Journal, Advances in Mathematical & Computational Sciences
Vol. 11 No. 3, September, 2023
www.isteams.net/mathematics-computationaljournal

Journal of Advances in Mathematical & Computational Sciences
An International Pan-African Multidisciplinary Journal of the SMART Research Group
International Centre for IT & Development (ICITD) USA

Available online at https://www.isteams.net/mathematics-computationaljournal.info
CrossREF Member Listing - https://www.crossref.org/06members/50go-live.html

Classification of Autism Spectrum Disorder (ASD) Using Multi-Layer Perceptron (MLP) Neural Network

[1]Ayorinde I. T. & [2]Bankole O. A.
[1&2]Department of Computer Science
University of Ibadan
Ibadan, Nigeria.
E-mails: [1]temiayorinde@yahoo.com; [2]bankoleseun2014@gmail.com
Phone Numbers: [1]+2348035289814; [2]+2347061936621

ABSTRACT

Autism Spectrum Disorder (ASD) is a neuro-developmental condition characterized by developmental disability, speech and language delays, abnormal social interactions, behaviour excesses and repetitive and stereotyped behavior. This research is aimed at improving the early diagnosis of ASD using deep learning technique. Raw datasets related to both Autism and Non-Autism cases were collected from the University of Califona Irvine (UCI) repository. Multi-layer perceptron (MLP) neural network was used for the classification. Keras was used to build the MLP neural network model with the aid of Jupyter Notebook. Some of the metrics used to evaluate the model are performance accuracy, probability, roc-auc curve and confusion matrix which shows the precision, recall and F1 score. After running 7 epochs, 98.3% performance accuracy was achieved for the training data while 98.1% performance accuracy was achieved for the test data. With higher epochs, better results can still be achieved. This shows that MLP performs very well in classifying ASD dataset, making it easy for early diagnosis of the disease.

Keywords: Autism spectrum disorder, Classification, Machine learning, MLP.

Ayorinde I. T. & Bankole O. A. (2023): Classification of Autism Spectrum Disorder Using Multi-Layer Perceptron Neural Network. Journal of Advances in Mathematical & Computational Science. Vol. 11, No. 3. Pp 77-88. dx.doi.org/10.22624/AIMS/MATHS/V11N3P5. Available online at www.isteams.net/mathematics-computationaljournal.

1. INTRODUCTION

Autism spectrum disorder (ASD) is a complex developmental disorder that describes certain challenges associated with communication (verbal and non-verbal), social skills, and repetitive behaviors [1]. The symptoms of autism are more visible and easy to identify in children of two to three years of age.

Autism is not just caused by a single factor but can be caused by combination of different factors such as highly environmental factors, genetic and non genetic susceptibility, advanced parental age and complications in pregnancy. According to [2], one out of every 68 children has autism in the United States.

Machine learning (ML) approaches offer promising tools in clinical research. It is the subset of artificial intelligence that focuses on giving machines the ability to learn in an unaided manner without any human intervention [3]. Machine learning deals with programs that learn from experience. Machine learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans.

Machine learning methods require little human involvement. ML works without being explicitly programmed to perform the desired result [4]. Supervised learning tasks, such as classification, is one of the common tasks in ML. In classification, a model is learned from the training data such that it can be used to predict or classify the class in a test dataset.

Deep neural network model, which is a machine learning technique, is an information processing system that mimics the behavior of biological neural networks, which was developed as a generalization of mathematical models of human knowledge [5]. It uses multiple hidden layers in a neural network architecture that mimics the brain's neural connections which are more accurate when used and improves in accuracy as more neurons are added.

The more it is exposed to real-time examples, the more it adapts. Neural Networks are capable of learning from faults thereby increasing its capacity to perform well. Deep neural network is used to classify data into predefined categorical class labels to know whether the patient has autism spectrum disorder or not. Moreover, deep neural networks are trained by using large sets of labeled data and neural network architectures that learn features directly from the data without the need for manual feature extraction.

