Research Article

Employing Machine Learning-Based Predictive Analytical Approaches to Classify Autism Spectrum Disorder Types

Muhammad Kashif Hanif 1, Naba Ashraf 1, Muhammad Umer Sarwar 1, Deleli Mesay Adine 2, and Reehan Yaqoob 1

1 Department of Computer Science, Government College University, Faisalabad, Pakistan
2 Department of Computer Science, Dilla University, Dilla, Ethiopia

Correspondence should be addressed to Deleli Mesay Adine; mesaya@du.edu.et

Received 28 May 2021; Revised 28 November 2021; Accepted 10 December 2021; Published 10 January 2022

Academic Editor: Sheng Du

Autism spectrum disorder is an inherited long-living and neurological disorder that starts in the early age of childhood with complicated causes [1]. The person with ASD can have mental disorders such as anxiety, miscommunication, and limited repetitive interest. ASD can affect the person ability to perform the function properly in the different stages of life. Therefore, initial diagnosis and treatment are tremendously important [2]. One of the most important symptoms of ASD is the behaviour of the affected person with others [3]. Normally, children with autism speak very little and stay quiet. They can adopt specific behaviour from movies and cartoons. For this reason, they can show a risky unexpected behaviour [4].

According to the World Health Organization, ASD affects about 1% population of worldwide [5] and the ratio around the world is increasing very rapidly [1, 6]. According to the Centers for Disease Control and Prevention, the prevalence of ASD has risen to approximately 1 in 68. The frequency of occurrence of ASD in males is approximately four times higher than that in females [7]. ASD can be present in ethnic, racial, and economic groups. In the United States, most children are not diagnosed with ASD until they reach four years [8]. ASD affects about 1.4% population in the region of South Asia [9].

There exists no biological test to diagnose ASD. Current practices to diagnose ASD rely on behavioural patterns [10]. Autism detection in the earlier stage can prevent the patient situation from more deteriorating and also help to decrease the costs that are linked with overdue diagnosis [1]. According to the Diagnostic and Statistical Manual of Mental Disorders, 5th Edition (DSM-5), the severity of ASD can be determined on a spectrum, which has three levels of severity ranging from mild to severe symptoms: Level 1: needs support, Level 2: needs substantial support, and Level 3: needs very substantial support [11, 12].

Machine learning is a growing field, which uses mathematical learning, statistical estimation, and information theories to find useful patterns in the large amount of data [13–16]. Recently, deep learning is the most trending area of research, which is the subset of machine learning and uses
the neural network architectures to model the high-level abstraction in data [17]. These structures contain several layers with processing units that apply linear or nonlinear transformations to the input data [17]. In recent years, different types of deep learning architectures have been already applied to the supervised and unsupervised datasets in the medical field [18–20].

Different studies have employed machine learning and deep learning techniques to predict, diagnose, and classify ASD [21]. The objective of this study was to apply machine learning and deep learning techniques to classify the severity of ASD. The proposed techniques employ cross-validation with hyperparameter tuning. Furthermore, this study performs statistical analysis to compare the models. The experimental results depict that DNN outperforms other methods.

The next section presents the related work. Then, the proposed methodology is discussed. Afterwards, the results are presented. The last section concludes the outcomes.

2. Review of Literature

Researchers have applied different techniques to diagnose the ASD. Dvornek et al. [22] applied the recurrent neural networks with long short-term memory for the classification of individuals with ASD and typical controls for the Autism Brain Imaging Data Exchange (ABIDE) dataset and attained 68.5% accuracy. Van den Bekerom [23] used the NSCH dataset and applied the four machine learning classification algorithms to classify the severity of ASD. The result to classify the severity of ASD attained the accuracy of 0.50% to 0.54% and also used the one-way method, which improved the prediction accuracy of the severity of ASD of 54.1% to 90.2%.

Heinsfeld et al. [24] used the machine learning techniques such as SVM, RF, and DNN in the ABIDE dataset, and the result showed the accuracy of 0.65% to 0.70%. Bi et al. [25] used the multiple SVM to classify the patients and normal controls. Altay and Ulas [26] applied the linear discriminant analysis (LDA) and K-nearest neighbor (KNN) to diagnose the ASD and attained 90.8% and 88.5% accuracy, respectively. Mohammadian Rad and Furlanello [27] applied deep learning to the automatic detection of stereotypical motor movements. They used the convolutional neural network (CNN) to learn a discriminative feature space from raw data. They also combined the long short-term memory with CNN to model the temporal patterns in a sequence of multi-axis.

