Bachelor Thesis in Computer Science
September 2023

Comparison of Machine Learning Algorithms on Identifying Autism Spectrum Disorder

Naga Sai Gayathri Aravapalli
Manoj Kumar Palegar

Faculty of Faculty, Blekinge Institute of Technology, 371 79 Karlskrona, Sweden

This thesis is submitted to the Faculty of Faculty at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Bachelor Thesis in Computer Science. The thesis is equivalent to Weeks weeks of full-time studies.

The authors declare that they are the sole authors of this thesis and that they have not used any sources other than those listed in the bibliography and identified as references. They further declare that they have not submitted this thesis at any other institution to obtain a degree.

Contact Information:
Author(s):
Naga Sai Gayathri Aravapalli
E-mail: naaa22@student.bth.se

Manoj Kumar Palegar
E-mail: mapg22@student.bth.se

University advisor:
Ph.D. Shahrooz Abghari
Department of Computer Science

Faculty of Faculty
Blekinge Institute of Technology
SE–371 79 Karlskrona, Sweden

Internet : www.bth.se
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57

Abstract

Background: Autism Spectrum Disorder (ASD) is a complex neurodevelopmental disorder that affects social communication, behavior, and cognitive development. Patients with autism have a variety of difficulties, such as sensory impairments, attention issues, learning disabilities, mental health issues like anxiety and depression, as well as motor and learning issues. The World Health Organization (WHO) estimates that one in 100 children have ASD. Although ASD cannot be completely treated, early identification of its symptoms might lessen its impact. Early identification of ASD can significantly improve the outcome of interventions and therapies. So, it is important to identify the disorder early. Machine learning algorithms can help in predicting ASD. In this thesis, Support Vector Machine (SVM) and Random Forest (RF) are the algorithms used to predict ASD.
Objectives: The main objective of this thesis is to build and train the models using machine learning(ML) algorithms with the default parameters and with the hyperparameter tuning and find out the most accurate model based on the comparison of two experiments to predict whether a person is suffering from ASD or not.
Methods: Experimentation is the method chosen to answer the research questions. Experimentation helped in finding out the most accurate model to predict ASD. Experimentation is followed by data preparation with splitting of data and by applying feature selection to the dataset. After the experimentation followed by two experiments, the models were trained to find the performance metrics with the default parameters, and the models were trained to find the performance with the hyperparameter tuning. Based on the comparison, the most accurate model was applied to predict ASD.
Results: In this thesis, we have chosen two algorithms SVM and RF algorithms to train the models. Upon experimentation and training of the models using algorithms with hyperparameter tuning. SVM obtained the highest accuracy score and f1 scores for test data are 96% and 97% compared to other model RF which helps in predicting ASD.
Conclusions: The models were trained using two ML algorithms SVM and RF and conducted two experiments, in experiment-1 the models were trained using default parameters and obtained accuracy, f1 scores for the test data, and in experiment-2 the models were trained using hyper-parameter tuning and obtained the performance metrics such as accuracy and f1 score for the test data. By comparing the performance metrics, we came to the conclusion that SVM is the most accurate algorithm for predicting ASD.

Keywords: Autism Spectrum Disorder(ASD), Classification, Data pre-processing, Feature selection, Machine learning algorithms, Random Forest Classifier, Support Vector Classifier.

Acknowledgments

We would like to express our sincere gratitude to our supervisor Shahrooz Abghari for his continuous support and guidance throughout the thesis. We are thankful for his time for reviewing our research and providing valuable suggestions for our thesis report. We are also thankful to our family and friends for their support.

Authors
Naga Sai Gayathri Aravapalli
Manoj Kumar Palegar

Contents

List of Figures

List of Tables

List of Acronyms

ASD Autism Spectrum Disorder

KNN K-Nearest Neighbors

LR Logistic Regression

ML Machine Learning

NB Naive Bayes

RF Random Forest

SVM Support Vector Machine

SVC Support Vector Classifier

SD Standard Deviation

Chapter 1

Introduction

A neurological disorder called ASD affects a person's ability to interact socially, communicate, and learn. Although autism can be identified at any age, the bulk of its symptoms start to show around the age of two and worsen over time. Patients with autism have a variety of difficulties, such as sensory impairments, attention issues, learning disabilities, mental health issues like anxiety and depression, as well as motor and learning issues [11].

On a global level, the incidence of autism is currently experiencing a significant and notable rise. The WHO estimates that one in 100 children have ASD [34]. While some people with this condition are able to live independently, others need constant care and support.

Although ASD cannot be completely treated, early identification of its symptoms might lessen its impact [34]. There is some optimism for the early identification of ASD based on a range of physical and physiological symptoms because of the application of ML in the precise prediction and detection of many disorders. Due to the symptoms of other mental health conditions being similar to those of ASD, diagnosing and analyzing ASD can be difficult and may result in false positive results [10]. However, if the characteristics of an ML-based model could be utilized to explain why it predicts ASD, medical professionals might be better equipped to make an informed choice during the early diagnosis phase.

