Recognition of ASD Based on fMRI Data Using Two Hybrid Learning Approaches

Sara Soltani Gerdefaramarzi , Hamid Reza Hakim Davoodi

Abstract

Diagnosing mental disorders such as Autism Spectrum Disorder (ASD) is very challenging. Given that the diagnosis of this condition by psychiatrists relies on multiple behavioral symptoms, many of which overlap with other mental disorders, there is a high likelihood of misdiagnosing patients with autism as healthy individuals. In this regard, it is necessary to provide a reliable method for diagnosing this mental disorder by leveraging advanced machine learning and deep learning techniques. In this paper, two hybrid methods of machine learning and deep learning have been designed and implemented. The first method combines an autoencoder and a deep neural network to extract beneficial features and properly tune the model's Hyperparameters. The second method combines a support vector machine model with a data augmentation strategy using simultaneous training of a generative adversarial network and an autoencoder to prevent model overfitting. It is worth mentioning that in the two proposed methods, the Extra-Trees algorithm is initially used for dimensionality reduction and feature selection. These two methods have been implemented on the entire publicly available dataset released by the Autism Brain Imaging Data Exchange, and the results of both have been compared with each other. Our study in this paper achieved a maximum accuracy of 75.6%, sensitivity of 70.8%, and specificity of 79.7%, surpassing many recent studies published in this field.

Keywords: fMRI, ASD, Autoencoder, Deep Neural Network, Generative Adversarial Network, Extra-Trees, Classification, Data Augmentation

Introduction

Diagnosing mental disorders such as Autism Spectrum Disorder (ASD) is very challenging. Given that the diagnosis of this condition by psychiatrists relies on multiple behavioral symptoms, many of which overlap with other mental disorders, there is a high likelihood of misdiagnosing patients with autism as healthy individuals. As indicated in [1], premature death is very common among patients with autism. Since there is no guaranteed medication for the complete recovery of individuals with autism, early diagnosis of this condition and subsequent medical interventions can slow the progression of the disease. In this regard, it is necessary to provide a reliable method for diagnosing this mental disorder. In this regard, numerous studies such as [2] have been conducted on diagnosing this condition using distinguishing behavioral data such as eye movements and facial expressions, or demographic information of individuals, including age, IQ, and gender.

In this paper, our goal is to classify patients with autism from healthy control subjects using fMRI data from the whole data repository ABIDE 1. Two hybrid methods of machine learning and deep learning have been designed and implemented. The first method combines an autoencoder and a deep neural network to ensure that advantageous features are extracted and that the model's parameters are properly tuned. The second method combines a support vector machine model for classification with a data augmentation strategy using simultaneous training of a generative adversarial network and an autoencoder to prevent model overfitting. It is worth mentioning that in the two proposed methods, the Extra-Trees algorithm is initially used for dimensionality reduction and feature selection.

Recently, the diagnosis of autism spectrum disorder (ASD) using fMRI data, particularly the ABIDE dataset, has garnered significant attention, and numerous studies have been conducted in this area. For instance, [3] utilized two subsets of fMRI data, where the first subset includes 59 high-functioning males with ASD and 59 age and IQ-matched typically developing (TD) males, and the other contains an additional age and IQ-matched cohort of 178 individuals (89 ASD; 89 TD) from the Autism Brain Imaging Data Exchange (ABIDE) and achieved 76.67% accuracy finally. In another study, [4] implemented a model using structural texture and functional connectivity features obtained from 3-dimensional structural magnetic resonance imaging (MRI) and 4-dimensional resting-state functional magnetic resonance imaging (fMRI) scans of 1111 total healthy and ASD subjects and gained 64.3% accuracy on ABIDE data. Additionally, the DNN model proposed in [5] achieved an accuracy of 90.39% on 182 subjects. The approach proposed in [6] got 60% accuracy using the combination of functional connectivity between ROIs and demographic information such as age, age-squared, gender, handedness, and site through 964 subjects. Furthermore, [7] obtained 68.5% accuracy using the recurrent neural networks with long short-term memory (LSTMs) for classification through the entire ABIDE 1 dataset.