Kong et al. [28] used the DNN for ASD classification. They extracted the features for each subject and ranked them, and the top 3000 features were used as input to DNN for classification. The proposed method attained 90.39% accuracy. Eslami and Saeed [29] proposed the Auto-ASD-Network model and classified the subjects of ASD from healthy subjects. They used the deep learning to find the useful patterns from the dataset. They applied the auto-tune model to optimize the hyperparameters of SVM and achieved 70% accuracy for the fMRI dataset. Wilson and Rajan [30] applied the deep learning algorithm to detect the ASD from the brain imaging dataset and attained 70% accuracy in the detection of ASD. Pream et al. [31] used the supervised machine learning techniques to identify the syndromic ASD. They attained 98% and 94% accuracy using SVM and decision tree, respectively. Nasser et al. [32] build the artificial neural network model to diagnose the ASD. The data were gathered from the ASD screening application that contains ASD test outcomes based on queries from users. Table 1 provides the summary of the related work.

3. Methodology

The classification of severity of ASD is difficult due to dependence on different features. This study proposes the machine learning models to classify the severity of ASD (Figure 1). In this work, the survey-based national survey of children’s health dataset is used, which was collected from 2011 to 2012. The survey was conducted in the United States [23]. The participants of survey were children of age 2 to 17 years in the United States. The dataset consists of 95 677 records.

Next, the dataset is preprocessed to clean the data and remove the irrelevant parts of dataset. First, the columns that have a single value or few unique values were identified. VarianceThreshold class was used to remove these columns. Next, the duplicate rows were identified and removed. The rows having null values in most of the columns were removed. Moreover, the imputation technique with mean value was employed to handle the remaining missing values. Label encoding was used to encode the target label into numeric values. The dataset has imbalance classes (Figure 2).

The classes in dataset were encoded to 0, 1, 2, and 3, which corresponds to no ASD, Level 1 ASD, Level 2 ASD, and Level 3 ASD, respectively.

Most of the machine learning algorithms are sensitive to the data scaling. For this reason, it is good practice to adjust the data representation [33]. In this study, different scaling techniques were used and StandardScaler technique showed better performance. Therefore, this data transformation technique was used for machine learning pipeline.

Furthermore, different dimensionality reduction techniques were considered [34, 35]. We employed principal component analysis (PCA) for this study. PCA rotates the dataset in a way such that features are statistically uncorrelated. Then, the subset of the rotated features are used based on their importance [33]. For the sake of model development, we used 70% of total dataset as training and the remaining 30% dataset as testing.

In this study, we considered random forest (RF), support vector machine (SVM), naïve Bayes (NB), K-nearest neighbor (KNN), and deep neural network algorithms to predict and classify the severity of ASD. We also applied hyperparameter tuning with stratified k-fold cross-validation for each machine and deep learning model to obtain the best parameters for each model. Stratified k-fold cross-validation was used to attain more reliable estimate of generalization of performance [33].

Cross-validation was used to estimate the performance for each parameter combination. For this purpose, the
Table 1: Summary of related work.