This served as the motivating force behind our effort because early ASD detection would improve the quality of life for patients and their families by minimizing the consequences of the symptoms and giving quality, responsible treatment. One of the most significant uses of data mining, which is helpful for decision-making in a wide range of everyday issues is classification algorithms. With the help of various categorization approaches, this suggested system aims to predict ASD in its early phases. ML algorithms can identify the early symptoms of ASD, which may not be very easily detected by human evaluation. ML algorithms can handle large datasets, identify patterns, and can generate predictive models that can be used for future diagnosis.

We are using SVM and RF algorithms due to their high accuracy, ability to handle large datasets, and interpretability [19]. SVM is a powerful algorithm for classification tasks and has the capability of performing binary and multi-class classification on a dataset. It works well in high-dimensional spaces and can handle complex datasets with non-linear boundaries. RF is a versatile algorithm that can handle both classification and regression tasks. It works well with large datasets, noisy data, and data with non-linear relationships between the features and the target variable. It is important to consider the specific characteristics, target variable, and the desired level of accuracy. The characteristics such as datatype and feature selection and also important to consider the target variable, whether the data is balanced or not, and the performance metrics.

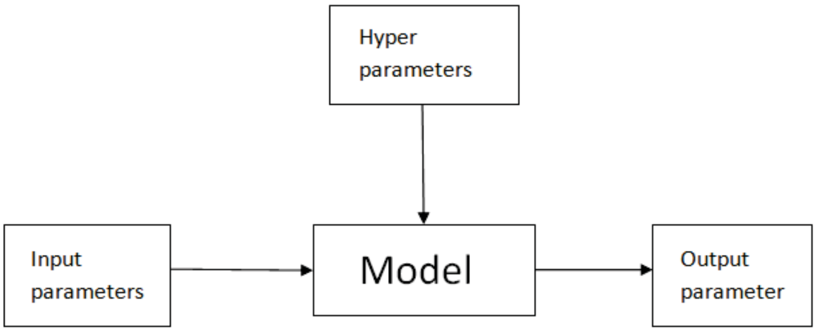

Figure 1.1: The input and output parameters of the model

In the data and the ML algorithms we have taken, the input parameters are the variables or attributes that the model uses as input to make predictions. Here, the considered variables are selected features from the data that are highly dependent on the response to produce output parameters. The output parameter is the target variable (class label) that the model is trained to predict. Here, the produced output is ASD traits from the data. Each ML algorithm can have one to various hyper-parameters. In the case of default values, the performance of the built models can be sub-optimal. Tuning hyperparameters allows the ML algorithms to use the most suitable values for their parameter based on the training set. The considered models to identify ASD are SVM and RF and the considered hyperparameters for SVM are 'C', and 'Kernel' and the hyperparameters for RF are 'n_estimators', 'max_depth', 'min_samples_split'. This process can be seen in the above figure.

1.1 Research Gap:

The similarity in symptoms between ASD and other mental health conditions can complicate the diagnosis process, potentially leading to false positive outcomes. However, using an ML-based model's features to explain ASD identification could enable healthcare providers to make better choices during the first diagnosis phase. It serves as a motivating force for this research [29].

1.2 Aim and Objectives

1.2.1 Aim:

The main aim of the thesis is to identify the best-performing algorithm among the selected ML algorithms, SVM and RF, in identifying ASD. To identify the best algorithm, a comparison of performance metrics is done between the selected ML algorithms.

1.2.2 Objectives:

1. Applying data preprocessing techniques on the selected data. The dataset contains numerical and categorical features where we performed data processing to transform the categorical data into ordinal to facilitate the task of ML algorithms.

2. Splitting the data into train and test datasets after data pre-processing. This division is essential to train the models on a subset of a data while evaluating their performance on another subset, ensuring that our models can identify well on unseen data.

3. Perform feature selection on trained data. Feature selection is used to select the features which are highly dependent on the response.

4. Performing two experiments.

    - Experiment-1 training and testing the data with default parameters.
    - Experiment-2 training and testing the data with hyper-parameter tuning.

5. Evaluating the performance of the build models in each experiment.

1.3 Research Questions

RQ1. How well do the chosen algorithms (SVM and RF) recognize ASD?

Motivation: The reason for studying this research question is to evaluate the effectiveness and limitations of ML algorithms in precisely identifying ASD. The results of these studies can help with enhancing the interpretability of the results and potentially improving the accuracy, resulting in better medical outcomes for people who have ASD.

RQ2. To what extent will parameter tuning affect the performance of the selected algorithms, and how?

Motivation: The reason for studying this research question is to explain how tuning the parameters can affect the performance of the RF and SVM algorithms. Tuning the parameters may lead to improvements in performing metrics such as accuracy and F1 score, which are important metrics. Hyperparameters of the classifier will influence the choice of features that may lead to producing good accuracy and F1 score.

1.4 Ethical, societal and sustainability aspects

The study makes use of a publicly accessible dataset from Kaggle, which doesn't affect the privacy of any users. This research takes measures to ensure privacy is not breached, unlike many others that compromise user privacy when examining consumer behavior.

With the creation of ML-based solutions and the enhanced accuracy and reliability of these studies can assist doctors in their diagnosis of ASD. By ensuring that individuals with ASD receive the treatment they require to survive, these characteristics can have a positive impact on society.