It is worth mentioning that only a few recent studies and articles, such as [8], [9], [10], and [11], have targeted the entire dataset. In other words, many studies have either considered a subset of the dataset or incorporated additional information, such as demographic data, into their models. Since we have considered the entire dataset in this paper, we have used the four proposed models in [8], [9], [10], and [11] as baselines to evaluate our models.

The structure of this paper is as follows: First, we provide a brief overview of the dataset used, the selection of useful features, and finally the classification using the two proposed hybrid models. Then, we discuss the experiments conducted and the results obtained from the models. In the final section, we present the conclusions of the paper and offer suggestions for future work.

ABIDE data set and preprocessing

In this study, the proposed approaches were applied to the entire ABIDE 1 dataset. This dataset includes fMRI data and demographic information such as gender, age, and IQ of individuals with autism and healthy controls, collected from 17 different centers. The dataset comprises 1112 samples, with 539 samples from individuals with autism and 573 healthy controls. It is noteworthy that each center used different parameters and protocols for scanning the data. Parameters such as repetition time (TR), echo time (TE), and the number of voxels vary between centers. Therefore, necessary preprocessing steps must be carried out on this dataset to prepare it for applying the models. The preprocessing was performed using the Configurable Pipeline for the Analysis of Connectomes (CPAC) [12]. The preprocessing steps include slice timing correction, motion correction, removal of nuisance signals, correction of low-frequency drifts, voxel intensity normalization, band-pass filtering (0.01~0.1 Hz), and spatially normalization.

To more accurately evaluate our proposed approaches, we compare our results with those published in recent papers, such as studies by [8], [9], [10], and [11] which share the same objective as our study and have been conducted under the same conditions and using the same data. Accordingly, we apply our models to all 871 samples in the dataset, which include 403 ASD subjects and 468 typical control subjects.

Feature Extraction

In fMRI data, functional connectivity analysis is of great importance, as it measures how different brain regions interact with each other, which can provide distinguishing information to differentiate between individuals with autism and healthy individuals. In this regard, each brain volume needs to be divided into a specified number of regions. In this paper, the CC400 [13] atlas is used to divide each brain volume into 392 regions of interest (ROIs). These regions are functionally homogeneous, meaning that the voxels within each region are functionally similar to each other. Then, for functional connectivity analysis, Pearson's correlation is used to examine the relationships between the time series of different regions [14]. Given two time series u and v, and their respective means $\bar{u}\ and\ \bar{v}$, the Pearson correlation coefficient between these two time series is calculated as follows:

$$\rho_{uv} = \frac{\Sigma(u_t - \bar{u})(v_t - \bar{v})}{\sqrt{\Sigma(u_t - \bar{u})^2}\sqrt{\Sigma(v_t - \bar{v})^2}}$$

Ultimately, we will have a correlation matrix with dimensions 392 * 392. However, given the symmetric nature of this matrix, only the upper triangular part is extracted and used. The remaining values, which amount to 392 * 391 / 2 = 76636 , are considered as the feature vector for each sample. However, the number of features is very high and negatively impacts the performance of the models. Therefore, dimensionality reduction is necessary. In this context, the Extra-Trees algorithm [15] is used. This algorithm, like Random Forests, constructs multiple decision trees during training but introduces a higher level of randomness by considering random thresholds for each feature, rather than searching for the best possible thresholds. Feature importance is calculated based on the average impurity reduction computed from all the trees built. For each tree, impurity reduction (Gini importance) for each feature is computed. Then, the Gini importance for each feature is averaged across all trees to obtain the final feature importance score. The higher the score, the more effective the feature is for classifying the data.