Multi-layer Perceptron (MLP) is a deep neural network that learns the relationship between linear and non-linear data [6]. It has the advantage of learning from examples and applying them when a similar event arises, making them able to work through real-time events. Hence, MLP are being preferred more for complex problem-solving. Deep learning networks have an automatic feature selection that can effectively discover high level features thus improving the performance and accuracy based on the previous experiences.

Deep learning emphasizes the need to develop accurate prediction models for detecting autism accurately and faster than traditional shallow learning models. Deep learning also, uses multiple hidden layers in a neural network architecture that mimics the brain's neural connections which are more accurate when used and improve in accuracy as more neurons are added. Hence, this study builds an improved autism disorder classification model that enhances early diagnosis of ASD using MLP.

2. RELATED WORKS

This section gives a brief description of related works. Other machine learning techniques different from MLP have being used with different sets of data. Authors in [7] trained and cross-validated an Support Vector Machine (SVM) classifier to differentiate ASD from other Developmental Disorders (DD) based on data from two standardized assessments which are the Autism Diagnostic Interview Revised (ADI-R) and the Social Responsiveness Scale (SRS). Using only five behavioral codes, results of the SVM screen algorithm showed a sensitivity of 89.2% for individuals of 10 years of age or older and 86.7% for individuals under 10 years old. A specificity of 59.0% was reported for individuals of 10 years of age or older and 53.4% for individuals under 10 years old.

Authors in [8] focused on the development of some classification models using machine learning algorithms such as Naive Bayes Algorithm, Decision Tree Algorithm, K-Nearest Neighbors Algorithm and Support Vector Algorithm with real world clinical dataset called CARS and its application in grading childhood autism. This helps the clinical pediatrician to diagnose the grades of autism in the earlier stages. This could serve as an additional mechanism to detect autism and treat children by the pediatricians.

According to [9], 70 children dataset who are diagnosed with ASD and 79 Typically Developed (TD) controls obtained from the Center for Autism Research through Children's Hospital of Philadelphia were used. The study analyzed their whole brain white matter connectivity with the help of a SVM and 10-fold cross validation. By extracting multiple diffusion features from each fiber cluster of each subject, they were able to classify subjects as ASD or TD. The model with the highest accuracy, 78.33%, occurred with 4697 valid fiber clusters. This model produced a sensitivity of 84.81% and specificity of 72.86%.

According to [1] the study proposed a new classification method based on the covering approach, called Rules-Machine Learning (RML). This method offers automatic classifications systems represented as rule sets. The rule sets inside the classifiers can be used by health professionals to assist in the diagnosis process or to advise individuals and their families whether they should seek further evaluation. The rules offered by the proposed method can be easily interpreted by novice users as well as parents, teachers, caregivers, and family members. The experimental tests showed that the RML derives classifiers that are highly competitive when compared to other existing learning approaches in ML such as Boosting, Bagging, decision trees, and rule. The performance evaluation of ML algorithms was based on common metrics such as predictive accuracy, sensitivity, harmonic mean, knowledge derived, and specificity.

3. MATERIALS AND METHOD

The overall system design for this study is shown in Figure 1. Raw datasets were gathered and preprocessed. Feature extraction and selection were performed on the preprocessed data. The data was eventually trained and tested and evaluated using standard metrics like the performance accuracy, probability, roc-auc curve and confusion matrix.