The use of ML algorithms for ASD identification may require significant computational resources, which can have environmental implications. To address this, it is important to explore energy-efficient computing solutions and to ensure that data centers and other computing facilities are designed with sustainability in mind. This can be achieved, with the help of feature selection and parameter tuning. It requires measuring the timely execution of the selected algorithms in a few scenarios, e.g., building a model using identified features from the feature selection process and the default parameters of each algorithm, then building a model using identified features and with parameter tuning.

1.4.1 Outline of Thesis

The outline of the thesis section describes the whole structure of the thesis. The first chapter describes the introduction of the thesis. Aim, objectives, and research questions are the few sections of this chapter. The aim and objectives section describes the goal of the thesis and the research questions are the questions that we would like to research. The next chapter is about the background of the thesis. This chapter includes several topics we used in this thesis and gives detailed information about the ML and ML algorithms and performance metrics that we calculated in this thesis. The next chapter describes the related work of the thesis. This chapter explains the related work of the ASD using ML algorithms. And the next chapter describes about methodology of the thesis and about how we achieved the aim and the process to achieve the aim. Next chapter describes the results of the problem we worked on. The next chapter is Discussions, this chapter discusses about the research questions and the analysis of the method. The next chapter discusses about the conclusions and future work. This chapter includes the main conclusions of the research work and about the future works of the thesis.

Chapter 2

Background

2.1 Autism Spectrum Disorder

Communication, behavior, and social interaction are affected by the complex neurodevelopment disease known as ASD. The term spectrum describes the broad variety of symptoms and degree of severity that may occur in people with ASD. The identification of ASD is often made in childhood and it rarely happens in adults. Observation, standardized test are the common techniques used in diagnosis process. Some of the common features of ASD are difficulty in understanding non verbal communication, difficulty in making eye contact.

ML is an approach for identifying ASD that involves training algorithms on the data in order to find characteristics and patterns that are specific to ASD. By using ML techniques we can increase the diagnosis speed and accuracy. The dataset of people with and without ASD is required for generating a machine learning model for ASD diagnosis. The dataset should include symptom profiles, medical history, and diagnostic standards. It is very important to label the dataset and based on the gold standard diagnosis each person should classify suffering or not suffering from ASD.

2.2 Machine Learning

Machine Learning is a branch of Artificial intelligence that enables the machine to automatically learn from the data, improve performance from past experiences, and make predictions [9]. It is classified into three types: supervised, unsupervised, semi-supervised, and reinforcement learning. It is useful for a wide range of applications and to predict the disorders which we used in this thesis [18].

Supervised learning: Supervised machine learning is based on supervision. That is, in supervised learning, the algorithm learns the relationship among the features with the class labels, and based on the training, the machine predicts the output [22].
Classification: The output variable is categorical, such as categorizing photos or predicting whether or not a customer will buy a product.
Regression: The output variable is continuous, such as forecasting the price of a house based on its features or product demand.

Unsupervised learning: It is a sort of ML in which the algorithm is trained on a dataset that was not previously labeled. Without any specified output variable to

forecast, the algorithm learns to detect patterns and relationships in the input data. The algorithm tries to group similar instances together or discover attributes that describe the data's underlying structure [3].
Clustering: The algorithm clusters together similar instances based on their characteristics.
Dimensionality Reduction: The algorithm minimizes the number of features in the data while retaining as much information as possible.

Semi-supervised learning: Combining aspects of supervised and unsupervised learning is known as semi-supervised learning. A small amount of labeled data and a large amount of unlabeled data are used to train a model in this process. The unlabeled data can help the model develop better representations and generalize well to new, unseen data [13].

Reinforcement learning: Reinforcement learning is a feedback-based learning process, in which an AI agent automatically explores its surroundings by hit and trial, taking action, learning from experiences and improving its performances [35].

2.3 Machine learning algorithms

This section describes the Ml algorithms used in this thesis:

Support Vector Machine
SVM is a classification and regression prediction tool that automatically prevents over-fitting to the data while maximizing predicted accuracy. SVMs were initially created to address the classification issue, but more recently they have been further developed to address regression issues. However, SVM can be sensitive to the choice of kernel function and hyperparameters [16].

Random Forest
A method for classification, regression, and anomaly detection is called Random Forest. The method functions by building a group of decision trees and combining their predictions. A random subset of the data and characteristics are used to train each decision tree. RF is resilient to overfitting and can handle categorical and continuous data. RF is widely used in applications including stock price prediction, text classification, and picture classification [4].

2.4 Experimental Environment

Python
Python is a widely used language in ML due to its simplicity, easy-to-learn syntax, and large community support. It offers a vast ecosystem of libraries and frameworks such as Scikit-learn, TensorFlow, Keras, and PyTorch. These tools help to enable quickly developing and deploying ML models. Python also has libraries like Pandas and NumPy, which simplify data preprocessing and manipulation. In this thesis,

python programming is used to develop models for various ML algorithms such as SVM and RF algorithms. Additionally, the following libraries are used to manipulate the data in the thesis [24].