Finally, we selected the 1935 initial features identified by the algorithm as useful and with high scores as the final features for our dataset as mentioned in [16].

First Model: A Combination of an Autoencoder and a Deep Neural Network

The proposed model is a combination of an autoencoder and a deep neural network, implemented using the TensorFlow deep learning framework. The model works as follows: the autoencoder maps the input space with 1935 features to a space with 128 features, and on the other side, a decoder reconstructs the mapped data back to the original space with 1935 features. To train the autoencoder, the Mean Squared Error (MSE) loss function is used to calculate the reconstruction error of the data, which is defined as follows:

$$\text{MSE loss} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2$$

Additionally, a deep neural network is designed to receive the 128 features output from the encoder as input. This deep neural network then maps the 128 features to a space with 64 features through a hidden layer. In the final output layer, the probability of the data belonging to class one is determined. A threshold of 0.5 is considered, where if the probability is greater than 0.5, the data belongs to class one, and if it is less, it belongs to class zero. This relationship is defined as follows:

$$\hat{y} = \begin{cases} 1, & \text{if } f(x) \geq 0.5 \\ 0, & \text{otherwise} \end{cases}$$

To train the deep neural network, the binary cross-entropy loss function is used. This function utilizes the true labels of the data, denoted as y, and the output of the neural network, which represents the probability of each data point belonging to the class of individuals with autism, denoted as f(x):

$$BCE\ loss = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log\left(f(x_i)\right) + (1-y_i)\log\left(1-f(x_i)\right)\right]$$

The key aspect of this model, which leads to its good results, is the simultaneous training of the autoencoder and the deep neural network. This model is trained over 500 epochs, and at each epoch, the reconstruction error from the decoder (MSE) is summed with the classification error from the neural network (BCE) and used to update the parameters. This simultaneous training ensures that the extracted features are optimal for both the reconstruction of the decoder and for possessing distinguishing information for more accurate data classification. The diagram of the first proposed model is shown in figure 1:

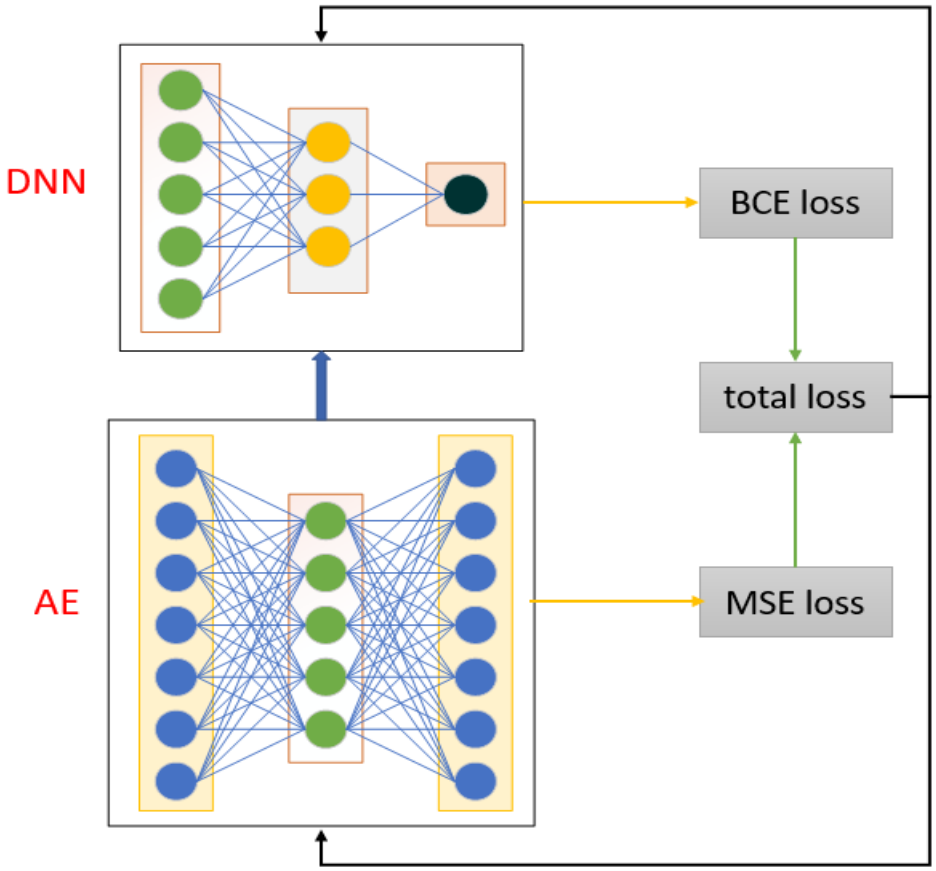

Figure 1| The structure of the proposed first model consists of an autoencoder and a deep neural network with a single hidden layer, which are trained simultaneously.

Second Model: The Use of the SVM Model alongside Data Augmentation through Generative Adversarial Network Architecture and Autoencoder

In this proposed method, according to [16] the SVM model is used for classifying samples, and the model Hyperparameters such as kernel type, regularization constant (C), and kernel coefficient (γ) are first tuned using the grid search method.

When working with machine learning or deep learning models, it is essential to have a sufficient amount of data to avoid overfitting. However, large training sets are not always available, and collecting new data can be costly, especially in fields like medical imaging. Therefore, data augmentation techniques using artificial intelligence can be utilized. In this regard, GAN architecture [17] was used to generate synthetic data.

Generative Adversarial Networks (GANs) are a class of machine learning frameworks consisting of two neural networks, the Generator and the Discriminator, competing in a zero-sum game. In this setup, the generator attempts to create increasingly realistic fake data to deceive the discriminator, while the discriminator strives to improve its accuracy in distinguishing fake data from real data. The generator aims to minimize the probability that the discriminator correctly identifies its fake outputs. Conversely, the discriminator aims to maximize its accuracy in detecting real versus fake data.

In this implementation, data augmentation is achieved using an autoencoder, which consists of an encoder and a decoder, and a simple GAN structure, which consists of a generator and a discriminator. It is worth mentioning that in this structure, the decoder functions as the generator. Initially, in the encoder with two hidden layers, the input data, which are the training data, are mapped from a space with 1935 features to a space with 512 features in the first layer, then to a space with 256 features in the next layer, and finally to a space with 128 features in the output layer. On the other hand, the decoder acts like a generator that aims to generate new samples and its task is to reconstruct the initial input space from the final mapped space. Therefore, it needs to transform the data from a space with 128 features back to a space with 1935 features. The discriminator is also a neural network model that distinguishes between fake and real data. This model returns an output representing the probability that the data is real or fake. The closer this probability is to one, the more real the data is, and the closer it is to zero, the more likely the data is fake. However, the code discriminator functions similarly to the discriminator, with the difference that it must determine the probability of the data being real or fake in the final mapped feature space with 128 features.

In the model training phase, the discriminator is first trained using binary cross-entropy loss function. The loss function for the discriminator is defined as follows:

$$\mathcal{L}_D = -\mathbb{E}_{x \sim p_{\text{data}}}[\log\ D(x)] - \mathbb{E}_{\hat{x} \sim p_{\text{fake}}}[\log\,(1 - D(\hat{x}))]$$

In this equation, x represents the real samples and $\hat{x}$ represents the samples generated by the generator in the original feature space with 1935 features.

The logic behind this equation is as follows: given that the discriminator's goal is to improve its ability to distinguish between real and fake data, the closer D(x) is to one (indicating the probability that the data is real), the better the discriminator is performing. This is because real data x is being correctly identified with high probability. In this case, the logarithm of this probability increases, and since this value needs to be minimized in the error calculation, it becomes negative. On the other hand, the better the discriminator performs in identifying fake data with a probability close to zero, the more effective it is. Therefore, the smaller $D(\hat{x})$ is, the larger 1 - $D(\hat{x})$ becomes, and consequently, the logarithm of this value increases. This value needs to be minimized in the error function, so it becomes negative. Finally, these values are summed to improve the discriminator's performance in handling both real and fake data.