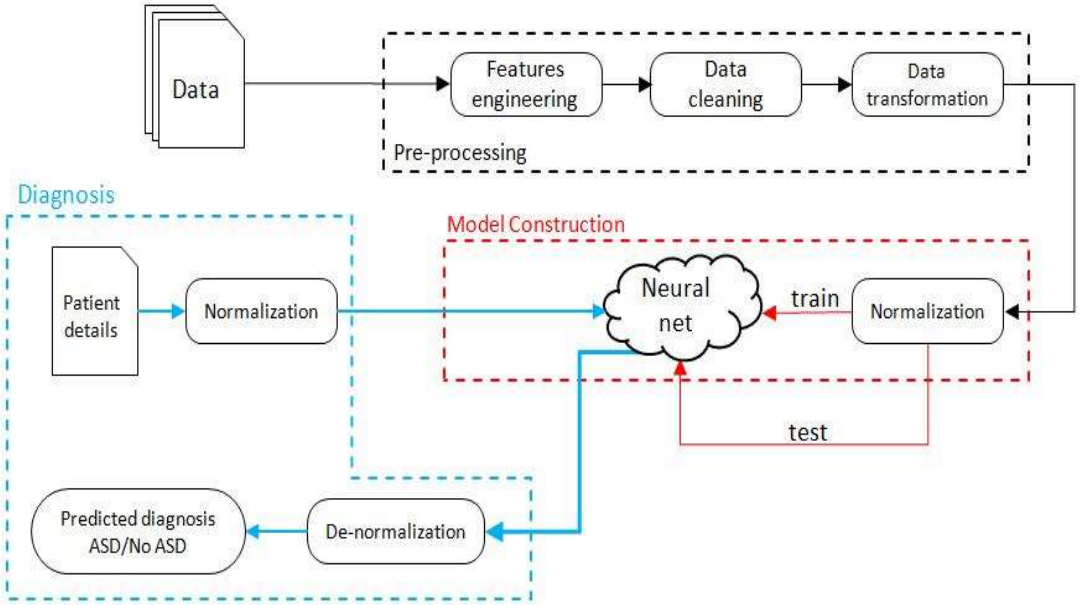

Figure 1: Methodology

3.1 Data Collection

The data collected was the Autistic Spectrum Disorder Screening Data for Toddlers from UCI machine learning repository from the Department of Digital Technology, Manukau Institute of Technology, Auckland, New Zealand. It contains 18 features for over one thousand data points of cases considered class type autistic, and control non-autistic patients. The dataset contains influential features to be used for further analysis especially in determining autistic traits and improving the classification of ASD cases.

3.2 Data Pre-processing

After the raw data were obtained, data pre-processing operations were performed on the dataset.

These include the following:

i. Feature Engineering: Careful feature selection was done to select relevant features.
ii. Data cleaning: Removing missing values, typographical and spelling errors were corrected, character casing are also regularized for uniformity.
iii. Data transformation: All data points (categorical, continuous, and binary) are encoded.

The preprocessing stage is necessary so as to remove noise, missing values and some unusable format which cannot be directly used for the machine learning models. The data preprocessing done in this study was carried out with the use of certain predefined python libraries which were imported to perform some specific jobs as shown in Figure 2.

Preprocessing

```
In [1]: import numpy as np
        import matplotlib.pyplot as plt
        import pandas as pd
```

Figure 2: Predefined Python Libraries

The python libraries are:

1. Numpy: This was used for adding any type of mathematical operation in the code. It is usually used for scientific calculation, adding large multidimensional arrays and matrices in Python.
2. Matplotlib: This was used to import any type of charts in Python for the code
3. Pandas: This was used for importing and managing the datasets. It is an open-source data manipulation and analysis library.

3.2.1 Feature Engineering

Feature engineering is the first step towards the data modeling. It has to do with preparing and selecting the proper input dataset that are relevant to the model and removing features that are irrelevant and that may create biased results and affect the accuracy and performance of the model. Contained in the data are features considered both important and not important for the diagnosis and classification of Autism. The features are been used to determine the result. The deep learning technique used in this study helps to effectively discover high level features thus improving the performance and accuracy over traditional shallow learning models.

3.2.2 Data Cleaning

The data cleaning stage helps to remove noise, missing values and some unusable format which cannot be directly used for the machine learning models.