Pandas:
Pandas is a widely used library in the field of ML due to its data manipulation, data handling, and data visualization capabilities. It provides data structures such as data frames and series that are essential for ML applications. It simplifies the data preprocessing and cleaning process, which is a crucial step in building accurate and reliable ML models. The library integrates seamlessly with other ML libraries, such as Scikit-learn, making it an essential tool for building end-to-end ML pipelines. In this thesis, pandas are used to read the data and manipulate the rows and columns [20].

Numpy:
NumPy is a fundamental library for scientific computing in Python, and it's extensively used in ML. It offers powerful tools for linear algebra, numerical operations, and array manipulation, making it a go-to choice for handling large datasets. NumPy provides an efficient N-dimensional array object, which is used for representing numerical data in machine learning models. Its fast and reliable performance makes it a crucial component of many popular machine-learning libraries like sci-kit-learn. In this thesis, NumPy is used to manipulate the dataset [7].

Scikit-learn:
Scikit-learn is a popular Python library for ML that offers simple and efficient tools for data mining and data analysis. It provides a wide range of machine-learning algorithms, including classification, regression, clustering, and dimensionality reduction. Scikit-learn is built on top of NumPy and it integrates well with other Python libraries like pandas and seaborn. In this thesis, the Sklearn module is used to import all the algorithms, preprocessing, and model selection techniques [12].

train_test_split: 'train_test_split' is a function in the scikit-learn library that is commonly used to split a dataset into a training set and a testing set. It takes an input dataset and a specified test size as input, and randomly splits the dataset into two separate subsets: a training set and a testing set. The training set is used to train a ML model, while the testing set is used to evaluate the performance of the model on unseen data. By splitting the data into training and testing sets, we can avoid overfitting and get a more accurate estimate of how well the model will perform on new, unseen data [6].

Stratified kfold:
Stratified k-fold is a cross-validation technique used in ML to ensure that the training and validation sets have proportional representation of the target variable classes. In this technique, the data is divided into k-folds, and each fold is split into training and validation sets while maintaining the proportion of classes in each set. This helps in reducing the bias that may arise from imbalanced class distributions and ensures that the model is trained and evaluated on a representative sample of the data. In this thesis, the Stratified kfold technique is used for the Ml models for better accuracy [25].

GridSearchcv:
GridSearchCV is a ML technique used for hyperparameter tuning. It involves searching for the best combination of hyperparameters for a given model. Hyperparameters are parameters that are set prior to training and cannot be learned from the data. GridSearchCV generates all possible combinations of hyperparameters, trains and evaluates a model for each combination using cross-validation, and returns the combination of hyperparameters that results in the best performance according to a specified evaluation metric. In this thesis, the GridSearchcv technique is used for Ml models to tune the hyperparameters for better accuracy [23].

2.5 Performance Metrics:

After training a ML model, it's important to evaluate its performance before making predictions. There are several performance metrics used in ML for this purpose. For classification problems, some commonly used metrics include the confusion matrix, classification report, classification accuracy, ROC curve, and AUC curve. Precision and recall are also important metrics for sorting algorithms. Among these performance metrics, Accuracy and F1 scores are calculated to provide insights into how well the model is performing and can be used to compare different models or fine-tune the parameters of the existing model for better performance [15]. Accuracy can determine the effectiveness of classification algorithms. The following Accuracy and F1 scores are calculated in this thesis.

Accuracy:
Accuracy is a common evaluation metric used in classification problems. It measures the proportion of correctly predicted outcomes to the total number of outcomes. It is calculated by dividing the number of correct predictions by the total number of predictions made [21]. Mathematically, it is defined as:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

Precision:
Precision is related to the accuracy of predictions made by the model. Precision calculates the proportion of the model's predictions that are accurate [27].

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

Recall:
Recall is also related to the accuracy of the predictions made by the model. Recall calculates the proportion of the relevant data points that were correctly identified by the model [27].

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

F1 score:
F1 score is an evaluation metric that measures the accuracy of the model. It is a combination of the precision and recall scores of a model. The accuracy statistic counts the number of times a model accurately predicted the entire data set. This can be a reliable metric only if the dataset is class-balanced; that is, each class of the dataset has the same number of samples [28].

$$\text{F1 score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

Chapter 3

Related Work

This chapter describes the overview of the previous work on predicting ASD. The following five papers helps us in understanding ASD and about how ML algorithms helps in predicting the disease.

Vaishali R, Sasikala R [30] researched on a ML based approach to classify Autism with optimum behaviour sets. In this study, a dataset for diagnosing ASD is used, and it consists of 21 features that were obtained from the UCI ML Repository. The inquiry uses a wrapper technique that draws inspiration from binary firefly optimization for swarm intelligence. According to the study's hypothesis, using small feature subsets could help a ML model improve classification accuracy. The findings confirm the hypothesis since they show that a subset of 10 features out of the original 21 may successfully distinguish between people with ASD and those who do not by using the swarm intelligence-based single-objective binary firefly feature selection wrapper.

Suman raj and Sarfaraz masood [26] proposed the analysis and detection of ASD using ML techniques. This study looks into the viability of using different ML algorithms, such as Naive Bayes, SVM, Logistic Regression, KNN, Neural Network, and Convolutional Neural Network, to predict and analyze cases of ASD in children, adolescents, and adults. Three different publicly accessible datasets that are specific to people without ASD diagnoses are used to evaluate the efficacy of these proposed strategies.

