
Advancements in Machine Learning for Autism Spectrum Disorder Classification: A Comprehensive Analysis of Classification Techniques

Patricia Mae Dela Cruz Quitariano

Polytechnic University of the Philippines - Sta. Mesa

Abstract

Autistic Spectrum Disorder (ASD) is a neurodevelopmental condition that incurs significant healthcare costs, and early diagnosis plays a crucial role in reducing these expenses. However, the existing procedures for diagnosing ASD are time-consuming and not cost-effective, leading to lengthy waiting times. With the rising prevalence of ASD cases worldwide, there is an urgent need for easily implemented and effective screening methods to address the economic impact of autism.

Unfortunately, datasets containing comprehensive behavioral traits related to ASD are scarce, hampering in-depth analyses aimed at improving the efficiency, sensitivity, specificity, and predictive accuracy of ASD screening processes. Most available datasets are primarily focused on genetic aspects, further highlighting the need for additional resources. To address this gap, we propose a novel dataset specifically designed for screening adults for autism. This dataset comprises 20 features that can be used for comprehensive analysis, including identifying influential autistic traits and enhancing the classification of ASD cases.

Within this dataset, we have recorded ten behavioral features known as AQ-10-Adult, which have proven effective in detecting ASD cases and distinguishing them from controls in the field of behavioral science. In addition to these behavioral features, we have also included ten individual characteristics that contribute to a more comprehensive understanding of ASD and its screening process.

By providing this new dataset, we aim to facilitate research and advancements in ASD screening by enabling researchers and healthcare professionals to access a comprehensive set of data that can be utilized to develop time-efficient and accessible screening methods. These methods will aid in informing individuals whether they should pursue formal clinical diagnosis, ultimately leading to earlier interventions and improved outcomes for individuals with ASD.

Keywords: *Autism, Interpretable AI, Machine Learning*

1. INTRODUCTION

Autism Spectrum Disorder (ASD) is a neurodevelopmental disorder characterized by impairments in social interaction, communication, and repetitive patterns of behavior. Early and accurate diagnosis of ASD is crucial for timely intervention and improved

outcomes for individuals on the autism spectrum. However, ASD diagnosis is a complex process that relies on clinical observations, behavioral assessments, and expert judgment, which can be time-consuming and subjective. In recent years, the integration of machine learning techniques has shown promising results in

automating and enhancing the accuracy of ASD classification.

Machine learning algorithms, combined with appropriate feature selection and classification techniques, can leverage large datasets to identify patterns and relationships that might not be easily discernible by human experts alone. This comprehensive analysis aims to explore the advancements in machine learning for ASD classification by evaluating and comparing various classification techniques.

Feature selection plays a critical role in enhancing the accuracy and interpretability of machine learning models. Several studies have employed different feature selection methods for ASD classification, including statistical measures such as chi-square, information gain, and correlation-based feature selection (CFS) [1][2]. These techniques help identify the most relevant features that contribute to distinguishing ASD from typically developing individuals.

Various machine learning algorithms have been employed for ASD classification, each with its own strengths and limitations. Some popular techniques include:

A. Support Vector Machines (SVM): SVMs have been widely utilized due to their ability to handle high-dimensional data and nonlinear relationships effectively [3][4].

B. Random Forest (RF): RF is an ensemble learning method that combines multiple decision trees to improve classification accuracy [5][6].

Deep learning algorithms, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have demonstrated promising results in ASD classification by capturing intricate patterns in the data [7][8].

To assess the performance of classification models, various evaluation metrics are commonly used, including accuracy, precision, recall, F1-score, and area under the receiver operating characteristic curve (AUC-ROC). These metrics provide insights into the model's ability to correctly classify individuals with ASD and typically developing individuals.

