2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE) | 978-1-6654-1974-1/20/$31.00 ©2020 IEEE | DOI: 10.1109/CSDE50874.2020.9411531

Autism Spectrum Disorder Detection in Toddlers for Early Diagnosis Using Machine Learning

Shirajul Islam
Department of Computer Science And Engineering
Brac University
Dhaka, Bangladesh
shirajul.islam@g.bracu.ac.bd

Tahmina Akter
Department of Computer Science And Engineering
Brac University
Dhaka, Bangladesh
tahmina.akter2@g.bracu.ac.bd

Sarah Zakir
Department of Computer Science And Engineering
Brac University
Dhaka, Bangladesh
sarah.zakir@g.bracu.ac.bd

Shareea Sabreen
Department of Computer Science And Engineering
Brac University
Dhaka, Bangladesh
shareea.sabreen@g.bracu.ac.bd

Muhammad Iqbal Hossain
Department of Computer Science And Engineering
Brac University
Dhaka, Bangladesh
iqbal.hossain@bracu.ac.bd

***Abstract*—Autism spectrum disorder (ASD) is a disorder where patients are unable to express and interact. Recently it is an issue to be concerned that one in 59 children has identified as an autism spectrum disorder patient. ASDs start from childhood but symptoms can be detected in adulthood. That is why these children are not being able to have proper treatment at an early age and that causes more complexity in their health. Research shows that a diagnosis of autism at an earlier age can be more reliable and stable. Therefore, our study aims to estimate ASD (autism spectrum disorder) at a sooner possible time and increase more accuracy than the previous research and reduce medical costs. In our thesis paper, we want to predict and distinguish between autistic and non-autistic children by using a machine learning approach. Firstly, we have gathered data from the surveillance side as much as possible. We also set some particular questions and try to find maximum accurate answers to all questions. Furthermore, supervised learning algorithms are applied to diagnosis whether children meet the symptoms for ASD. Among all applied algorithms KNN and Random Forest shows maximum accuracy and speed to diagnosis. Above all, our final goal is to create an online tool that can provide machine learning-based analysis to a user to detect autism at an early age precisely.**

***Index Terms*—ASD, Toddler, Machine learning, Prediction, Treatment, KNN, Random Forest Classifier**

I. Introduction

AUTISM is also referred to as autism spectrum disorder (ASD) which is a neurodevelopmental disorder. Various impairments in social communication and interaction is caused by it. Also, the existence of unvaried patterns of behavior or activity. From research by the Center for Disease Control, it is found that an estimated 1 out of 54 children is affected by autism in the United States [1]. Detecting autism earlier in one life can make a big difference than treating it later. If it is detected early the toddler can get improvement in his or her communication skill through therapy. From the age of 12 months to 18 months, symptoms start to show, and if detected earlier and treated accordingly [2]. We aim to detect autism at an early age so that necessary steps can be taken to prevent it from getting worse. Early detection can help in not spending a lot in the future as it has been eliminating those situations such as developing social skills and so on. According to WHO every year among 160 children, one is diagnosed with ASD traits all over the world [3]. Treating ASD earlier is always the best option for toddlers as they are still developing. Against this huge burden, only 200 psychiatrists and limited professionals are serving. As doctors have to depend on observing the responses of toddlers as well as listening to the concerns of their parents so it is not easy to make an ASD diagnosis at all. That is why the objective of the work is to detect ASD symptoms at a premature age at minimum time and search for maximum accurate dataset to improve the accuracy of previous research and using maximum data. Besides, this work focuses on developing a model using supervised Machine Learning techniques.

Another purpose of our research is to generate a mobile application so that anyone can use them anywhere and detect whether their child is in the very beginning stage of autism. Some people don’t want to go to the doctor or hospital because of the fear of society mostly in Bangladesh. So we thought of making a mobile application based on our model so that it becomes easier for them to test early and take care of their children. We will try to develop it in the future extension of our research.