Khondaker A et.al [2] proposed a cloud-based, automated system designed for autism screening and confirmation is called Smart Autism. This innovation is especially important in areas with limited resources since it prevents prompt interventions due to delayed autism detection. The suggested solution is a three-tiered, mobile, interactive, integrated system that serves age groups spanning from 0 to 17 years old. Initially, a smartphone app is used to administer a picture-based screening questionnaire. If signs of autism appear, a virtual evaluation with video recording and cloud-based professional evaluation follows. Referral to the closest Autism Resource Center (ARC) for a conclusive assessment occurs if ambiguity continues.

J.A.Kosmicki et.al [17]. proposed a search for a minimal set of behaviors for autism detection through feature selection-based ML. That is he supposed a searching method for the least set of traits for autism detection. In this, the authors used a ml approach to evaluate the clinical assessment of ASD. The ADOS was performed on the subset of behaviors of children based on the autism spectrum. In this work different ML algorithms were employed, involving stepwise backward feature identification on score sheets from 4540 individuals.

Kaushik Vakadkar et.al [31]. proposed the detection of ASD in children using ml techniques. He applied models such as SVM, RFC, Naïve Bayes (NB), Logistic Regression (LR), and KNN to dataset. Predictive models were then created utilizing these algorithms. The study's main objective is to identify young children's possible sensitivity to ASD in order to aid in a more effective diagnostic process. Notably, the results show that among the models evaluated on the dataset, Logistic Regression has the highest accuracy.

Bram van den Bekerom [32] proposed that they used ml for the detection of ASD. Children's ASD has been predicted by taking into certain characteristics, including developmental delay, learning difficulty, and speech/language difficulties. Also, it has been demonstrated that adding variables like physical activity, early birth, and birth weight improves accuracy. The 1-away method was also successful in predicting the severity of ASD with considerable accuracy. The accuracy increased significantly as a result of this strategy, from 54.1% to 90.2%. The fact that the severity estimates were entirely dependent on the opinions of the kids' caretakers highlights the need for more study in this area.

Wall et al. [33] used an Alternating Decision Tree (ADTree) to reduce the screening time and faster detection of ASD traits. They used the Autism Diagnostic Interview, Revised (ADI-R) method and achieved a high level of accuracy with data from 891 individuals. However, the test was limited to the age of 5 to 17 and failed to predict ASD for different age groups (children, adolescents, and adults).

Bone et al. [8] applied ML for the same purpose and used a support vector machine (SVM) to obtain 89.2% sensitivity and 59% specificity. Their research included 1264 individuals with ASD and 462 individuals with NON-ASD traits. However, due to the wide range of ages (4-55 years), their research was not accepted for people of all age groups as a screening approach.

Bekerom [32] used several ML techniques, including naive Bayes, SVM, and random forest algorithm to determine ASD traits in children like developmental delay, obesity, and less physical activity, and compared those results.

Heinsfeld [14] applied a deep learning algorithm and neural network to identify ASD patients using a large brain imaging dataset from the Autism Imaging Data Exchange (ABIDE I) and achieved a mean classification accuracy of 70% with an accuracy range of 66% to 71%. The SVM classifier achieved a mean accuracy of 65%; while the RFC achieved a mean accuracy of 63%.

Chapter 4
Method

This chapter outlines the methodology used in the experimentation phase to accomplish the objectives of the thesis and address the research questions. The experiment involved comparing the performance of two ML algorithms trained and tested on the dataset for identifying ASD. The two algorithms are compared based on evaluating the performance metrics such as accuracy and the F1 score. The most appropriate algorithm for identifying ASD has been chosen. The experimentation procedure can be described as follows:

- Collecting a suitable dataset to identify ASD is the first step to proceeding with the experiment.
- Data preparation is an important step that involves data preprocessing and splitting the data into train and test sets.
- The proposed methodology is designed with two experiments.
  - Exp-1: Training the selected ML algorithms (SVM, RF) with default parameters.
  - Exp-2: Training the selected ML algorithms (SVM, RF) with hyper-parameter tuning.
- Evaluating the performance of both experiments on the test data set.
- Comparative analysis of the results for the two experiments and selecting the best algorithm to predict ASD.

