1 for traditional machine learning algorithms). However, when the DNN is pretrained, the weights and biases of the hidden layers are accordingly updated to classify ASD and healthy subjects. As a result of the pretraining techniques, the weights and biases of the encoder and the deduced latent space representation are also updated to create an improved discriminative representation of the data. After completing the training of the DNN, the first two hidden layers can be used to extract a more discriminative representation of features and then adapted as the input for other machine learning algorithms. To enhance the performance of the feature extraction at first, we train the AE from scratch until it converges. Then, we train the DNN with pretraining by adapting the AE weights to the DNN classification network. After completing the training of the DNN, we use the first two hidden layers to extract features and train different shallow machine learning algorithms such as SVM and KNN.

Table 3 shows the performance of traditional machine learning algorithms with the features adapted from a pretrained AE. Comparing Tables 1 and 3, we can observe that there is an increase in performance due to pretraining. The highest accuracy of 74.6% is achieved using the KNN classifier with the cosine kernel and the threshold value of $T=0.2$, and the results are highlighted in bold in Table 3. In summary, the AE network and pretraining methods are indeed capturing the inherent characteristic of the raw connectivity model derived from fMRI time series. The latent representation learned by the pretrained AE can help boost the traditional machine learning algorithms. We observed the similar performance of the thresholding method, and it is necessary to involve such a technique to filter out the random disturbance caused by some uncontrolled noise generated in fMRI scanning.

3.3.3. Comparison with some of the previous work in the literature. In this section, we compare the results with some of previous state-of-the-art studies on ABIDE I data set using fMRI data. Table 4 shows a comparison of the proposed study with other state-of-the-art methods. We have compared our study with only those studies, where the entire ABIDE 1 data set is used. From our perspective, this type of study may have potential to find some real-world applications as the whole data set mimics the clinical settings for different locations. As shown in Table 4, our proposed AE-based classifiers with pretraining outperform all the previous studies. The comparison results indicate that our graph-theoretic approach-based feature extraction methods may have grasped the inherent properties of the connectivity-based networks and may have some reference meaning for biomarker discovery. The spectrum of Laplacian matrix and other graph theory enabled features can reduce the dimensionality of the original high-dimension connectivity model, and both the traditional machine learning algorithms and DNNs-based methods all have better performance by resorting to these newly constructed features.

3.3.4. Results with different atlases, connectivity network, and classifiers. We use the preprocessing time series from Dadi et al. (2019) to benchmark the proposed AE-based DNN classifier and find whether the proposed feature extraction and deep learning algorithms are sensitive to different parameterization combinations. The findings are generally well aligning with the reference (Dadi et al., 2019) except some minor difference. The comparison configurations are described in Section 2.7. Here, we only report the findings in the sequel.

Table 3. Performance Analysis of Features Extracted from the Combination of the Pretrained Autoencoder and the Traditional Machine Learning Algorithms