In the second step, it is necessary to train the encoder and decoder together. The loss function for the generator or decoder is binary cross-entropy and is defined as follows:

$$\mathcal{L}_G = -\mathbb{E}_{\hat{x}\sim p_{\text{fake}}}[\log\ D(\hat{x})]$$

We know that the goal of the generator is to produce data in such a way that the discriminator cannot distinguish it from real data. Therefore, we want the discriminator to assign a probability close to one when encountering fake or reconstructed data, essentially recognizing them as real data. Consequently, the higher the probability, the greater the logarithm value. To minimize the error, this value needs to be negative. On the other hand, the reconstruction error of the decoder, which determines the difference between the real data and the reconstructed data, is calculated using the mean squared error (MSE) and is defined as follows:

$$\mathcal{L}_{rec} = \mathbb{E}_{x\sim p_{\text{data}}}[\parallel x - \hat{x} \parallel^2]$$

Finally, the two error values calculated above ($\mathcal{L}_G$, $\mathcal{L}_{rec}$) are summed together to use the combined loss for updating the parameters of the encoder and decoder, thereby training both models. So, the autoencoder and the GAN model need to be trained simultaneously, and the reconstruction error of the decoder should be used along with the GAN structure's error at each step to train the generator and the encoder. Simultaneous training of the autoencoder and the GAN architecture allows the generator to produce more realistic data. This is because, in addition to the GAN error at each step, the reconstruction error of the decoder is also considered, and efforts are made to minimize this error as well.

After these three models, the encoder, the decoder(generator), and the discriminator, have been trained for 100 epochs and the loss has been minimized, samples are generated through the decoder in the feature space with 128 features. These samples are then reconstructed and transformed back to the original feature space with 1935 features by decoder, serving as synthetic data. With data augmentation, for each training sample, one synthetic sample is generated, resulting in the doubling of the training dataset size. In figure 2, a schematic of the GAN structure alongside the autoencoder, which is used for data augmentation, is shown:

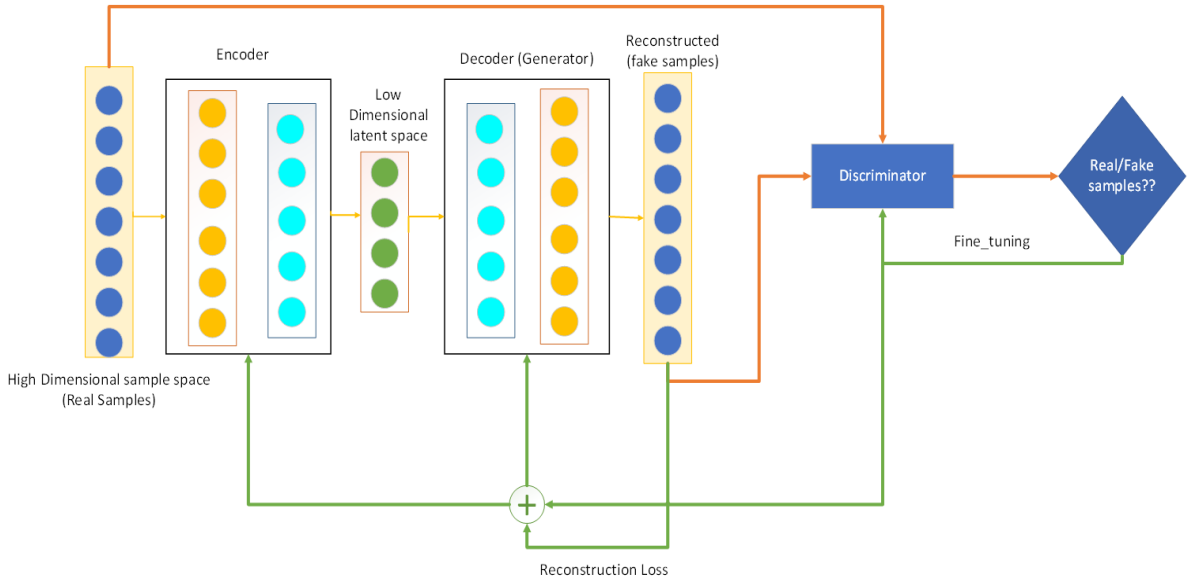

Figure 2| A data augmentation structure consisting of an autoencoder and a GAN architecture, where these two models are trained simultaneously.

Experiments and Results

Given that the NVIDIA Tesla T4 is a versatile and powerful GPU, it is well-suited for modern AI and machine learning workloads, offering a balance of performance, efficiency, and scalability, the models were executed using a T4 GPU.

In this section, we evaluate the performance of the two proposed hybrid models, discussed earlier, by applying k-fold cross-validation on the entire dataset. Given that the models are applied to the entire dataset containing a sufficient number of samples, we use k=10 for k-fold cross-validation by randomly shuffling and splitting each data set into 10 folds. This ensures that there are enough samples for testing (87 samples) and training (784 samples) in each iteration. To evaluate the models, we use the metrics of accuracy, sensitivity, and specificity, similar to other studies such as [8], [9], [10], and [11]. Accuracy measures the proportion of total samples correctly classified by the model. Sensitivity measures the proportion of actual positive cases (individuals with autism) that are correctly identified by the model. Specificity measures the proportion of actual negative cases (healthy individuals) that are correctly identified by the model.

To evaluate our implemented models, we considered baselines such as Support Vector Machine and Random Forest (implemented using the scikit-learn library), as well as models proposed by other papers on the same dataset with similar objectives, such as [8], [9], [10], and [11]. The SVM and Random Forest models were run on the preprocessed dataset (the preprocessing for these baselines and our proposed model is exactly the same as described). However, since the feature dimensions were not reduced, these two baselines were executed on the dataset with 76,636 features. The best

values for the hyperparameters of these two models (kernel type, regularization constant (C), and kernel coefficient (γ) for SVM and the number of trees in the Random Forest) were determined using the Grid Search method.

Experiments Using the CC400 Atlas

In this section, using the 10-fold cross-validation method, we applied both proposed models in this paper along with the provided baselines on all 871 samples of the dataset using the CC400 atlas. Table 1 compares the accuracy, sensitivity, and specificity metrics obtained from our first and second proposed models (with and without data augmentation), and the Random Forest and Support Vector Machine models (without feature dimension reduction).

Method	Accuracy	Sensitivity	Specificity
Second Model	**75.6**	**70.8**	**79.7**
Second Model(no aug)	74.1	67	80.3
First Model	74	70.7	76.9
Random Forest	60.6	30.3	86.8
SVM	69.7	61.4	76

Table 1| Classification results using 10-fold cross-validation based on CC400 atlas

As indicated by the results in the table above, both of our proposed models performed better than the other baselines with accuracies of 74% and 75.6%, respectively. Additionally, the data augmentation in the second model using the GAN architecture was able to increase the accuracy by 1% to 2%.