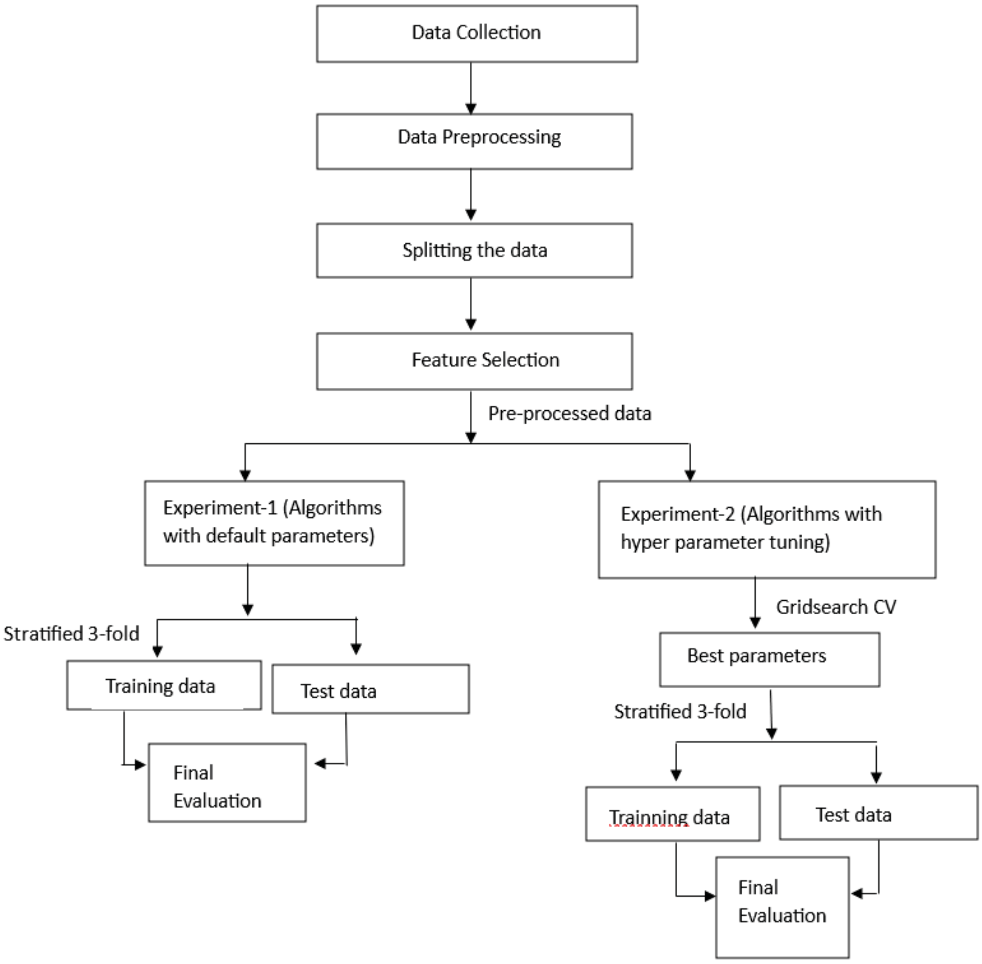

Figure 4.1: Flowchart for methodology

4.1 Data Collection

The dataset used in this thesis identifying ASD Disorder is collected from the website Kaggle [1]. This labeled dataset contains 1054 instances and 18 attributes, including class variables. Figure 4.2 shows the overview of the dataset.

	Case_No	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	Age_Mons	Qchat-10-Score	Sex	Ethnicity	Jaundice	Family_mem_with_ASD	Who completed the test	ASD
0	1	0	0	0	0	0	0	1	1	0	1	28	3	f	middle eastern	yes	no	family member	No
1	2	1	1	0	0	0	1	1	0	0	0	36	4	m	White European	yes	no	family member	Yes
2	3	1	0	0	0	0	0	1	1	0	1	36	4	m	middle eastern	yes	no	family member	Yes
3	4	1	1	1	1	1	1	1	1	1	1	24	10	m	Hispanic	no	no	family member	Yes
4	5	1	1	0	1	1	1	1	1	1	1	20	9	f	White European	no	yes	family member	Yes

Figure 4.2: Overview of the dataset

Data type: Predictive, Nominal/categorical, binary, and continuous
Attributes:

- The Q-Chat-10 questionnaire has 10 items with possible answers of "Always," "Usually," "Sometimes," "Rarely," and "Never."
- The dataset maps the values for items A1-A10 to either "1" or "0."

- For questions A1-A9, [31] the description for the questions has explained below:
  - A1: Child responding to you calling his/her name.
  - A2: Ease of getting eye contact from child.
  - A3: Child pointing to objects he/she wants.
  - A4: Child pointing to draw your attention to his/her interests.
  - A5: If the child shows pretense.
  - A6: Ease of the child to follow where you point/look.
  - A7: If the child wants to comfort someone who is upset.
  - A8: Child's first words.
  - A9: If the child's uses basic gestures. A response of "Sometimes," "Rarely," or "Never" is assigned a value of "1."
- For question A10, if the child daydreams/stares at nothing, a response of "Always," "Usually," or "Sometimes" is assigned a value of "1."
- To determine if a child exhibits potential ASD traits, the scores for all 10 questions are added together.
- If the total score is greater than 3, then the child may have ASD traits.
- If the total score is 3 or less, then no ASD traits are observed.

4.2 Data Preparation

4.2.1 Data Preprocessing

Preprocessing is commonly used to remove columns that are not relevant or useful for a specific analysis or modeling task. It can also help to reduce the dimensionality of the dataset, which can improve the efficiency and performance of models.

Data preprocessing is performed on the dataset to address missing or null values and to balance the data. The methods from Pandas are used to replace specific values in a data frame for toddlers with numeric values. This type of preprocessing is commonly used to convert categorical variables with text values into numerical variables that can be used in models. It is useful to ensure that the data is consistent and free from errors. Figure 4.3 shows the overview of the dataset after preprocessing.