Thresholding condition	*Linear SVM, % (stdv)*		*Medium Gaussian SVM, % (stdv)*		*Coarse Gaussian SVM, % (stdv)*		*Medium KNN, % (stdv)*		*Cosine KNN, % (stdv)*		*Weighted KNN, % (stdv)*		*Subspace discriminant, % (stdv)*	
	ACC	*AUC*	*ACC*	*AUC*	*ACC*	*AUC*	*ACC*	*AUC*	*ACC*	*AUC*	*ACC*	*AUC*	*ACC*	*AUC*
All edges with positive PCC	62.8 (5.5)	66.1 (8.0)	66.3 (4.1)	70.1 (5.5)	55.3 (2.5)	60.8 (4.3)	67.8 (2.5)	71.4 (2.7)	66.1 (2.9)	69.9 (3.1)	67.7 (1.3)	72.4 (1.2)	60.7 (4.3)	63.3 (6.2)
All edges with negative PCC	66.0 (2.1)	71.4 (3.6)	66.6 (1.4)	74.2 (2.1)	63.7 (4.8)	68.1 (4.8)	66.8 (2.6)	72.4 (1.5)	66.7 (3.6)	72.3 (2.9)	67.2 (2.2)	73.8 (2.3)	66.6 (2.7)	70.9 (3.7)
$T = 0$	67.2 (4.4)	71.8 (4.8)	68.3 (2.1)	71.9 (2.2)	64.4 (6.7)	68.3 (7.0)	67.0 (2.6)	70.5 (3.1)	66.1 (3.4)	70.5 (2.5)	65.6 (2.9)	70.3 (3.0)	64.9 (4.1)	68.9 (5.3)
$T = 0.1$	67.0 (3.2)	71.2 (2.9)	69.1 (2.0)	74.7 (2.4)	66.6 (3.7)	71.2 (2.8)	68.4 (2.6)	74.5 (2.3)	67.9 (2.8)	76.0 (2.2)	67.9 (3.9)	75.1 (3.0)	67.1 (2.3)	71.1 (2.5)
$T = 0.2$	**72.8 (2.9)**	**76.3 (3.4)**	**73.7 (1.6)**	**76.7 (1.6)**	**73.0 (2.5)**	**77.9 (1.4)**	**73.2 (1.7)**	**77.7 (2.1)**	**74.6 (1.9)**	**78.7 (2.1)**	**72.7 (2.2)**	**77.2 (2.0)**	**74.4 (1.1)**	**77.8 (1.5)**
$T = 0.3$	66.1 (3.3)	71.7 (3.3)	66.4 (2.9)	71.9 (2.0)	63.0 (6.1)	67.7 (5.5)	67.1 (2.6)	72.8 (2.6)	66.9 (3.6)	72.1 (2.8)	66.4 (3.3)	72.9 (3.4)	67.8 (1.7)	71.6 (3.8)
$T = 0.4$	71.1 (4.7)	75.9 (3.8)	73.4 (1.1)	78.8 (1.0)	67.3 (3.1)	73.4 (3.1)	73.2 (2.5)	78.2 (1.8)	73.5 (1.2)	78.8 (1.1)	73.6 (1.5)	78.5 (1.5)	69.8 (3.8)	75.6 (2.8)
$T = 0.5$	64.4 (7.7)	69.9 (7.0)	70.3 (1.3)	73.1 (1.7)	60.2 (6.9)	64.9 (6.3)	67.7 (3.1)	72.5 (3.2)	67.2 (3.7)	72.8 (3.6)	70.6 (3.6)	76.7 (3.0)	64.0 (7.5)	68.5 (7.8)
$T = 0.6$	63.8 (3.1)	64.4 (3.2)	65.2 (3.8)	71.2 (3.0)	61.6 (2.8)	59.0 (1.9)	63.7 (1.8)	70.5 (1.5)	64.7 (1.9)	70.0 (2.2)	64.9 (2.6)	71.1 (2.5)	60.4 (3.0)	60.1 (1.9)

Table 4. Comparison of Accuracy of Proposed Method and the State-of-the-Art Classification Methods

Methods	*Accuracy (%)*
Heinsfeld et al. (2017)	70.0
Eslami et al. (2019)	70.1
Wong et al. (2018)	71.1
Sakib et al. (2019)	77.7
Xing et al. (2019)	66.8
Proposed autoencoder-based feature extractor	74.6
Proposed autoencoder-based DNN classifier	**79.2**

DNN, deep neural network.

3.3.4.1. Choice of atlas of the parcellation

The choice of atlas normally has a great impact on the prediction accuracy. There are two distinct types of atlas construction methods: anatomical and data driven (Dadi et al., 2019). We used the same atlas categories as stated in Dadi et al. (2019), and the data are publicly available at http://preprocessed-connectomes-project.github.io/abide and http://team.inria.fr/parietal/files/2015/07/MSDL_ ABIDE.zip web sites. During the experiments, we only vary the atlases and keep PCCs as the default connectivity measures and DNN as the classifiers. The results appear to suggest that the multi-subject dictionary learning (MSDL) atlas shows the robust performance out of all the anatomical and data-driven atlas approaches, but it does not show dramatic performance gain. This observation is a bit off from what has been reported in Dadi et al. (2019) as MSDL clearly outperforms many other atlases. This seems to suggest that the atlas may be insensitive for our proposed graph-theoretic-based network construction approach. Data-driven and anatomical atlases are acceptable and can effectively parcellate the brain regions under our settings.

3.3.4.2. Choice of connectivity network parameterization

Like Section 3.3.4.1 with all other configuration fixed, we compare the performance of full correlations, partial correlations, and tangent space parameterization for connectivity network construction. Again, this reports similar pattern as in Dadi et al. (2019). Tangent space parameterization tends to outperform full correlations or partial correlations. Functional connectivity matrices built with tangent space parameterization have slightly better performance than full and partial correlation measures. While partial correlation approaches are currently the dominant approach for connectivity matrix estimation in the literature, our simulation results corroborate and resonate with the findings in Dadi et al. (2019) that tangent space parameterization might be another alternatives and may have some potentials from geometric point of view.