II. Proposed Model

A. Dataset Description

To apply supervised algorithms, we have used 1054 datasets. It has been categorized into three areas including medical, health and social science. Besides, the attribute type is categorical, continuous and binary. The queries of the Q-CHAT and AQ tools are 10. Each item value assigned from Kaggle and UCI ML repository [4]. The contained datasets are for toddlers and it represents 30.76% female and 69.24% male in toddlers. There are a total of 1054 toddler case values. In addition, the following ten questions were asked to the parent, self, caregiver and medical staff. The Q-CHAT questions possible answers are- 'Always, Usually, Sometimes, Rarely and Never' which are selected as '0' or '1' in the dataset. If their reply was always /usually/ sometimes then '1' is allocated. Otherwise it is 0 for 10 (A10). Again for question 0-9 (A0-A9), the response was recorded as '1' for the answer Sometimes / Rarely / Never. If any child cuts more than 3(Q-CHAT-10-score) then the child will be detected with ASD otherwise no ASD traits are detected. Nevertheless, figure 3.1 and figure 3.2 describe the feature of data in different columns [4].

Variable in Dataset	Corresponding Toddler Features
A1	Does your child look at you when you call his/her name?
A2	How easy is it for you to get the eye contact of your child?
A3	Does your child point to indicate that s/he wants something?
A4	Does your child point to share interest with you?
A5	Does your child pretend? (e.g. care for dolls, talk on toy phone)
A6	Does your child follow where you're looking?
A7	If you or someone else in the family is visibly upset, does your child shows signs
A8	Would you describe your child's first word?
A9	Does your child use simple gestures? (e.g. wave goodbye)
A10	Does your child stare at nothing with no apparent purpose?

Fig. 1: Details of variables mapping to the Q-Chart-10 screening methods

The values were collected based on the Q-CHAT questionnaires. Here, most of the data are boolean or binary type which will be fitting for the classifiers to compute and will not need preprocessing. Besides these there are also some integer and string type value fields which will need conversion before we use them in any classifiers. Otherwise we will not be able to reach the optimal result. The table below contains the data type of each field and also the explanation of the features along with the data acquiring method.

Feature	Answer type	Description
A1: Question1 Answer	Binary (0,1)	The answer of the question based on the screening method used
A2: Question1 Answer	Binary (0,1)	The answer of the question based on the screening method used
A3: Question1 Answer	Binary (0,1)	The answer of the question based on the screening method used
A4: Question1 Answer	Binary (0,1)	The answer of the question based on the screening method used
A5: Question1 Answer	Binary (0,1)	The answer of the question based on the screening method used
A6: Question1 Answer	Binary (0,1)	The answer of the question based on the screening method used
A7: Question1 Answer	Binary (0,1)	The answer of the question based on the screening method used
A8: Question1 Answer	Binary (0,1)	The answer of the question based on the screening method used
A9: Question1 Answer	Binary (0,1)	The answer of the question based on the screening method used
A10: Question1 Answer	Binary (0,1)	The answer of the question based on the screening method used
Age	Number	Toddlers (months), children, adolescent, and adults(year)
Score by Q-chart-10	Number	1-10(less than or equal 3 no ASD traits > 3 ASD traits)
Sex	Character	Male or female
Ethnicity	String	List of common ethnicities in text format
Born with jaundice	Boolean (true, false)	Whether the case was born with jaundice
Family member with ASD history	Boolean (true, false)	Whether any immediate family member has a PDD
Who is completing the test	String	Parent, self, caregiver, medical staff, clinician etc.
Why are you taken the screening	String	Use input textbox
Class variable	String	ASD traits or No ASD traits (automatically assigned by the ASD Tests app). (yes/No)

Fig. 2: Features collected and their descriptions

B. Data preprocessing

The acquired dataset needed some modifications before we could test our classifier algorithms on it. We preprocessed the dataset in such a way that it was able to provide prime output. Previously we found that unsorted and unprocessed data affected our result scores hugely. To get rid of the problems we followed some particular steps so the algorithms would be able to give more precise results. Most of our values in the dataset were binary and boolean values. They were based on polar questions primarily. However, few of the question criteria required non integer and non boolean type answers. These data were recorded in string format. For our algorithm to give the optimal result we needed to convert these string type values to binary. We used one hot encoding to transform the values to binary. This technique was applied on only the values of the column 'Ethnicity'. One hot encoding is a method that converts categorical variables into a type which can be given to ML algorithms to do a better job of projection. For this process we first needed to switch the string values of 'Ethnicity' column by default then applied one hot encoding on these values to get the unique binary correspondents. We also used Standard deviation in 'Age_mons' and 'Qchat-10-Score' columns which transforms the value within the range of -3 to 3. Lastly, we dropped two column values that indicated the case number and who completed the Q-CHAT questionnaires test as they were not needed for our analysis. After this we were able to read data for our experimentation. Next we decided the ML algorithms which we will use for our experimentation based on our data size, the data features and the target set of results we are looking for. Initially we choose the following algorithms:

- Random forest Classifier.
- Support Vector Machine.
- Decision Tree Classifier.
- Gaussian Naive Bayes Classifier.
- K Nearest Neighbor.
- Logistic Regression.