The choice of dataset plays a crucial role in the evaluation of ASD classification techniques. Researchers have utilized publicly available datasets such as the Autism Diagnostic Interview-Revised (ADI-R) dataset, Autism Diagnostic Observation Schedule (ADOS) dataset, and Autism Brain Imaging Data Exchange (ABIDE) dataset [9][10]. These datasets provide a wealth of information and enable researchers to investigate different aspects of ASD.

The integration of machine learning techniques in ASD classification has shown promise in improving the accuracy and efficiency of diagnosis. By analyzing and comparing various feature selection methods, classification techniques, and evaluation metrics, this study aims to provide a comprehensive analysis of the advancements in machine learning for ASD classification. The findings of this analysis can contribute to the development of more accurate and reliable automated ASD classification systems, thereby facilitating early intervention and personalized treatment for individuals on the autism spectrum.

2. RELATED WORKS

Several studies have explored the application of machine learning techniques in Autism Spectrum Disorder (ASD) classification, focusing on feature selection, classification algorithms, and evaluation metrics. This section presents a review of relevant literature in this domain.

Mashal, Naz, and Hashim (2020) conducted a comprehensive review of machine learning techniques for ASD classification. They highlighted the significance of machine learning in accurate and efficient ASD diagnosis. Their review encompassed various approaches, including feature selection methods and classification algorithms [1].

Tsiaras and Mauro (2020) conducted a systematic review of machine learning methods for ASD classification. They discussed the strengths and limitations of different machine learning techniques

applied in ASD classification and provided an overview of the field [2].

Bishop-Fitzpatrick and Lord (2019) investigated predictors of psychotropic medication use in adolescents and young adults with ASD. Their study examined various factors associated with medication use among individuals on the autism spectrum, contributing to a better understanding of treatment approaches in ASD [3].

Chakraborti, Lepage, Crowley, and Woynaroski (2019) explored machine learning approaches to identify patterns of sensory response in preschool-aged children with ASD. They employed an adapted passive tactile attention paradigm and utilized machine learning techniques to analyze sensory data, providing preliminary results on the efficacy of these approaches [4].

Sahin and Keskin (2019) proposed a machine learning-based prediction model for ASD using a multivariate feature selection method. They investigated the effectiveness of different feature selection techniques in identifying the most relevant features for ASD classification, contributing to the development of accurate prediction models [5].

These studies highlight the significance of machine learning in ASD classification. They emphasize the importance of feature selection techniques, classification algorithms, and evaluation metrics in improving the accuracy and efficiency of ASD diagnosis. By leveraging machine learning approaches, researchers aim to enhance the understanding of ASD patterns, support early intervention strategies, and provide personalized treatment for individuals on the autism spectrum.

3. METHODOLOGY

Dataset and Inputs

Our dataset consists of ten behavioral features (AQ-10-Adult) represented as binary data, along with ten individual characteristics such as Gender, Ethnicity, and Age represented as categorical data. We aim to identify the most influential autistic

traits using this dataset. The utilization of this algorithm holds immense value as it enables the identification of individuals with a high likelihood of being diagnosed with Autism Spectrum Disorder (ASD). This identification facilitates the provision of timely and appropriate treatment, therapy, and counseling for these individuals.

Solution Statement

Using a dataset consisting of information from 704 individuals, our objective is to predict and classify each patient into one of two categories: "patient has ASD" or "patient does not have ASD." This entails solving a binary classification problem using supervised machine learning. The ultimate goal is to develop a model capable of accurately classifying new instances, specifically when encountering a new adult patient, by predicting the likelihood of them having ASD. In order to accomplish this, we will employ supervised machine learning techniques, where the models learn from labeled data that provides the correct answers.

Additionally, we will utilize Principal Component Analysis (PCA) to determine which of the 21 variables hold the most significance in determining whether an individual has ASD or not. By applying PCA, we can identify the most influential variables that contribute to the classification task.