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	Age_Mons	Sex	Ethnicity	Jaundice	Family_mem_with_ASD	Who completed the test	ASD
0	0	0	0	0	0	0	1	1	0	1	28	0	0	1	0	0	0
1	1	1	0	0	0	1	1	0	0	0	36	1	1	1	0	0	1
2	1	0	0	0	0	0	1	1	0	1	36	1	0	1	0	0	1
3	1	1	1	1	1	1	1	1	1	1	24	1	2	0	0	0	1
4	1	1	0	1	1	1	1	1	1	1	20	0	1	0	1	0	1
5	1	1	0	0	1	1	1	1	1	1	21	1	7	0	0	0	1
6	1	0	0	1	1	1	0	0	1	0	33	1	4	1	0	0	1
7	0	1	0	0	1	0	1	1	1	1	33	1	4	1	0	0	1
8	0	0	0	0	0	0	1	0	0	1	36	1	4	0	0	0	0
9	1	1	1	0	1	1	0	1	1	1	22	1	5	0	0	1	1
10	1	0	0	1	0	1	1	0	1	1	36	1	2	1	1	0	1
11	1	1	1	1	0	1	1	1	0	1	17	1	0	1	0	0	1
12	0	0	0	0	0	0	0	0	0	0	25	0	0	1	0	0	0
13	1	1	1	1	0	0	1	0	1	1	15	0	0	1	0	0	1
14	0	0	0	0	0	0	0	0	0	0	18	1	0	0	0	0	0
15	1	1	1	0	1	0	1	1	0	1	12	1	7	0	0	0	1
16	0	0	0	0	0	0	0	0	0	0	36	1	0	0	1	0	0
17	1	1	1	0	1	1	1	1	0	1	12	0	0	1	0	0	1
18	1	0	0	0	1	0	0	0	0	1	29	0	0	0	0	0	0
19	1	1	1	0	1	0	1	1	0	1	12	0	7	0	0	0	1

Figure 4.3: Overview of the dataset after pre-processing

4.2.2 Splitting the data

After preprocessing the data, the dataset is then split into training and testing datasets using the train_test_split function from sklearn's model_selection module. The test data is used to assess the model's performance on unseen data, whereas the training set is utilized to construct and train the ML model. Splitting of the data can make sure that the model's performance estimations are accurate and have the potential to extend the data using newly collected sets of data. The dataset is divided in the ratio 80:20 i.e. the data is divided into 80% train data and 20% test data.

```
# drop target feature 'ASD', and 'results' as it is the sum of the first 10 features (A1 to A10 scores)
y = toddlers_df['ASD']
X = toddlers_df.drop(['ASD'], axis=1)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=True)
X_train.head(10)
```

Figure 4.4: Splitting the data using train_test_split function

4.2.3 Feature Selection

After the data is divided, the focus will be on the training set. Finding the most important and instructive features for the task is the main aim of feature selection. In order to apply feature selection to the training set, it is necessary to examine how each feature relates to the target variable. The risk of overfitting is reduced, and the model's effectiveness and capacity to generalize are both enhanced by choosing only the most important features.

```
#feature selection chi2 test
num_features_to_select = 10  # Choose the number of features you want to select
selector = SelectKBest(score_func=chi2, k=num_features_to_select)
X_train_selected = selector.fit_transform(X_train, y_train)
X_test_selected = selector.transform(X_test)
selector.get_support()

array([ True,  True,  True,  True,  True,  True,  True,  True,  True,
       False, False, False,  True, False, False, False])
```

Figure 4.5: Feature Selection code snippet

After performing feature selection on the training set, it's crucial to align the test set with the same set of selected features. By doing this, we can be sure that the model is assessed consistently and that the performance findings are accurate. Any features that were not chosen during the feature selection procedure on the training set are removed from the test set.

To reduce the model overfitting, the chi-squared test is used to select the required features that are highly dependent on the response. The selected features are "A1, A2, A3, A4, A5, A6, A7, A8, A9, Ethnicity". Based on the selected features, further steps are followed [5].

To determine the best algorithm among the selected algorithms, we are doing two experiments, experiment-1 shows training and evaluating the models with the default parameters, and experiment-2 shows training and evaluating the models with the hyper-parameter tuning. By comparing the two experiments, the best model is determined.

4.3 Model Validation

We used stratified 3-fold cross-validation on the train set to accurately assess the performance of the models. By ensuring that the distribution of the target classes in each fold is representative of the entire dataset, stratified cross-validation helps detect concerns with class imbalance. Divide the train set into three subsets (folds) when using 3-fold cross-validation. On two of these folds, the model is trained and validated on the remaining fold. This process is performed three times, each time using a different fold for validation. This results in three models, each trained and evaluated on a different subset of the data. This validation process is applied for both the models in both experiments.