Experiments and Results on Other atlases

Given that each atlas divides the brain volume into a different number of regions, the connections between various regions will also differ for each atlas, providing distinct information. Therefore, the choice of atlas significantly impacts the performance of the proposed models. In this regard, the performance of both proposed approaches is evaluated using the Power264 [18] and CC200 [13] atlases, which divide the brain volume into 264 and 200 ROIs respectively, and the results are compared. Similar to the CC400 atlas, the Pearson correlation between different regions is calculated and considered as features. Then, the 1935 best features obtained through the Extra-Tree algorithm are selected as the essential features needed. In fact, out of the 19,900 features obtained from the CC200 atlas and the 34,716 features obtained from the Power264 atlas, only 1,935 final features are selected. The results of applying both proposed models along with the considered baselines using 10-fold cross-validation on all dataset samples with the use of the Power264 and CC200 atlases are shown in the two tables below, respectively:

Method	Accuracy	Sensitivity	Specificity
Second Model	72.4	68.5	75.8
Second Model(no aug)	74.2	69.4	78.4
First Model	73.3	68.2	77.8
Random Forest	60	33	81.7
SVM	65.5	60.1	70.8

Table 2| Classification results using 10-fold cross-validation based on Power264 atlas

Method	Accuracy	Sensitivity	Specificity
Second Model	73.4	67.9	78.1
Second Model(no aug)	72.8	65.2	79.4
First Model	71.9	67	76.2
Random Forest	66.3	60.8	71.4
SVM	68.3	64.4	72

Table 3| Classification results using 10-fold cross-validation based on CC200 atlas

As evident from the results above, the models perform better on the CC400 atlas compared to the CC200 and Power264 atlases under the same conditions, because the connectivity among the ROIs in CC400 provides more distinguishing information than in CC200 or Power264 and the best brain atlas is CC400 [19].

Recent Papers Considered as Baselines

In table 4, the results of the 4 proposed models presented in [8], [9], [10], and [11] which have been considered as baselines for our models, are shown:

Method	Accuracy	Sensitivity	Specificity
Proposed Model in **[11]**	66.9	53.2	78.3
Proposed Model in **[8]**	70	74	63
Proposed Model in **[9]**	70.3	68.3	72.2
Proposed Model in **[10]**	71.1	83.7	58.5

Table 4|Classification Results Reported by Recent Papers

By comparing the results of our two proposed models with the results from recent articles, we observe that our proposed approaches have successfully increased the accuracy of data classification by 4% to 6%, demonstrating superior performance.

Conclusion and Future Work

In this paper, our primary goal was to classify individuals with autism from healthy individuals using fMRI data. To achieve this, we used the entire ABIDE 1 dataset. Functional connectivities were used as features due to their valuable information, but the Extra-Tree algorithm was applied to reduce dimensionality and select useful features. Two combined models were then proposed. The first model involves implementing an autoencoder for extracting lower-dimensional useful features in conjunction with a deep neural network with one hidden layer, which is trained simultaneously. The second model employs a Support Vector Machine for classification along with data augmentation using a GAN architecture alongside the autoencoder. Finally, we applied these two models using 10-fold cross-validation on the dataset and achieved a significant improvement in classification accuracy compared to the results obtained from many recent papers that worked on the entire ABIDE 1 dataset with the same objective. Specifically, the first model achieved an accuracy of 74%, while the second model reached an accuracy of 75.6%.

[1] T. Hirvikoski, E. Mittendorfer-Rutz, M. Boman, H. Larsson, P. Lichtenstein, and S. Bölte, "Premature mortality in autism spectrum disorder," *The British Journal of Psychiatry,* vol. 208, no. 3, pp. 232-238, 2016.

[2] A. Zunino *et al.*, "Video gesture analysis for autism spectrum disorder detection," in *2018 24th international conference on pattern recognition (ICPR)*, 2018: IEEE, pp. 3421-3426.

[3] M. Plitt, K. A. Barnes, and A. Martin, "Functional connectivity classification of autism identifies highly predictive brain features but falls short of biomarker standards," *NeuroImage: Clinical,* vol. 7, pp. 359-366, 2015.

[4] B. Sen, N. C. Borle, R. Greiner, and M. R. Brown, "A general prediction model for the detection of ADHD and Autism using structural and functional MRI," *PloS one,* vol. 13, no. 4, p. e0194856, 2018.