3.2.3 Data transformation

All data points (categorical, continuous and binary) are encoded or transformed to make it more useful and to bring it to a state that the machine understands. Feature encoding is basically performing transformations on the data such that it can be easily accepted as input to the model while still retaining its original meaning. Categorical data in this study such as ethnicity, age, and sex among others are encoded into numbers. Part of the encoding is shown in Figure 3.

```
In [5]: #Encoding categorical nomnal  data
        from sklearn.preprocessing import LabelEncoder, OneHotEncoder
        genderEncoder  = LabelEncoder()
        fitGender = genderEncoder.fit(features[: , 11])
        features[: , 11] = fitGender.transform(features[: , 11])

        ethnicityEncoder  = LabelEncoder()
        fitEthnicity = ethnicityEncoder.fit(features[: , 12])
        features[: , 12] = fitEthnicity.transform(features[: , 12])

        jaundiceEncoder  = LabelEncoder()
        fitJaundice = jaundiceEncoder.fit(features[: , 13])
        features[: , 13] = fitJaundice.transform(features[: , 13])

        hereditaryEncoder  = LabelEncoder()
        fitHereditary = hereditaryEncoder.fit(features[: , 14])
        features[: , 14] = fitHereditary.transform(features[: , 14])

        fillEncoder  = LabelEncoder()
        fitFill = fillEncoder.fit(features[: , 15])
        features[: , 15] = fitFill.transform(features[: , 15])

        yLabelEncoder  = LabelEncoder()
        yLabel = yLabelEncoder.fit_transform(yLabel)
```

Figure 3: Encoding Categorical, Continuous and Binary Data.

3.3 Data Normalization

The dataset is finally normalized. Normalization is re-scaling of the data from the original range so that all values are within the range of 0 and 1. Normalization is performed on the dataset to improve training time and accuracy of the model. All values are in the same range and in the same scale so that no other variable dominates the other as shown in Figure 4.

```
0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,0
0.8994514303376652,-0.89085185147509,-0.8181118081858187,-1.035013560902976,0.9570313949443363,0.8549889824812247,-1.35
0.9146591207600402,1.0276660746446844,0.8549889824812248,0.2571207739240201,0.6492958957227125,0.056320442016709896,-
0.6162024026097259,-0.42893911461295253,0.1909526546295512,1
0.8994514303376652,-0.89085185147509,-0.8181118081858187,0.9661709157970442,-1.0448978009317815,-1.1696057147987542,-
1.355870140678606,-0.9146591207600402,1.0276660746446844,-1.1696057147987542,-2.008588066681719,0.6492958957227125,-
0.4137600789742452,-0.6162024026097259,-0.42893911461295253,0.1909526546295512,0
0.8994514303376652,-0.89085185147509,1.222326814005439,0.9661709157970442,0.9570313949443363,-
1.1696057147987542,0.7375337578416593,1.0933034802834856,1.0276660746446844,0.8549889824812248,-
1.5050972132137768,0.6492958957227125,0.526400963007665,-0.6162024026097259,-0.42893911461295253,0.1909526546295512,1
0.8994514303376652,1.1225210997139203,1.222326814005439,0.9661709157970442,0.9570313949443363,0.8549889824812247,-
1.355870140678606,1.0933034802834856,1.0276660746446844,0.8549889824812248,1.012357054125933,0.6492958957227125,0.05632
0.6162024026097259,-0.42893911461295253,0.1909526546295512,1
0.8994514303376652,-0.89085185147509,1.222326814005439,0.9661709157970442,0.9570313949443363,0.8549889824812247,0.73753
0.9146591207600402,1.0276660746446844,0.8549889824812248,-0.8757336463788493,0.6492958957227125,-
0.4137600789742452,1.6228433965282005,2.3313332030871923,0.1909526546295512,1
0.8994514303376652,1.1225210997139203,1.222326814005439,0.9661709157970442,0.9570313949443363,0.8549889824812247,0.7375
0933034802834856,1.0276660746446844,0.8549889824812248,-1.1274790731128204,0.6492958957227125,-0.4137600789742452,1.622
0.42893911461295253,0.1909526546295512,1
-1.111788770656106,-0.89085185147509,1.222326814005439,0.9661709157970442,0.9570313949443363,-1.1696057147987542,0.7375
0.9146591207600402,-0.9730787311878073,-1.1696057147987542,0.38299348729100563,0.6492958957227125,-
0.4137600789742452,1.6228433965282005,-0.42893911461295253,0.1909526546295512,1
0.8994514303376652,-0.89085185147509,-0.8181118081858187,0.9661709157970442,0.9570313949443363,0.8549889824812247,-1.35
0.9146591207600402,1.0276660746446844,0.8549889824812248,1.012357054125933,0.6492958957227125,0.056320442016709896,-
0.6162024026097259,-0.42893911461295253,0.1909526546295512,1
```