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Authors</th>
<th>Objectives</th>
<th>Method</th>
<th>Dataset</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dvornek et al [22]</td>
<td>Used the long short-term memory network to identify the autism from resting-state fMRI.</td>
<td>Recurrent neural networks (long short-term memory)</td>
<td>ABIDE</td>
<td>68.5%</td>
</tr>
<tr>
<td>2</td>
<td>Van den Bekerom [23]</td>
<td>To use the machine learning to predict the ASD.</td>
<td>Machine learning algorithms (SVM, RF, and NB) and 1-way method</td>
<td>National Survey of Children’s Health (NSCH) data</td>
<td>0.49% to 0.54% in 4 classes using machine learning and 54.1% to 90.2% using the 1-way method</td>
</tr>
<tr>
<td>3</td>
<td>Heinsfeld et al [24]</td>
<td>To apply the deep learning algorithm to identify the ASD.</td>
<td>SVM, RF, DNN</td>
<td>ABIDE</td>
<td>65%, 63%, 70%</td>
</tr>
<tr>
<td>4</td>
<td>Bi et al [25]</td>
<td>To apply the random SVM to classify the ASD.</td>
<td>Random support vector cluster</td>
<td>ABIDE</td>
<td>96.15%</td>
</tr>
<tr>
<td>5</td>
<td>Altay and Ulas (2018)</td>
<td>To use the classification method to identify the children who have ASD or not.</td>
<td>Discriminant analysis, KNN</td>
<td>292 samples</td>
<td>91%, 89%</td>
</tr>
<tr>
<td>6</td>
<td>Kong et al [28]</td>
<td>To identify the ASD using DNN.</td>
<td>DNN</td>
<td>ABIDE</td>
<td>90.39%</td>
</tr>
<tr>
<td>7</td>
<td>Eslami and saeed [29]</td>
<td>To diagnose the ASD using the Auto-ASD-Network based on the deep learning and SVM.</td>
<td>DNN, SVM, state-of-the-art classifier</td>
<td>4 fMRI datasets</td>
<td>80%</td>
</tr>
<tr>
<td>8</td>
<td>Wilson and rajan [30]</td>
<td>To use the brain imaging dataset to identify the ASD.</td>
<td>DL algorithms</td>
<td>ABIDE</td>
<td>70%</td>
</tr>
<tr>
<td>9</td>
<td>Sudha and Vijaya [31]</td>
<td>To identify the ASD by supervised algorithms.</td>
<td>Decision tree, SVM, MLP</td>
<td>Dataset collected from the ASD screening app</td>
<td>98%, 96%, 95%</td>
</tr>
<tr>
<td>10</td>
<td>Nasser et al [32]</td>
<td>To identify the ASD by artificial neural network.</td>
<td>ANN</td>
<td></td>
<td>100%</td>
</tr>
</tbody>
</table>

Figure 1: Proposed methodology.
training and validation data are split for each parameter setting. The accuracy values were computed for each parameter setting for each split in the cross-validation. The mean validation accuracy is calculated for each parameter setting [33]. The best parameters for each model were obtained using the best cross-validation performance [33]. The model was retrained using these best parameters to attain the best performance. Then, the model was evaluated using test data.

For each classifier, we used the pipeline function of Sklearn library and employed standard scalar, PCA, and classifier in this order. For each classifier, PCA parameter n_components having range (2, 10, 5) is used. RF is the combination of many trees and one of the most successful ensemble learning methods [36]. The different values of n_estimators, min_samples_split, min_samples_leaf, and max_features parameters for RF classifier were considered. To obtain the best parameters for RF using grid search, parameter values of n_estimators, min_samples_split, min_samples_leaf, and max_features were considered, respectively. After that, we applied the fivefold cross-validation with grid search for the training and validation datasets. After applying the fivefold cross-validation with grid search, the best parameter value obtained for n_estimators is 200, min_samples_split is 5, min_samples_leaf is 4, max_features is auto, and n_components of PCA is 7. Then, the RF classifier was retrained using these best parameters.

SVM is one of the best classification methods that can handle multiple variables. Kernel, gamma, and C parameters were considered for SVM classifier. The different parameter values of kernel, gamma, and C used were ["linear," "rbf"], [0.001, 0.01, 0.1], and [0.001, 0.01, 0.1], respectively. After applying cross-validation with grid search, the best parameters obtained for kernel, gamma, and C were ["linear," 0.001], and 0.001, respectively. For PCA, n_components value is 2. Then, the SVM is retrained with these best parameters. For NB, var_smoothing parameter containing the np.logspace (0, −9, num = 20) is used and value 1.0 is obtained as the best value using cross-validation and hyperparameter tuning. The best parameter of n_components for PCA is 2. KNN is a nonparametric method and is based on the proximity of the sample feature and the training set [37]. The n_neighbors and p parameters of KNN with values [1–7] and [1, 2, 5] were used, respectively. After that, we applied the fivefold cross-validation in combination with grid search for the training dataset and validation dataset. The best parameter value for n_neighbors is 7, p is 1, and n_components of PCA is 7.

Moreover, cross-validation and grid search were applied to DNN to find the best parameters. The parameters considered for DNN were activation, batch_size, epochs, optimizer, learn_rate, momentum, init_mode, dropout_rate, and weight_constraint. The parameter values and best values for each parameter are listed in Table 2.