For these algorithms we need to import the required libraries individually for each of the classifiers. Our goal by applying these classifiers was to find the accuracy, precision and recall scores for which we imported the necessary libraries. For KNN, Random Forest and Naive Bayes classifiers we imported AdaBoost classifier and GradientBoosting classifier along with the other required ones. Some of the common libraries all of the libraries used were pandas, numpy, plt, plot ROC curve etc. Our experiments were done using Google colab notebook. We imported libraries and the preprocessed data on colab to compute our target result. At the beginning of the process we set the features of the data to X and the output to Y. We then splitted our data set into a test and train set, where 20% of the dataset were used for testing and 80% were used for training.

C. Model Description

The proposed model ensures a more accurate result to the research about autism detection at an early age of autism. We used different machine learning algorithms to get more accurate results like the Support Vector machine, Random Forest, Naive Bayes and KNN, Logistic Regression and Decision Tree. First, we collected the questions by following a standard. Those who are required to answer them we collected their answers and sent them to data preprocessing. Data preprocessing is adjusting if it is a non-matrix format type or not, how many attributes and instances are there and how many we need to run in a specific algorithm. In our case there are 1054 instances and 18 attributes including the class variables.

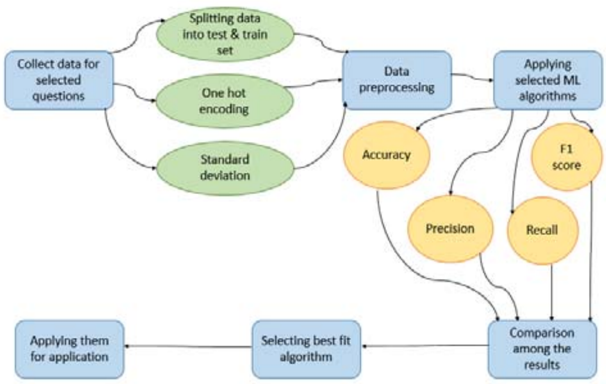

Fig. 3: Model flow diagram

We started working by collecting data of question sets to train our machine learning algorithm. For that we imported the necessary libraries. After that we allocated 80% of our dataset for training and 20% for testing. We have set the X matrix for feature and Y matrix for output. By using one hot encoding we have transformed a column ‘ethnicity’ to unique binary values. For data preprocessing we have also used standard deviation in ’Age_mons’ and ’Qchat-10-Score’ which transforms the value within the range of -3 to 3. After preprocessing we started applying our ML algorithms. We used Logistic Regression, decision tree, Random forest, Naive Bayes, SVM and KNN. Logistic regression and decision trees have been later excluded as they overfit our dataset. But the other algorithms resulted in a good way, they did not overfit or underfit as our dataset is less for the other algorithm. Accuracy of all the algorithms has been compared to each other for better results. Furthermore we found the score for precision and recall. Precision does not depend on accuracy as for precision the values are not always the same. After getting precision,accuracy and recall we compared the results and selected the best fit algorithm for our model. Lastly we apply them for the application of our model. Our goal is to improve the accuracy of the result differing from the other autism detection research. Besides this our second target is to get as much data possible to train our model. We have planned to offer a mobile application of this model in the future and for that we will be needing a fund sponsor.

III. Result and Analysis

Firstly, to build a model the important thing is to find the goodness of a model. And the most valuable work is going to be how good the predictions are. In here, the following diagram shows the outcome of a model. This is called a ROC curve. In a ROC curve, it can be determined how an algorithm performs by observing the AUC.

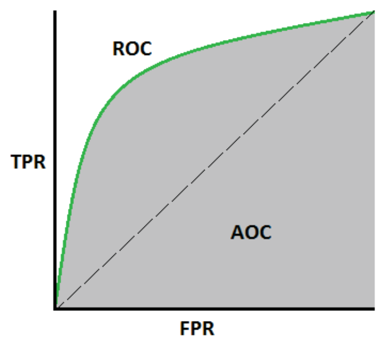

Fig. 4: ROC curve example [5]

Also by using Confusion matrix, it can be determined of the goodness of a model. Now a confusion matrix which basically is for defining the results of a classification algorithm. Accuracy, Precision, Recall, F1 score and other measures except AUC can be calculated from the left most four values. And these four values/parameters are true negative, true positive, false negative and false positive which are the answers of class(output). In the following table the true positives and true negatives are shown in green color because these are correctly predicted. Also, false positives and false negatives are shown in red color because these are incorrectly predicted. These red color values are needed to be minimized. The details of these values are-

	Predicted Class	
Actual Class	Class = Yes	Class = No
Class = Yes	True Positive	False Negative
Class = No	False Positive	True Negative

Fig. 5: Confusion matrix parameters

True Positives(TP): True positives are correctly found values. This means the genuine result and the anticipated result both are yes.