Figure 4: Re-scaling the processed data into specific range

3.4 Data Training and Testing

To ultimately achieve predictive modeling, the dataset was divided into training set and test set. The training set consists of 80% of the whole dataset and test set consists of 20%. These percentages were chosen so as to get enough dataset trained while using the remaining dataset for testing. This is shown in Figure 5.

```
#Split dataset into train and test
from sklearn.model_selection import train_test_split
feature_train, feature_test, yLabel_train, yLabel_test = train_test_split(features, yLabel, test_size = 0.2, random_state = 0)
```

Figure 5: Splitting Dataset into Training and Testing Data

4. RESULTS AND DISCUSSION

The framework used in building this deep neural network was Keras and TensorFlow. Keras is an Open Source neural network library written in Python that runs on top of TensorFlow. TensorFlow performs the computation and development of the models. It is designed to be user friendly, modular, and extensible. Also, the metrics are being calculated and implemented using the Keras Application Programming Interface (API) during model training and evaluation. In addition to offering standard metrics for classification and regression problems, Keras also allows the definition and reporting of custom metrics when training deep neural network.

.4.1 Training and Testing the Model Using Keras

Training and testing are important to evaluate the performance of a fully specified classifier. Also, to proof that the model is robust enough to work with data it has not seen before, the model in this study was trained with 80% of the dataset. The model sees and learns from this data. Training data is used to adjust the weights or parameters of the network. 20% of the dataset was used for testing the model. Training occurs over epochs and each epoch is split into batches. One epoch comprises of one or more batches. The training process runs for a number of seven (7) epochs and used a relatively small batch size of 20. The number of dataset rows that are considered before the model weights are updated within each epoch are also set. This is called the batch size and set using the batch_size argument.

4.2 Evaluating the Model Using Keras

Evaluating the performance of the deep neural network on the autism dataset is required after training the neural network. This gives an idea of how well the dataset has been modeled.

4.2.1 Performance Accuracy Result

The output on each epoch shows the loss and accuracy on both the training dataset and test dataset. Also, The **acc** is the accuracy of the model at training time and does not give a good evaluation of the model whereas the **val_acc** is the accuracy of the model at testing time and since the model has not seen the testing data before, it is a better evaluation metric. This is shown in Figure 6. With 7 epochs, 98.3% accuracy was recorded for the training dataset while 98.1% accuracy was achieved for the test dataset.

```
In [10]: model.fit(feature_train, yLabel_train, batch_size=20,
                   epochs=7, verbose=2, validation_data=(feature_test, yLabel_test))

         Train on 843 samples, validate on 211 samples
         Epoch 1/7
          - 0s - loss: 0.2349 - acc: 0.9336 - val_loss: 0.2190 - val_acc: 0.9147
         Epoch 2/7
          - 0s - loss: 0.1842 - acc: 0.9419 - val_loss: 0.1784 - val_acc: 0.9336
         Epoch 3/7
          - 0s - loss: 0.1503 - acc: 0.9526 - val_loss: 0.1497 - val_acc: 0.9384
         Epoch 4/7
          - 0s - loss: 0.1247 - acc: 0.9597 - val_loss: 0.1260 - val_acc: 0.9526
         Epoch 5/7
          - 0s - loss: 0.1051 - acc: 0.9727 - val_loss: 0.1078 - val_acc: 0.9621
         Epoch 6/7
          - 0s - loss: 0.0892 - acc: 0.9798 - val_loss: 0.0943 - val_acc: 0.9621
         Epoch 7/7
          - 0s - loss: 0.0762 - acc: 0.9834 - val_loss: 0.0817 - val_acc: 0.9810
```