4. Results and Discussion

The objective of this work was to classify the severity of ASD. This section describes the results of different machine and deep learning models considered to classify the severity of ASD. For the model development, we used different functions of Keras and Scikit-learn libraries. Keras is easy to use, fast, and open-source neural network library that runs on top of Theano or TensorFlow. It provides the easy workflow to train and define the neural network in just a few lines of code, but Keras does not handle the low-level computation. To build the DNN model, we adapted the sequential model. Scikit-learn library was used to design SVM, NB, RF, and KNN models. It is the library of machine learning that is written in Python programming language. For visualization of results, matplotlib, seaborn, and PyCaret libraries were used. Moreover, the Google Colab platform was used to create models and obtain the results.

To evaluate the performance of these machine and deep learning models, learning curves and precision-recall curves were used. A learning curve is used to show the performance when there is change in the training set or time. In this study, we used model performance on x-axis and test set on y-axis for learning curve. It can detect high variance or high bias in the model. The dataset considered in this work has class imbalance. For this reason, we used the precision-recall curve to check the performance of different models [38]. Machine and deep learning models were trained on dataset using different parameters. Table 2 lists the best parameters obtained for each model using grid search with cross-validation.

Figure 3 summarizes the learning curves of different parameters for DNN model and also shows the loss and accuracy on training sets and test sets. DNN parameters considered were batch_size, epochs (Figure 3(a)), optimization algorithm (Figure 3(b)), learn_rate and momentum (Figure 3(c)), weight initialization (Figure 3(d)), activation function (Figure 3(e)), and dropout regularization (Figure 3(f)). For each case considered, the train and test loss is decreasing in second epochs and these losses converge. The accuracy of train and test is almost the same. This means the model is neither underfitting nor overfitting. Moreover, this shows that model does not have high variance or high bias. The reason for better performance of model is the use of stratified cross-validation.
<table>
<thead>
<tr>
<th>Model</th>
<th>Parameters</th>
<th>Best Parameters</th>
</tr>
</thead>
</table>
| RF    | (i) \text{n \_estimator} = [50, 150, 200]  
(ii) \text{min \_samples \_split} = [2, 5, 10]  
(iii) \text{min \_samples \_leaf} = [1, 2, 4]  
(iv) \text{max \_features} = [\text{auto}, \text{rbf}]  
(v) \text{PCA n \_components} = 7 | (i) \text{n \_estimator} = 200  
(ii) \text{min \_samples \_split} = 5  
(iii) \text{min \_samples \_leaf} = 4  
(iv) \text{max \_features} = \text{auto}  
(v) \text{PCA n \_components} = 7 |
| SVM   | (i) \text{kernel} = [\text{linear}, \text{rbf}]  
(ii) \text{gamma} = [0.001, 0.01, 0.1]  
(iii) \text{C} = [0.001, 0.01, 0.1]  
(iv) \text{PCA n \_components} = 2 | (i) \text{kernel} = \text{linear}  
(ii) \text{gamma} = 0.001  
(iii) \text{C} = 0.001  
(iv) \text{PCA n \_components} = 2 |
| NB    | (i) \text{var \_smoothing} = \text{np.logspace} (0, 9, \text{num} = 20) | (i) \text{var \_smoothing} = 1.0  
(ii) \text{PCA n \_components} = 2 |
| KNN   | (i) \text{n \_neighbors} = [1–7]  
(ii) \text{p} = [1, 2, 5] | (i) \text{n \_neighbors} = 7  
(ii) \text{p} = 1  
(iii) \text{PCA n \_components} = 7 |
| DNN   | (i) \text{activation function} = [\text{softplus}, \text{softmax}, \text{softsign}, \text{relu}, \text{tanh}, \text{sigmoid}, \text{hard\_sigmoid}, \text{linear}]  
(ii) \text{batch \_size} = [10, 20, 50, 80, 100]  
(iii) \text{epochs} = [10, 20, 40, 50, 80, 100]  
(iv) \text{optimizer} = [\text{RMSprop}, \text{SGD}, \text{Adagrad}, \text{Adadelta}, \text{Adam}, \text{Adamax}, \text{Nadam}]  
(v) \text{learn \_rate} = [0.001, 0.01, 0.1, 0.2, 0.3]  
(vi) \text{momentum} = [0.0, 0.2, 0.4, 0.6, 0.8, 0.9]  
(vii) \text{init \_mode} = [\text{uniform}, \text{lecun \_uniform}, \text{normal}, \text{zero}, \text{glorot \_normal}, \text{glorot \_uniform}, \text{he \_normal}, \text{he \_uniform}]  
(viii) \text{dropout \_rate} = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]  
(ix) \text{weight \_constraint} = [1–5] | (i) \text{activation function} = \text{relu}  
(ii) \text{batch \_size} = 10  
(iii) \text{epochs} = 10  
(iv) \text{optimizer} = \text{SGD}  
(v) \text{learn \_rate} = 0.001  
(vi) \text{momentum} = 0.0  
(vii) \text{init \_mode} = \text{uniform}  
(viii) \text{dropout \_rate} = 0.0  
(ix) \text{weight \_constraint} = 1 |