True Negatives(TN): True negatives are also correctly found results. This means the genuine and anticipated result both are no.

False Positives(FP): This means the real result is no but the anticipated result is yes

False Negatives(FN): This means the real result is yes but anticipated result is no.

To identify how good a model has performed can be measured by some parameters namely Precision, Accuracy, F1 Score and Recall. By using these four values we can calculate Precision, Accuracy, F1 Score and Recall. These scores are:

Accuracy: Accuracy means how close the measured value is compared to the standard value. Proportion of the total number of predictions that are actually correct.

$$Accuracy = (TP + TN)/(TP + FP + FN + TN) \quad (1)$$

Precision: It indicates to the adjacency of two or more measurements to each other. If we take any measurement 3 times and every time we get the same value though the value is not close to the standard value then the measured value will be considered as precise but not accurate. So precision does not depend on accuracy.

$$Precision = TP/(TP + FP) \quad (2)$$

Recall(Sensitivity): Recall actually determines how many of the real positives a model captures by labeling it as true positive. So, recall is accurately anticipated positive inspection ratio to all the inspections in the real table.

$$Recall = TP/(TP + FN) \quad (3)$$

F1 Score: F1 takes both false negative and false positive in the count and takes a weighted average. F1 is usually more useful than accuracy.

$$F1Score = 2*((Precision*Recall)/(Precision+Recall)) \quad (4)$$

So, these parameters and ROC curve show the performance of a model [6].

We have implemented four supervised machine learning classifier algorithms in this paper. These are K-NN, Naive Bayes, SVM and Random Forest. For every experiment we used 20% data for testing and 80% for training the model.

SVM Classifier Experiment: Supporting vector machines is a different type of machine learning classifier that shows some different kind of result than other models. SVM is a model which works best for limited data processing. As we are dealing with less than two thousand data, this algorithm was quite fitting for the task. In our experiment of SVM, we basically used the SVM that performed very well. Before some proper preprocessing and without using the SVM it showed almost 71-74% accuracy which was not so good. After preprocessing and by using the SVM it showed us 83% accuracy. We used gamma as 0.7 and C as 1.0 for parameters in applying the classifier. Then the result of its accuracy showed 83% almost. The precision score it showed is 89%. Moreover, 88% and 88% of Recall and F1 score. These scores can be seen in the following results table. The time it took while generating the result was 1.91 seconds which is quite higher than some other models.

Naive Bayes Classifier Experiment: In Naive bayes experiment we used the Gaussian Naive bayes which showed some promising results and the results were better than Supporting Vector Machine classifier. We kept the parameters for Naive Bayes as default. We did not apply any different parameters for Naive bayes as it showed some good results in default parameters. It gave 89% of accuracy which is better than the first algorithm and the precision score is 100% which is impressive. Also, Recall score was 84% which is not so good as before and the F1 score it showed is 91%. It took 1.53 seconds to complete the experiment and to produce the result which is the lowest among all other algorithms.

Random Forest Classifier Experiment: The third algorithm we used is Random Forest algorithm which performed as one of the best performers. We kept parameters of random forest as best as possible. Then the results it showed are better than the first two algorithms. It showed 93% of accuracy which is the second best result we have got. Precision was 92% which is good enough also. Then the recall and F1 score was 100% and 96% which is also better than others. All the results of this algorithm were good except one thing and that is the time it took was higher than any other algorithms. 2.30 seconds is the highest time it took than other algorithms.