3.3.4.3. Choice of classifier

As shown in Tables 1–3, in Section 3.2, we can clearly observe that even with the same feature compression mechanism learnt by AEs, deep models outperform all the shallow models such as SVM, KNNs, and its variants. Especially the deep models have an enormous capacity and it can still have the potential to improve with a larger training data set. The fMRI data sets for ASD are gradually increasing and we expect to DNN-based classifiers evolve with this data acquisition trend. The performance gap between classic shallow models and deep learning models could be even enlarged in the future. Our results seem to lean toward deep learning methods despite its disadvantage that it is hard to explain the biomarkers and data-hungry nature.

3.4. Discussion

3.4.1. Data acquisition heterogeneity. A lot of studies have devoted to using predictive models on network connectivity for brain disorder diagnosis. Even the fundamental steps of a machine learning pipeline are universal as shown in Figure 4. A challenge to such standard approach is the heterogeneity of data acquisition protocols, prediction settings, and other experimental variations such as different scanner configurations and clinical questionnaires. We have seen many excellent machine learning algorithms for intersite data (Khosla et al., 2019). However, many of these models do not generalize well into intrasite data sets while this seems to be the typical clinical settings. This type of heterogeneity hinders many researches from being applied to clinical practices. Hence, we should always consider the real application scenarios when we design a machine learning algorithm for fMRI data classifications.

3.4.2. Feature construction and model selection. We systematically compare frequently used functional network-based classification methods. Even though the connectivity-based fMRI classification seems to be straightforward, different feature construction methods such as parcellation atlases and covariance estimation have relatively significant impact on the predicative model performance. Unfortunately, there is no standard protocol for fMRI data preprocessing and connectivity model construction. This has led to diverse preprocessing pipelines and renders the task of model performance comparison very difficult for ASD diagnosis. Researchers tend to select a particular combination to report the results, and this sometimes biases the realistic potential of a model, which makes it difficult to benchmark different algorithms. Clearly, more systematic benchmarking is needed to fairly evaluate all the models.

We also find that classic models such as SVM appear as a good default choice of classifiers, and they always provide a good starting point to tackle fMRI classification problems. SVM seems to be the go-to classifier for many. The tangent space parameterization is less utilized probably due to its mathematical complexity. Even there exist no systematic benchmarking methods, our simulation seems to suggest that deep learning-based models in general are more promising compared with shallow models in many of the parameter combinations. The graph theory enabled feature construction, aka, the spectrum of Laplacian matrix and other graph properties shows great potential in this classification task, and it can to some extent reduces the dimensionality and captures the inherent nature of the connectivity model from a network point of view. As we can see, this type of feature construction is fairly simple but very effective.

4. CONCLUSIONS

In this study, we have presented the deep learning-based ASD diagnosis methods with the features extracted from functional brain networks based on graph theory and AEs (Mostafa et al., 2019; Mostafa et al., 2020). In the study, a DNN was pretrained with an AE, and we adapted the latent representations learnt by the AE to traditional classification algorithms for benchmarking and transferred the weights to a DNN-based classifier for improved performance. The performance improvement is consistent for both deep models and traditional machine learning algorithms. It has shown that the classification accuracy of the DNN-based classifier has increased due to the pretraining and outperformed all the classic predictive models. Our proposed diagnosis method has achieved a classification accuracy of 79.2%, which is better than the state-of-the-art methods on the ABIDE 1 data set. An AE-based feature selection method proposed for the diagnosis of ASD has been proposed for improved performance. In this method, we have demonstrated that the learning of the DNN classifier can be incorporated with the AE for the dimensionality reduction. Traditional machine learning classifiers were also implemented to evaluate the performance of the AE-based feature selection method and achieved a classification accuracy of 74.6%. Our proposed feature construction methods and deep learning model can improve the diagnosis of ASD more accurately and precisely than the state-of-the-art methods. Finally, we applied our pipeline to different data preprocessing and covariance estimation configurations and discussed the comparison results.

AUTHOR DISCLOSURE STATEMENT

The authors declare they have no competing financial interests.

FUNDING INFORMATION

This work is supported by the Natural Science and Engineering Research Council of Canada (NSERC).

REFERENCES

Abraham, A. 2017. Deriving reproducible biomarkers from multisite resting-state data: An autism-based example. *Neuroimage* 147, 736–745.

Arbabshirani, M.R., Plis, S., Sui, J., et al. 2017. Single subject prediction of brain disorders in neuroimaging: Promises and pitfalls. *Neuroimage* 145, 137–165.