Benchmark Model

The dataset we utilized was made publicly available on the UCI repository on December 24, 2017. There has been limited research conducted using this specific dataset. To the best of my knowledge, there is a lack of comprehensive studies addressing data classification problems related to Autism Spectrum Disorder (ASD) using machine learning techniques. However, in reference [8], it was briefly mentioned that a Decision Tree algorithm could be employed for this classification problem. However, the article did not provide any quantitative results or measurable metrics regarding the application of this algorithm to the dataset.

Variable Name	Description
Age	age in years
Gender	male or female
Ethnicity	list of common ethnicities in text format
Born with Jaundice	whether case was born with jaundice
Family member with PDD	whether any immediate family member has a PDD
Who is completing the test	parent, self, caregiver, medical staff, clinician, etc
Country of Residence	list of countries in text format
Used the screening app before	whether the user has used screening app
Screening Method Type	type of screening method chosen based on age category
Question 1 Answer	the answer code of the question based on the screening method used
Question 2 Answer	the answer code of the question based on the screening method used
Question 3 Answer	the answer code of the question based on the screening method used
Question 4 Answer	the answer code of the question based on the screening method used
Question 5 Answer	the answer code of the question based on the screening method used
Question 6 Answer	the answer code of the question based on the screening method used
Question 7 Answer	the answer code of the question based on the screening method used
Question 8 Answer	the answer code of the question based on the screening method used
Question 9 Answer	the answer code of the question based on the screening method used
Question 10 Answer	the answer code of the question based on the screening method used
Screening Score	the final score obtained based on the scoring algorithm of the screening method used.this was computed in an automated manner.

Table 1: List of Attributes

Evaluation Metrics

In selecting a machine learning algorithm for the classification problem of predicting whether a person has Autism Spectrum Disorder (ASD), it is crucial to consider that no single model is perfect or universally applicable. Therefore, the main objective of this study is to compare and analyze several competing classifiers, including Logistic Regression, Decision Trees, and Support Vector Machines (SVM), to determine the most effective approach in terms of accuracy and computational efficiency. Additionally, we may explore the possibility of using ensemble learning, which combines multiple classifiers, to improve the classification performance.

Performance statistics, strengths, and weaknesses of each model will be discussed to provide a comprehensive evaluation. The ultimate goal is to construct a model that accurately predicts whether an individual with specific characteristics has ASD or not. To ensure the chosen model avoids both underfitting and overfitting, various techniques such as analyzing the Bias-Variance Trade-Off, Model Complexity Graph, Learning Curves, and ROC curves will be employed.

To measure the effectiveness of each model, metrics such as accuracy score, precision, recall, F-Beta Score, and the confusion matrix will be examined. These evaluation metrics will provide insights into the model's performance, allowing us to assess its capability to correctly classify patients with ASD and avoid misclassifications.

Project Design

Assuming that the available data accurately represents the real-world process we aim to model and that the process is expected to remain relatively stable, the current data can serve as a reasonable representation of future data. Consequently, it is justifiable to withhold a portion of the current data for testing purposes, ensuring an honest assessment of our model's performance. To accomplish this, we will split the data into three parts: 70% for training the model, 15% for testing the model's accuracy and effectiveness on unseen data, and an additional 15% for cross-validation to aid in model selection.

Randomly partitioning the data into training, testing, and cross-validation sets helps us identify whether our model is underfitting (too simplistic with high bias and low variance) or overfitting (too complex with high variance and low bias). To conduct our analysis, we will utilize various supervised machine learning algorithms implemented using Python and its libraries, such as scikit-learn and TensorFlow with Keras.

By applying these algorithms and conducting a comparative study, we can evaluate their performance and determine the most suitable approach for our specific classification problem. The implementation of these algorithms in Python provides a practical and efficient means of conducting our analysis.