[5] Y. Kong, J. Gao, Y. Xu, Y. Pan, J. Wang, and J. Liu, "Classification of autism spectrum disorder by combining brain connectivity and deep neural network classifier," *Neurocomputing,* vol. 324, pp. 63-68, 2019.

[6] J. A. Nielsen *et al.*, "Multisite functional connectivity MRI classification of autism: ABIDE results," *Frontiers in human neuroscience,* vol. 7, p. 599, 2013.

[7] N. C. Dvornek, P. Ventola, K. A. Pelphrey, and J. S. Duncan, "Identifying autism from resting-state fMRI using long short-term memory networks," in *Machine Learning in Medical Imaging: 8th International Workshop, MLMI 2017, Held in Conjunction with MICCAI 2017, Quebec City, QC, Canada, September 10, 2017, Proceedings 8*, 2017: Springer, pp. 362-370.

[8] A. S. Heinsfeld, A. R. Franco, R. C. Craddock, A. Buchweitz, and F. Meneguzzi, "Identification of autism spectrum disorder using deep learning and the ABIDE dataset," *NeuroImage: Clinical,* vol. 17, pp. 16-23, 2018.

[9] T. Eslami, V. Mirjalili, A. Fong, A. R. Laird, and F. Saeed, "ASD-DiagNet: a hybrid learning approach for detection of autism spectrum disorder using fMRI data," *Frontiers in neuroinformatics,* vol. 13, p. 70, 2019.

[10] E. Wong, J. S. Anderson, B. A. Zielinski, and P. T. Fletcher, "Riemannian regression and classification models of brain networks applied to autism," in *Connectomics in NeuroImaging: Second International Workshop, CNI 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 2*, 2018: Springer, pp. 78-87.

[11] A. Abraham *et al.*, "Deriving reproducible biomarkers from multi-site resting-state data: An Autism-based example," *NeuroImage,* vol. 147, pp. 736-745, 2017.

[12] C. Craddock *et al.*, "The neuro bureau preprocessing initiative: open sharing of preprocessed neuroimaging data and derivatives," *Frontiers in Neuroinformatics,* vol. 7, no. 27, p. 5, 2013.

[13] R. C. Craddock, G. A. James, P. E. Holtzheimer III, X. P. Hu, and H. S. Mayberg, "A whole brain fMRI atlas generated via spatially constrained spectral clustering," *Human brain mapping,* vol. 33, no. 8, pp. 1914-1928, 2012.

[14] Y. Zhang, H. Zhang, X. Chen, S.-W. Lee, and D. Shen, "Hybrid high-order functional connectivity networks using resting-state functional MRI for mild cognitive impairment diagnosis," *Scientific reports,* vol. 7, no. 1, p. 6530, 2017.

[15] A. Pinto, S. Pereira, D. Rasteiro, and C. A. Silva, "Hierarchical brain tumour segmentation using extremely randomized trees," *Pattern Recognition,* vol. 82, pp. 105-117, 2018.

[16] Y. Liu, L. Xu, J. Li, J. Yu, and X. Yu, "Attentional connectivity-based prediction of autism using heterogeneous rs-fMRI data from CC200 atlas," *Experimental neurobiology,* vol. 29, no. 1, p. 27, 2020.

[17] I. Goodfellow *et al.*, "Generative adversarial nets," *Advances in neural information processing systems,* vol. 27, 2014.

[18] J. D. Power *et al.*, "Functional network organization of the human brain," *Neuron,* vol. 72, no. 4, pp. 665-678, 2011.

[19] X. Yang, M. S. Islam, and A. A. Khaled, "Functional connectivity magnetic resonance imaging classification of autism spectrum disorder using the multisite ABIDE dataset," in *2019 IEEE EMBS international conference on biomedical & health informatics (BHI)*, 2019: IEEE, pp. 1-4.