Baldi, P. 2012. Autoencoders, unsupervised learning, and deep architectures, 37–50. *In* Guyon, I., Dror, G., Lemaire, V., Taylor, G.W., and Silver, D.L., eds. *ICML Unsupervised and Transfer Learning*, vol. 27 of JMLR Proceedings. JMLR.org

Belkin, M. 2003. Problems of Learning on Manifolds (PhD Thesis). Department of Mathematics, The University of Chicago.

Cox, R.W. 1996. AFNI: software for analysis and visualization of functional magnetic resonance neuroimages. *Comput. Biomed. Res.* 29, 162–73.

Dadi, K., Rahim, M., Abraham, A., et al. 2019. Benchmarking functional connectome-based predictive models for resting-state fMRI. *Neuroimage* 192, 115–134.

Desikan, R.S., Ségonne, F., Fischl, B., et al. 2006. An automated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based regions of interest. *Neuroimage* 31, 968.

Du, Y., Fu, Z., and Calhoun, V.D. 2018. Classification and prediction of brain disorders using functional connectivity: Promising but challenging. *Front. Neurosci.* 12, 525.

Eslami, T., Mirjalili, V., Fong, A., et al. 2019. ASD-DiagNet: A hybrid learning approach for detection of autism spectrum disorder using fMRI data. *Front. Neuroinform.* 13, 70.

Glover, G.H. 2011. Overview of functional magnetic resonance imaging. *Neurosurg. Clin. North Am.* 22, 133–139.

Guo, H., Yin, W., Mostafa, S., et al. 2020. Diagnosis of ASD from rs-fMRIs based on brain dynamic networks, pgs. 166–177. In: Cai, Z., Mandoiu, I., Narasimhan, G., Skums, P., and Guo, X. (eds): Bioinformatics Research and Applications. ISBRA 2020. Lecture Notes in Computer Science, vol. 12304. Springer, Cham.

Guo, X., Dominick, K.C., Minai, A.A., et al. 2017. Diagnosing autism spectrum disorder from brain resting-state functional connectivity patterns using a deep neural network with a novel feature selection method. *Front. Neurosci.* 11, 460.

Han, K., Li, C., and Shi, X. 2018. Autoencoder Feature Selector. arXiv preprint arXiv:1710.08310 v1.

Heinsfeld, A.S., Franco, A.R., Craddock, R.C., et al. 2017. Identification of autism spectrum disorder using deep learning and the ABIDE dataset. *Neuroimage Clin.* 17, 16–23.

Hirvikoski, T., Mittendorfer-Rutz, E., Boman, M., et al. 2016. Premature mortality in autism spectrum disorder. *Br. J. Psychiatry* 208, 232–238.

Hosseini-Asl, E., Keynto, R., and El-Baz, A. 2016. Alzheimer's disease diagnostics by adaptation of 3D convolutional network, 126–130. *In 2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, USA.

Jenkinson, M., Beckmann, C.F., Behrens, T.E.J., et al. 2012. FSL. *Neuroimage* 62, 782–790.

Khosla, M., Jamison, K., Ngo, G.H., et al. 2019. Machine learning in resting-state fMRI analysis. *Magn. Reson. Imaging* 64, 101–121. Artificial Intelligence in MRI.

Kong, Y., Gao, J., Xu, Y., et al. 2019. Classification of autism spectrum disorder by combining brain connectivity and deep neural network classifier. *Neurocomputing* 324, 63–68.

Logothetis, N.K. 2008. What we can do and what we cannot do with fMRI. *Nature* 453, 869–878.

Lord, C., Rutter, M., Goode, S., et al. 1989. Autism diagnostic observation schedule: A standardized observation of communicative and social behavior. *J. Autism Dev. Disord.* 19, 185–212.

Lord, C., Rutter, M., Le Couteur, A., et al. 1994. Autism Diagnostic Interview-Revised: A revised version of a diagnostic interview for caregivers of individuals with possible pervasive developmental disorders. *J. Autism Dev. Disord.* 24, 659–685.

Lundervold, A.S. 2019. An overview of deep learning in medical imaging focusing on MRI. *Z. Med. Phys.* 29, 102–127.

Martino, A. 2019. Autism Brain Imaging Data Exchange I ABIDE I. Available at: http://fcon_1000.projects.nitrc.org/indi/abide/abide_I.html, last Accessed May 24, 2019.

Martino, A.D., Yan, C.-G., and Milham, M.P. 2014. The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism. *Mol. Psychiatry* 19, 659–667.

Masuda, N., Sakaki, M., Ezaki, T., et al. 2018. Clustering coefficients for correlation networks. Front. Neuroinform. DOI: 10.3389/fninf.2018.00007.