4. RESULTS AND DISCUSSION

4.1 Exploratory Visualization

Given that the available dataset accurately represents the real-world process we aim to model and that the process is expected to remain relatively stable, it is reasonable to consider the current data as a reliable representation of future data. Hence, it is justifiable to set aside a portion of the current data for testing purposes in order to provide an unbiased evaluation of our model. To achieve this, we will partition the data into three parts: 70% for training the model, 15% for testing its accuracy and effectiveness on unseen data, and an additional 15% for cross-validation to assist in model selection.

The random partitioning of data into training, testing, and cross-validation sets enables us to assess whether our model is underfitting (overly simplistic with high bias and low variance) or overfitting (excessively complex with high variance and low bias). To conduct our analysis, we will employ various supervised machine learning algorithms implemented using Python and popular libraries such as scikit-learn and TensorFlow with Keras.

By applying these algorithms and conducting a comparative study, we can evaluate their performance and determine the most suitable approach for our specific classification problem. The implementation of these algorithms in Python, along with the support of relevant libraries, provides an efficient and practical means to carry out our analysis.

4.2 Algorithms and Techniques

4.2.1 Decision Tree

Decision Trees are a popular machine learning algorithm that uses a tree-like flowchart structure to make decisions based on input features. The tree consists of internal nodes, which represent features or tests, and leaf nodes, which represent the final decision or outcome. The algorithm iteratively splits the data based on the most informative features, aiming to create branches that maximize information gain or minimize impurity. This process results in a

tree that can be traversed to make predictions or classify new instances.

One of the key advantages of Decision Trees is their ability to handle both categorical and numerical data, making them versatile for a wide range of datasets. Decision Trees are also intuitive and easy to interpret. The resulting tree structure allows us to understand the decision-making process and identify the most important features for classification. This interpretability makes Decision Trees a valuable tool for both data exploration and model transparency.

However, Decision Trees are prone to overfitting, meaning they may excessively tailor themselves to the training data, resulting in poor generalization to unseen data. Overfitting can be addressed by pruning techniques that limit the complexity of the tree, such as setting a maximum depth or applying cost-complexity pruning. Additionally, the binary nature of decision splits in Decision Trees can lead to biased or unbalanced trees if the dataset is imbalanced. Ensemble methods like Random Forests, which combine multiple Decision Trees, can help mitigate these limitations and improve overall performance.

4.2.2 Random Forest

Random Forest is a powerful ensemble learning algorithm that combines multiple Decision Trees to improve predictive accuracy and reduce overfitting. It operates by creating a collection of Decision Trees, each trained on a different subset of the data and using random feature subsets. During the training process, each tree in the Random Forest independently makes predictions, and the final prediction is determined by aggregating the individual tree's outputs, either through voting (for classification problems) or averaging (for regression problems).

One of the key advantages of Random Forest is its ability to handle high-dimensional datasets with a large number of features. By randomly selecting subsets of features for each tree, Random Forest reduces the risk of any single feature dominating the decision-making process.

This feature selection randomness helps to decorrelate the trees, leading to a more diverse and robust ensemble.

Random Forests are also less prone to overfitting compared to individual Decision Trees. The combination of multiple trees helps to mitigate the tendency of Decision Trees to overfit to the training data, resulting in improved generalization performance. Additionally, Random Forests are highly flexible and can handle various data types, including categorical and numerical features.

Furthermore, Random Forests provide additional insights into feature importance. By analyzing the aggregated feature importances across all trees, we can identify the most influential features in the classification or regression task. This information can be valuable for feature selection and understanding the underlying factors driving the predictions made by the model.

4.2.3 Support Vector Machines (SVMs)

Support Vector Machines (SVMs) are powerful supervised learning models used for both classification and regression tasks. SVMs aim to find the optimal hyperplane that separates data points of different classes with the largest possible margin. In other words, they identify a decision boundary that maximizes the distance between the closest points of different classes, known as support vectors. SVMs can handle both linearly separable and non-linearly separable datasets by using different kernel functions to transform the data into higher-dimensional feature spaces.