**Figure 3:** Continued.
Next, the learning curves for different machine and deep learning models were computed (Figure 4). The best parameters were obtained for KNN, NB, SVM, RF, and DNN models using cross-validation and hyperparameter tuning. These models were retrained using these best parameters. The learning curves for these models have score on x-axis and training examples on y-axis. Here, higher score means better performance of the model. These learning curves show that training and cross-validation scores are almost the same. The score for training and cross-validation remains almost the same with an increase in training examples. These models attained score in the range of 72% to 80%. The DNN model is good fit as training loss and validation loss gradually decrease and reach a point of stability. Moreover, the gap between training loss and validation loss is small, which means the model has low variance (Figure 4(e)).

Furthermore, confusion matrices were used to visualize the performance for different classes (Figure 5). Most of the classification algorithms correctly classified the data. Class 1 has large number of records in dataset, and most of the algorithms correctly classify data belonging to this class. However, the NB algorithm does not perform well for data belonging to class 3. The reason is distribution of majority class in the training dataset.

Finally, the precision-recall curves were computed (Figure 6). The precision-recall curve has precision on the y-axis and recall on the x-axis. There is trade-off between precision and recall values for different thresholds. For low recall values, the precision value is high. For recall value greater than 0.8, KNN shows a stair step area. This means that a small change in threshold reduces the precision with minor gain in recall value. DNN algorithm attains better average precision compared with other machine learning algorithms. The average precision values are different in the precision-recall curve (Figure 6) and classification algorithm performance (Table 3). The reason is that the precision-recall curve computes the precision of one class with all other classes in the multi-class dataset.

Table 3 shows the performance of all the machine learning and deep learning algorithms for the NSCH dataset. Accuracy is one of the most used metrics to evaluate classification models. DNN exhibits better performance compared with other approaches and attained the highest accuracy result of 87% in the NSCH dataset. The previous study for this dataset attained 50% to 54% accuracy [23].

These algorithms can be evaluated using a statistical test to check whether they have the same performance. The performance comparison of all the supervised classification models together is difficult. Moreover, the quality and performance of a supervised classification model should be evaluated on independent data [39]. For this reason, 5 × 2 cv paired t-test was employed to compare the performance of two models [39, 40]. In this work, a level of significance $\alpha = 0.05$ is used for 5 × 2 cv paired t-test. For comparison of performance of classification models, this method divides the dataset five times into 50% training data and 50% test data. Two models were fit on the training data and evaluated on the test data for each of the 5 iterations in 5 × 2 cv paired t-test. Moreover, training data and test data are rotated to compute the performance again [40]. Table 4 shows pairwise performance comparison of classification algorithms. For all cases, $p \leq \alpha$. This means that the performance of these algorithms is significantly different for this dataset.

Furthermore, the Kruskal–Wallis and Friedman tests were applied to compare the performance of all models together (Table 5). Table 5 shows the resultant $p$ value after applying the tests. The null hypothesis for these tests is that the performance of all classifiers is the same. In this case, the null hypothesis is rejected, which means that the performance of classifiers is not the same.
Figure 4: Learning curves for different models. (a) KNN. (b) NB. (c) SVM. (d) RF. (e) DNN.

Figure 5: Confusion matrix for different models. (a) KNN. (b) NB. (c) SVM. (d) RF. (e) DNN.
**Figure 6:** Precision-recall curve for different models. (a) KNN. (b) NB. (c) SVM. (d) RF. (e) DNN.
5. Conclusion

The early detection of ASD can help to improve the learning capabilities. This study presents different machine and deep learning techniques to classify the severity of ASD. Different experiments were conducted using stratified k-fold cross-validation and hyperparameter tuning. The objective was to attain the best parameters for each machine and deep learning model and retrain the model using these parameters to obtain better performance. The results depict that DNN has better performance when compared with other models. In the future work, we will apply these techniques to different datasets and modalities.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References