Figure 6: Performance Accuracy Result

4.2.2 Probability Result

The probability recorded from this study's model when trained was 98.5% as seen in Figure 7. With this percentage, there is the probability of ASD among the patients. That is, the model predicts that certain patients have 98.50% chance of having ASD.

```
In [17]: prediction = model.predict(data)

In [18]: print(prediction[0][0]*100,'% probability of ASD')

         98.50325584411621 % probability of ASD
```

Fig 7: Probability Result

4.2.3 Confusion Matrix

The Confusion Matrix is a table that describes the performance of a classification model on a set of test data for which the true values are known. It is used to visually observe how well the deep neural network performs on a classification algorithm. It usually gives the true positive, false positive, false negative and true negative ratios as explained below.

i. True Positive: This shows that the model correctly predicted a class as having autism.
ii. False Positive: This shows that the model incorrectly predicted an autism case as present but is absent
iii. False Negative: This shows the model incorrectly predicted an autism case as absent but is present
iv. True Negative: This shows that a model correctly predicted Negative cases as negative.

Figure 8 shows the result of the confusion matrix.

```
In [13]:
predict_y = model.predict(data_x)
round_pred_y = np.around(predict_y)
conf_mat = confusion_matrix(data_y, round_pred_y)
class_rep = classification_report(data_y, round_pred_y)

print(conf_mat)
print(class_rep)

[[268   0]
 [ 71 503]]
```

Figure 8: Result of the confusion matrix generated

From the confusion matrix result, there are 268 True Positive, 0 False Negative, 71 False Positive and 503 True Negative.

4.2.4 Precision

Precision simply means that an algorithm returned substantially more relevant results than irrelevant ones. The negative predictive value (NPV) and positive predictive value (PPV) represent the percentages of negative and positive diagnostic test instances that are true negative and true positive outcomes respectively. The precision result from this study is shown in Figure 9.

	precision	recall	f1-score	support
0.0	0.79	1.00	0.88	268
1.0	1.00	0.88	0.93	574

Figure 9: Results Obtained for Precision, Recall and F1 Score

From Figure 9, precision for class 0.0 (non-ASD) and class 1.0 (ASD) are shown. A precision of 0.0 is the precision of the model in predicting class 0.0 (non-ASD) that is, how precise is the model in predicting a patient with no ASD. The class 1.0 is the precision of the model in predicting class 1.0 (ASD). 79% was gotten for non-ASD and 100% for ASD. This is calculated by converting into percentage by multiplying the value by 100. This significantly shows that this study's model performs excellently in terms of precision.

4.2.5 Sensitivity/Recall

Recall, also known as sensitivity is the ratio of true positives to the actual total positives in the dataset. It is the ability to correctly detect true positives of each available category. In this study, it calculates the percentage of the autism spectrum disorder that is truly positive. The result is shown in Figure 9. From the result, 100 % was recorded for class 0.0 (non-ASD) while 88% was recorded for class 1.0 (ASD).

4.2.6 F1-Score

The F1 score makes use of both precision and recall values. It accounts for both false positives and false negatives. The higher the precision and recall, the higher the F1-score. It ranges between 0 and 1. The closer it is to 1, the better the model. From this study, the F1 score for the model as seen in Figure 9 is 0.88 for class 0.0 (non-ASD) and 0.93 for class 1.0 (ASD). These values show that this model is a good classifier.

4.2.7 ROC-AUC Curve

The **Receiver Operating Characteristics (ROC) - Area Under The Curve (AUC) metric gives** performance measurement at various threshold settings. The ROC curve plots true positive and false positive rates while the AUC measures the area underneath the entire ROC curve. The higher the AUC, the better the model is at predicting correct classes, hence, the better the model is at distinguishing between patients with ASD and non-ASD. Figure 10 shows the ROC-AUC curve obtained in this study. The AUC value is 0.938 which shows a very good classification model.