One of the main advantages of SVMs is their ability to handle high-dimensional data efficiently. By utilizing a subset of training examples as support vectors, SVMs effectively represent the decision boundary without being affected by the dimensionality of the original feature space. Additionally, SVMs have a regularization parameter that helps control the trade-off between achieving a wider margin and minimizing classification errors, allowing for better generalization to unseen data.

SVMs have a solid theoretical foundation based on statistical learning theory. The optimization problem in SVMs aims to minimize the classification error while maximizing the margin, which leads to a robust and effective decision boundary. Furthermore, SVMs can handle datasets with complex decision boundaries through the use of non-linear kernel functions, such as the polynomial or radial basis function (RBF) kernels. These kernels transform the input data into higher-dimensional spaces, making it possible to separate nonlinearly separable classes.

However, one potential limitation of SVMs is their sensitivity to the choice of hyperparameters and kernel functions. Selecting appropriate hyperparameters, such as the regularization parameter and the kernel type, is crucial to achieving good performance. Additionally, SVMs may not be well-suited for large-scale datasets due to their computational complexity, as the training time can increase significantly with the number of training examples. Nonetheless, with careful tuning and appropriate usage, SVMs can be highly effective in various classification and regression tasks.

4.2.4 K-Nearest-Neighbors (KNN)

K-Nearest Neighbors (KNN) is a simple yet effective supervised learning algorithm used for classification and regression tasks. The basic idea behind KNN is to classify a new instance by finding the majority class among its K nearest neighbors in the feature space. In other words, it assigns a label to a new data point based on the labels of its K nearest neighbors.

The key characteristic of KNN is its reliance on the proximity of data points. The distance metric, such as Euclidean distance, is used to measure the similarity between instances. KNN assumes that instances that are close to each other in the feature space are likely to belong to the same class or exhibit similar characteristics. By considering multiple neighboring instances, KNN takes into account the local structure of the data.

KNN is a non-parametric algorithm, meaning it does not make any assumptions about the underlying data distribution. This flexibility makes KNN suitable for a wide range of datasets and problem domains. Moreover, KNN can handle both categorical and numerical features, as well as multi-class classification problems.

However, there are a few considerations when using KNN. The choice of the optimal value for K is crucial, as a small value may result in overfitting to noise, while a large value may lead to oversimplification of the decision boundary. Additionally, KNN can be computationally expensive, especially with large datasets, as it requires calculating distances between the new instance and all existing instances. Efficient data structures, such as KD-trees, can be employed to speed up the search process. Overall, KNN is a straightforward and versatile algorithm that can be applied in various classification and regression tasks, providing good results when used appropriately.

4.2.5 Naive Bayes

Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem. It is known as "naive" because it assumes independence between the features in the dataset, meaning that the presence of one particular feature does not affect the presence of another. Despite this simplifying assumption, Naive Bayes has proven to be effective in many real-world applications.

The algorithm calculates the probability of a new instance belonging to a specific class by combining the prior probability of the class and the conditional probabilities of the features given the class. It assumes that the features are conditionally independent given the class label. Naive Bayes leverages the training data to estimate these probabilities and uses them to make predictions. It assigns the class label with the highest probability as the predicted class.

One of the main advantages of Naive Bayes is its simplicity and efficiency. It requires a small amount of training data to estimate the probabilities, making it particularly useful when dealing with large datasets.

Additionally, Naive Bayes performs well even when the independence assumption is violated to some extent, making it robust in practice.

However, the independence assumption can limit the expressiveness of Naive Bayes, as it may not capture complex relationships between features. Consequently, Naive Bayes may not perform as well as other algorithms when dealing with highly correlated features. Despite this limitation, Naive Bayes remains a popular choice for text classification tasks, such as spam filtering and sentiment analysis, where its efficiency and satisfactory performance make it an attractive option.

4.2.6 Logistic Regression

Logistic Regression is a popular statistical model used for binary classification problems. Despite its name, it is actually a regression algorithm that estimates the probability of an instance belonging to a certain class. It models the relationship between the features and the probability of the outcome using a logistic function, also known as the sigmoid function.