Mensch, A., Mairal, J., Thirion, B., et al. 2018. Stochastic subsampling for factorizing huge matrices. *IEEE Trans. Signal. Process.* 66, 113–128.

Mensch, A., Varoquaux, G., and Thirion, B. 2016. Compressed Online Dictionary Learning for Fast. Resting-State fMRI Decomposition, pgs. 1282–1285. In IEEE 13th International Symposium on Biomedical Imaging (ISBI). Prague, Czech Republic.

Meszlenyi, R.J., Hermann, P., Buza, K., et al. 2017. Resting state fMRI functional connectivity analysis using dynamic time warping. *Front. Neurosci.* 11, 75.

Mijalkov, M., Kakaei, E., Pereira, J.B., et al. 2017. BRAPH: A graph theory software for the analysis of brain connectivity. *PLoS One* 12, 0178798.

Mostafa, S, Tang, L., and Wu, F.X. 2019. Diagnosis of autism spectrum disorder based on Eigenvalues of brain networks. *IEEE Access* 7, 128474–128486.

Mostafa, S., Yin, W., and Wu, FX. 2020. Autoencoder based methods for diagnosis of autism spectrum disorder, 39–51. *In* Măndoiu, I., Murali, T., Narasimhan, G., Rajasekaran, S., Skums, P., and Zelikovsky, A., eds. *Computational Advances in Bio and Medical Sciences. ICCABS 2019. Lecture Notes in Computer Science*, vol. 12029. Springer, Cham.

Murphy, K.P. 2013. *Machine Learning: A Probabilistic Perspective.* MIT Press, Cambridge, Mass. [u.a.]. Available at: https://www.amazon.com/Machine-Learning-Probabilistic-Perspective-Computation/dp/0262018020/ref=sr_1_2?ie=UTF8&qid=1336857747&sr=8-2. Last accessed on May 25, 2020.

Poldrack, R.A., Baker CI, Durnez J, et al. 2017. Scanning the horizon: towards transparent and reproducible neuroimaging research. *Nat. Rev. Neurosci.* 18, 115–126.

Power, J.D., Cohen, A.L., Nelson, S.M., et al. 2011. Functional network organization of the human brain. *Neuron* 72, 665–678.

Russell, S., and Norvig, P. 2020. *Artificial Intelligence: A Modern Approach,* 4th ed. Pearson. ISBN 978-0134610993.

Schmidhuber, J. 2015. Deep learning in neural networks: An overview. *Neural Netw.* 61, 85–117.

Shen, D., Wu, G., and Suk, H.-I. 2017. Deep learning in medical image analysis. *Annu. Rev. Biomed. Eng.* 19, 221–248.

Varoquaux, G., Sadaghiani, S., Pinel, P., et al. 2010. A group model for stable multi-subject ICA on fMRI datasets. *Neuroimage* 51, 288–299.

Watanabe, T., and Rees, G. 2017. Brain network dynamics in high-functioning individuals with autism. *Nat. Commun.* 8, 16048.

Wong, E., Anderson, J.S., Zielinski, B.A., et al. 2018. *Riemannian Regression and Classification Models of Brain Networks Applied to Autism,* 78–87. Springer, Cham.

Xing, X., Ji, J., and Yao, Y. 2019. Convolutional neural network with element-wise filters to extract hierarchical topological features for brain networks, 780–783. *In* 2018 IEEE International Conference on Bioinformatics and Biomedicine. BIBM 2018. IEEE, Madrid.

Yahata, N., Morimoto, J., Hashimoto, R., et al. 2016. A small number of abnormal brain connections predicts adult autism spectrum disorder. *Nat. Commun.* 7, 11254.

Yao, D., Liu, M., Wang, M., et al. 2019. Triplet graph convolutional network for multi-scale analysis of functional connectivity using functional MRI. *In* Zhang, D., Zhou, L., Jie, B., and Liu, M., eds. *Graph Learning in Medical Imaging. GLMI 2019. Lecture Notes in Computer Science,* vol. 11849. Springer, Cham.

Yin, W., Li, L., and Wu, F.X. 2020. Deep learning for brain disorder diagnosis based on fMRI images. Neurocomputing, in press.

Zhu, G., Jiang, B., Tong, L., et al. 2019. Applications of deep learning to neuro-imaging techniques. *Front. Neurol.* 10, 869.

Address correspondence to:
Prof. Fang-Xiang Wu
Division of Biomedical Engineering
Department of Mechanical Engineering
Department of Computer Science
University of Saskatchewan
Saskatoon
Canada

E-mail: faw341@mail.usask.ca