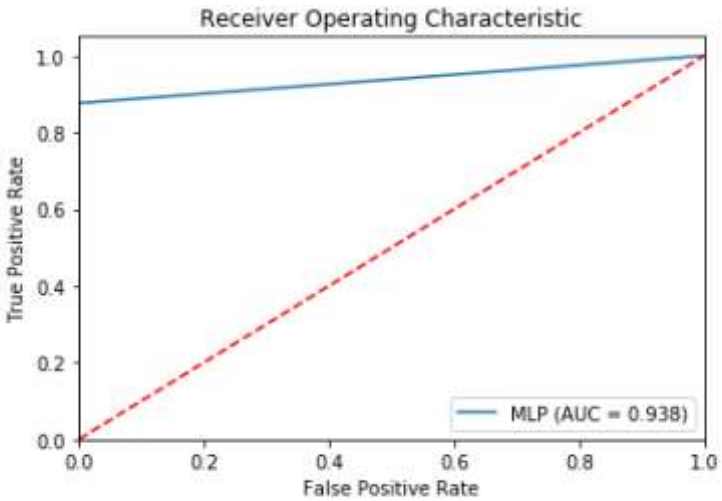

Figure 10: ROC-AUC Curve

5. CONCLUSION

This study has shown the efficacy of MLP in classifying dataset for ASD for early diagnosis of the disease. From this study, it is evident that MLP is a good deep neural network algorithm for classification problems as it gives training data accuracy of 98.3% and testing data accuracy of 98.1% in this study.

REFERENCES

[1] Thabtah F. and Peebles D. (2020). A new machine learning model based on induction of rules for autism detection. Health Informatics Journal, 2020 Mar;26(1):264-286. https://doi.org/10.1177/1460458218824711

[2] Towle P and Patrick P. (2016) Autism spectrum disorder screening instruments for very young children: a systematic review. New York: Hindawi Publishing Corporation.

[3] Toh C. and Brody J. P. (2021). Applications of Machine Learning in Healthcare. An open access peer-reviewed chapter In When Artificial Intelligence Meets the Internet of Things. Smart Manufacturing. DOI: 10.5772/intechopen.92297. https://www.intechopen.com/chapters/72044.

[4] Duggal N. (2022). Top 10 Machine Learning Applications and Examples in 2022. https://www.simplilearn.com/tutorials/machine-learning-tutorial/machine-learning-applications

[5] López O. A. M., López A. M. and Crossa J. (2022). Multivariate Statistical Machine Learning Methods for Genomic Prediction. ISBN 978-3-030-89009-4 ISBN 978-3-030-89010-0 (eBook). https://doi.org/10.1007/978-3-030-89010-0

[6] Bento C. (2021). Multilayer Perceptron Explained with a Real-Life Example and Python Code: Sentiment Analysis. https://towardsdatascience.com/multilayer-perceptron-explained-with-a-real-life-example-and-python-code-sentiment-analysis-cb408ee93141

[7] Bone, D., Goodwin, M. S., Black, M. P., Lee, C. C., Audhkhasi, K., & Narayanan, S. (2015). Applying machine learning to facilitate autism diagnostics: pitfalls and promises. Journal of autism and developmental disorders, 45(5), 1121–1136. https://doi.org/10.1007/s10803-014-2268-6.

[8] Kanimozhiselvi1 C. S., Jayaprakash D. and Kalaivani K. S. (2019). Grading Autism Children Using Machine Learning Techniques. International Journal of Applied Engineering Research ISSN 0973-4562 Volume 14, Number 5 (2019) pp. 1186-1188 © Research India Publications. http://www.ripublication.com

[9] Zhang F., Savadjiev P., Cai W., Song Y., Rathi Y., Tunç B., Parker D., Kapur T., Schultz R. T., Makris N., Verma R., O'Donnell L. J. (2018). Whole brain white matter connectivity analysis using machine learning: an application to autism. Neuroimage. 2018;172:826–837. doi: 10.1016/j.neuroimage.2017.10.029