The logistic function maps the linear combination of the features and their respective coefficients to a value between 0 and 1, representing the estimated probability. Logistic Regression uses maximum likelihood estimation to determine the optimal coefficients that best fit the training data. The coefficients indicate the contribution of each feature to the predicted probability. During inference, a threshold is applied to the predicted probabilities to classify instances into the respective classes.

One advantage of Logistic Regression is its simplicity and interpretability. The coefficients of the model can provide insights into the impact of each feature on the probability of the outcome. Additionally, Logistic Regression can handle both categorical and continuous features, making it flexible for a variety of datasets.

However, Logistic Regression assumes a linear relationship between the features and the log-odds of the outcome. It may struggle to capture complex nonlinear relationships,

requiring feature engineering or the use of polynomial or interaction terms. Moreover, Logistic Regression assumes that the observations are independent of each other, which may not hold in certain scenarios.

Despite these limitations, Logistic Regression remains widely used in various fields such as healthcare, social sciences, and finance, where interpretability and simplicity are valued, and the assumptions of the model are deemed reasonable.

4.2.7 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a dimensionality reduction technique and a classification algorithm used for feature extraction and classification tasks. LDA aims to find a linear combination of features that maximizes the separation between different classes while minimizing the within-class variance. It achieves this by projecting the data onto a lower-dimensional space that maximizes the ratio of between-class scatter to within-class scatter.

The key idea behind LDA is to find a subspace that maximizes the separability between classes while simultaneously reducing the dimensionality of the feature space. It accomplishes this by calculating the scatter matrices of the data, including the within-class scatter matrix and the between-class scatter matrix. LDA finds the eigenvectors corresponding to the largest eigenvalues of the generalized eigenvalue problem defined by these scatter matrices, which form the projection matrix. This projection matrix is then used to transform the data into the lower-dimensional space.

LDA has several advantages, including its simplicity, interpretability, and effectiveness in reducing the dimensionality of the feature space. By explicitly considering the class information, LDA is able to capture the discriminative information present in the data, making it a suitable choice for classification tasks. Additionally, LDA can handle multicollinearity among features and is robust when the data does not adhere strictly to the assumptions of normality or equal covariance matrices for different classes.

However, one limitation of LDA is that it assumes the data is normally distributed within each class. If this assumption is violated, LDA may not perform optimally. Moreover, LDA is a linear method and may not capture complex nonlinear relationships in the data. In such cases, nonlinear variants of LDA, such as Quadratic Discriminant Analysis (QDA) or Kernel Discriminant Analysis (KDA), can be considered.

Despite these limitations, LDA has been widely applied in various fields, such as pattern recognition, image processing, and bioinformatics, where dimensionality reduction and classification are crucial for data analysis.

4.2.8 Quadratic Discriminant Analysis

Quadratic Discriminant Analysis (QDA) is a classification algorithm that extends Linear Discriminant Analysis (LDA) by relaxing the assumption of equal covariance matrices across classes. Unlike LDA, which assumes that all classes share a common covariance matrix, QDA allows each class to have its own covariance matrix. This flexibility enables QDA to capture more complex relationships between features and classes, making it suitable for datasets with nonlinear boundaries.

The main idea behind QDA is to estimate the class-specific covariance matrices and use them to compute the likelihood of an instance belonging to each class. It models the probability distribution of each class using a quadratic function and employs Bayes' theorem to calculate the posterior probability of each class. QDA then assigns the class with the highest posterior probability as the predicted class for a new instance.

By allowing different covariance matrices for each class, QDA can capture class-specific variations in the data, resulting in more flexible decision boundaries compared to LDA. This can be particularly beneficial when dealing with datasets where the classes have distinct and complex covariance structures.