K-NN Classifier Experiment: This algorithm was the last successful implementation of our experiments and it showed the best result among all other algorithms. We used parameters of n_neighbors as 5, metric as âminkowskiâ and p equal to 2. It performed in this metric minkowski is better than others. The accuracy it showed is 98% which is the

finest among all results we got. The scores of Precision, Recall and F1 score as 100%, 97% and 99%. All are good scores as they had higher results than others. The Precision score is better than some others. And Precision indicates to the adjacency of two or more measurements to each other. Also, high precision rate means low false positive rate which means the score is high and this is good. Moreover, it should be mentioned that this algorithm did a good time management like it took only 1.55 seconds to complete the calculation.
The result of applying the algorithms are given below:

Method	Accuracy	Precision	Recall	F1 Score	Time
SVM	0.83	0.89	0.88	0.88	1.91 sec
Naive Bayes	0.89	1.0	0.84	0.91	1.53 sec
Random Forest	0.93	0.92	1.0	0.96	2.30 sec
K-NN	0.98	1.0	0.97	0.99	1.55 sec

TABLE I: Results after applying Classifier algorithms

Firstly, to obtain a good result from a dataset we had to do some preprocessing that is necessary for a machine learning model to perform well. Also, in machine learning we should do some proper data preprocessing for future proofing that means the acceptability of models and to perform well depends on preprocessing also. So, what we did in preprocessing is, we kept all of the values of features in binary except three features values. In those three features one was “Ethnicity” and it was in string type. For the string type data we did encoding which is named one hot encoding to transform the string type data to a binary value but unique. We could have used only simple numerical values but numerical values for 11 types of ethnicity will not give better results after applying algorithms. Now, one hot encoding is different from simple binary code because it represents a value as a unique value and there is no dependency or serial like numerical values. And for the other two datas, “Age_Mons” and “Qchat-10-Score” we did standard deviation that transformed the datas to the range of -3 to 3 which were previously numerical values without any range. These data preprocessing helped really well in algorithms performance of obtaining better results. For instance, Support Vector Machine gave accuracy results of 71-73% highest without the preprocessing of datas and when the values of “Ethnicity” were converted into numerical values only. But after applying preprocessing the algorithm “SVM” did a better job like the accuracy increased by 10%.So, that is why we used these preprocessing materials for all the algorithms.

Secondly, we experimented a total of six algorithms on our dataset. The algorithms were Random Forest, SVM, K-NN, Naive Bayes, Logistic Regression and Decision Tree classifier. Every one of them were classifier algorithms. From these six algorithms, two of the algorithms were named Logistic Regression and Decision Tree classifier which resulted in overfit of data. So, we dropped these two algorithms and kept the best performed algorithms in our experiments. From the rest 4 algorithms K-NN showed us the most promising result and took one the lowest possible time while calculating compared to others. Then the second best performed classifier algorithm was Random Forest Classifier. In algorithms we chose K-NN because closeness of features is the main basement of K-NN. A data point is classified by it based on how classification of its neighbours is done. It works better on data which are not complex and our dataset is complication free(noise free) as we pre-processed the data. Also, it works well in small dataset. K-NN normally calculates the euclidean distance of unknown data points from all the points to find the nearest neighbours values. Then by default it takes 5 neighbours values of euclidean distance to classify. We used 33 as neighbours value because that is the square root of our total data. Square root of total data is one of the best to take as the nearest neighbour at k value. And the 33 nearest values of euclidean distance did help to classify that a child has autism or not. So these are the reasons why K-NN performed really well in our model.

Thirdly, the second best result given algorithm is Random Forest and we used it because Random Forest shows no overfitting of data in result. By using multiple trees it reduces the risk of overfitting. Also, it takes less training time though it did the opposite in terms of timing in our case. Random forest method operates by constructing multiple decision trees when it is in the training phase. And from those decision trees, random forest takes the majority voted result and that is why it can give a higher accuracy in results.

After that, the Naive Bayes classifier has given the third good result which is 89% accuracy. Naive Bayes classifier works on the basis of contingent probability from the Theorem of Bayes. It is very easy to implement and simple. It is not sensitive to irrelevant features and that is what we found it would have given good results even if we had irrelevant features. It needs less training data and we chose it as we have less amount of data which is less than two thousand. Moreover, it is fast and it takes 1.53 seconds of time to train and calculate the results which is the lowest compared to other algorithms. But it gave less accuracy than the Random Forest and K-NN classifier because those algorithms could obtain better results as those have got some advantages in our dataset.

In the case of Support Vector Machine, it gave good results but the lowest among other algorithms. Support Vector Machine creates a decision boundary by dividing the data into two categories. In splitting the data, the boundary should be in a way where maximum space is found that separates the two classes. In our case the Support Vector Machine could not place the boundary in the best place where it separates the two classes in maximum space means the support vector and the hyperplane was not as far as possible, that is why it showed less better results than others.

The ROC curve in figure 6 demonstrates the true positive rate versus false positive rate curve of algorithms. Here we

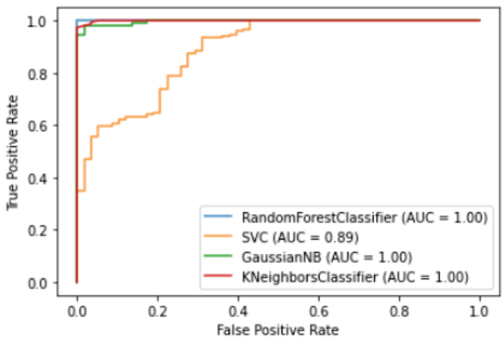

Fig. 6: ROC curve comparing Random Forest Classifier with KNN, Gaussian NB and SVM

can see that the Support Vector Machine has the lowest area under the curve and that is why the orange curve looks like this. And the other curves show that the GaussianNB is a little bit less accurate than the K-NN and Random forest classifier algorithms. So, It means the curve demonstrates Random Forest and K-NN is showing better performance than others.

IV. Conclusion and Future Work

Our proposed model allows us to have a more accurate result in terms of detecting autism at an early age. Questions which are provided to the parents to identify if their children are in danger or out of danger is set in a way to maintain their privacy. Using the dataset from Q-CHAT and AQ tools, our proposed model can predict using SVM, Random Forest, Naive Bayes and KNN with 83%, 93% ,89% and 98% accuracy in case of toddlers. Algorithms which are supervised are selected to run our dataset after preprocessing it. We used SVM, Random Forest, Naive Bayes and KNN to get our output more accurately. This outcome showed better performance compared to the others. Our result showed marginal performance in terms of accuracy (93%, 98%). The only limitation of our model is the lack of enough large data to train our model. The limitations on the characteristics of design or methodology that impacted or influenced the interpretation of our thesis research, the prime candidate is the lack of dataset. While implementing two algorithms we found that because of the lack of our dataset our result was overfitting. This modeling error occurred because a function corresponds too closely to a particular set of data. As a result, it failed to fit additional data and affected the accuracy of predicting future observations and we had to drop those two algorithms. In addition, it was not helpful even after preprocessing our data. Previously, many tried to detect autism at different ranges of age but we tried to emphasize on early ASD detection. The purpose of choosing such an age limitation in our model is to get as accurate results as possible. With the help of more accurate results and more data to train our model we can get tremendous work done. Many countries are struggling to detect autism as early as possible but with our model and the set of questionnaires we collected the problem can be solved effortlessly. Our objective for future work is to collect as much possible data from various sources and enhance the accuracy more. Moreover, we are thinking of building a user-friendly mobile application for end users based on our proposed model so that any individual can use the application to predict the early autism symptoms effortlessly so they can seek professional help if needed. Since diagnosing autism is quite a costly and lengthy process, it has been postponed for countless children. To conclude, with the help of our proposed model individuals can be guided at a very early age that will limit the situation from getting any worse and reduce costs associated with delayed diagnosis.

References

[1] J. Baio, "Prevalence of autism spectrum disorders: Autism and developmental disabilities monitoring network, 14 sites, united states, 2008. morbidity and mortality weekly report. surveillance summaries. volume 61, number 3." *Centers for Disease Control and Prevention*, 2012.

[2] S. E. Bryson, L. Zwaigenbaum, and W. Roberts, "The early detection of autism in clinical practice," *Paediatrics & child health*, vol. 9, no. 4, pp. 219–221, 2004.

[3] P. Mesa-Gresa, H. Gil-Gómez, J.-A. Lozano-Quilis, and J.-A. Gil-Gómez, "Effectiveness of virtual reality for children and adolescents with autism spectrum disorder: an evidence-based systematic review," *Sensors*, vol. 18, no. 8, p. 2486, 2018.

[4] F. Thabtah and D. Peebles, "A new machine learning model based on induction of rules for autism detection," *Health informatics journal*, vol. 26, no. 1, pp. 264–286, 2020.

[5] N. DONGES, "A complete guide to the random forest algorithm," *Built In*, vol. 16, 2019.

[6] E. Solutions and . Name, "Accuracy, precision, recall amp; f1 score: Interpretation of performance measures," Nov 2016. [Online]. Available: https://blog.exsilio.com/all/accuracy-precision-recall-f1