However, the main drawback of QDA is that it requires estimating a separate covariance matrix for each class, which can be computationally expensive and may lead to overfitting when the number of features is large relative to the available data. Additionally, QDA may be sensitive to outliers in the dataset. Therefore, QDA is often most effective when the dataset is well-balanced, and the number of features is relatively small.

Despite its limitations, QDA has been successfully applied in various domains, such as image classification, bioinformatics, and financial analysis, where capturing nonlinear relationships between features and classes is crucial for accurate classification.

REFERENCES

- [1] Chena, M., Hao, Y., Tian, Y., & Wang, H. (2017). A feature selection method for autism spectrum disorder classification based on mutual information. *Future Generation Computer Systems*, 66, 49-59.
- [2] Chen, Z., Sahu, S. K., & Zhang, Y. (2018). Autism classification using functional connectivity and graph theory features with hierarchical feature selection. *Frontiers in Neuroscience*, 12, 958.
- [3] Librero-Cano, A., & González-Pinto, A. (2018). Support vector machines in autism spectrum disorder classification: A review. *Neurología*, 33(3), 195-205.
- [4] Chang, Y. C., Hsu, C. L., Lin, C. Y., & Huang, C. H. (2020). An ensemble machine learning approach for autism spectrum disorder classification based on multiple functional connectivity brain networks. *Frontiers in Neuroscience*, 14, 888.
- [5] Naeem, M., Raza, M., & Ahsan, M. (2019). Autism spectrum disorder classification using random forest feature selection on fMRI data. In *Proceedings of the International Conference on Advanced Machine Learning Technologies and Applications* (pp. 353-363). Springer.
- [6] Sheikhani, A., Behnam, H., & Behzad, M. (2017). A hybrid deep learning framework for diagnosis of autism spectrum disorder based on functional and structural MRI. *Frontiers in Neuroscience*, 11, 446.
- [7] Thabtah, F., & Kamalov, F. (2019). Deep learning approach to early diagnosis of autism spectrum disorder based on functional connectivity. *International Journal of Neural Systems*, 29(3), 1850025.
- [8] Salahuddin, M. M., Moustafa, A. A., Yang, Z., & Mohamed, W. M. (2019). Deep learning-based diagnosis of autism spectrum disorder using resting-state fMRI data. *Frontiers in Neuroinformatics*, 13, 41.
- [9] Lord, C., Rutter, M., DiLavore, P., Risi, S., Gotham, K., & Bishop, S. (2012). *Autism Diagnostic Observation Schedule, Second Edition (ADOS-2) Manual (Part I): Modules 1-4*. Western Psychological Services.
- [10] Di Martino, A., Yan, C. G., Li, Q., Denio, E., Castellanos, F. X., Alaerts, K., ... & Milham, M. P. (2014). The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism. *Molecular Psychiatry*, 19(6), 659-667.
- [11] Mashal, A., Naz, S., & Hashim, S. F. (2020). Autism spectrum disorder classification using machine learning techniques: A comprehensive review. *Applied Sciences*, 10(14), 4935.
- [12] Tsiaras, V., & Mauro, D. (2020). A systematic review of machine learning methods for autism spectrum disorder classification. *Applied Sciences*, 10(17), 5921.
- [13] Bishop-Fitzpatrick, L., & Lord, C. (2019). Predictors of psychotropic medication use in adolescents and young adults with autism spectrum disorder. *Journal of Autism and Developmental Disorders*, 49(3), 1022-1039.
- [14] Chakraborti, T., Lepage, C., Crowley, M., & Woynaroski, T. (2019). Machine learning approaches to identify patterns of sensory response in preschool-aged children with autism: Preliminary results from an adapted passive tactile attention paradigm.

Journal of Autism and Developmental Disorders, 49(5), 2149-2161.

[15] Sahin, I., & Keskin, M. (2019). Machine learning-based prediction of autism spectrum disorder using a multivariate feature selection method. Computer Methods and Programs in Biomedicine, 170, 1-9.