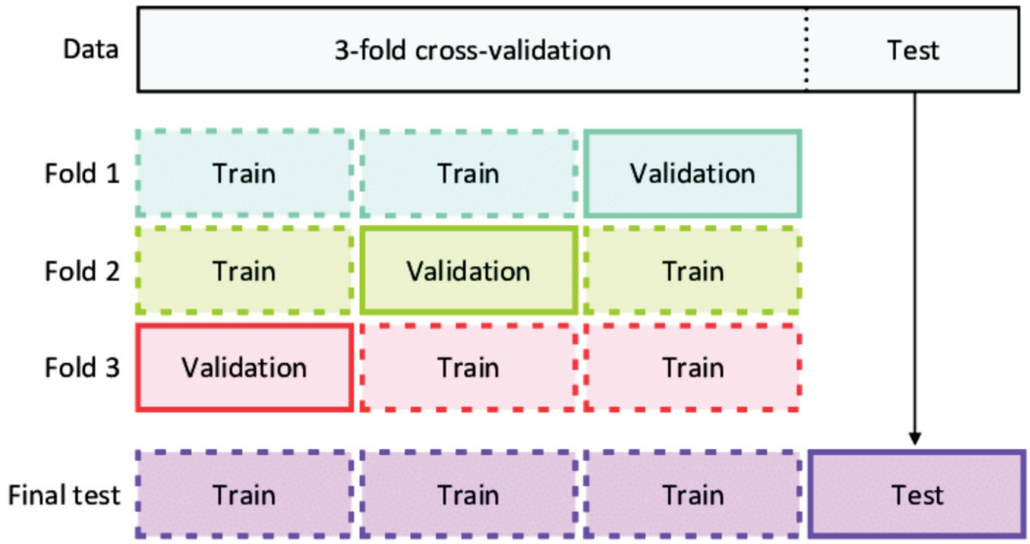

Figure 4.6: The process involved in 3-fold cross-validation and final test

4.4 Experiment-1: Default parameters

4.4.1 Training and Testing the models

In this section, we trained SV and RF models using the pre-processed data without altering their internal parameters. We imported these models from scikit-learn and trained them on the training data (X_train_selected and y_train) using the 'fit' method. We then evaluated their performance using key metrics like accuracy and F1 score.

To ensure the correctness of these models, we applied them to a test dataset containing unseen data. This allowed us to measure how well the models could make accurate predictions on unseen data. The evaluation included computing accuracy and F1 score on the test data. This process ensures that the models can effectively classify beyond the training data.

4.5 Experiment-2: Hyper-parameter tuning

4.5.1 Training and Testing the model

Just as we conducted in the previous experiment, the same approach is applied here. We perform hyperparameter tuning using GridSearchCV and search through a predefined range of hyperparameters for each model. Cross-validation is used by GridSearchCV to evaluate the model's performance. This helps in identifying the set of hyperparameters that produces satisfactory results. The parameters used in GridSearchCV are "classify, param_grid, cv=stratified_kfold, scoring". The parameters used in hyperparameter tuning in SVM were 'C' and 'Kernel', and the parameters used in RF were n_estimators, max_depth, min_samples_split.

The parameter **'C'** is the regularization parameter in SVM. The values specified are [0.1, 1, 10] which is a range to start with. Smaller values represent a weaker reg-

ularization and larger value represents stronger regularization. The 'Kernel' specifies the type of kernel function. The two common choices were taken, one is 'linear' and the other is 'rbf'.
'linear': This corresponds to a linear kernel. It creates a linear decision boundary.
'rbf' (Radial Basis Function): This corresponds to an RBF kernel. It is capable of modeling complex, non-linear decision boundaries.

```
# Experiment-3: SVM with Hyper parameter tuning

classify=svm.SVC()

stratified_kfold = StratifiedKFold(n_splits=3, shuffle=True, random_state=42)
# GridSearchCV for parameter tuning
param_grid = {'C': [0.1, 1, 10], 'kernel': ['linear', 'rbf']}
grid_search_svm = GridSearchCV(classify, param_grid, cv=stratified_kfold, scoring='accuracy')
grid_search_svm.fit(X_train_selected, y_train)

# Best parameters
best_params = grid_search_svm.best_params_
print('Best parameters:', best_params)
```

Figure 4.7: Code snippet for using parameters and selecting the best parameters of SVM with hyperparameter tuning

In RF, **n_estimators**: Controls the number of trees in the ensemble. The values specified are [50, 100, 150], representing the number of trees to consider.
max_depth: Determines the maximum depth of each tree. The values specified are [None, 10, 20], providing options for no maximum depth and two specific depth values.
min_samples_split: Sets the minimum number of samples required to split a node. The values specified are [2, 5, 10], indicating the minimum number of samples required to split a node.

```
# Experiment : Random Forest for Hyperparameter tuning

clf=RandomForestClassifier()

stratified_kfold = StratifiedKFold(n_splits=3, shuffle=True, random_state=42)
# GridSearchCV for parameter tuning
param_grid = {'n_estimators': [50, 100, 150],
              'max_depth': [None, 10, 20],
              'min_samples_split': [2, 5, 10]}
grid_search_rf = GridSearchCV(clf, param_grid, cv=stratified_kfold, scoring='accuracy')
grid_search_rf.fit(X_train_selected, y_train)

# Best parameters
best_params = grid_search_rf.best_params_
print('Best parameters:', best_params)
```

Figure 4.8: Code snippet for using parameters and selecting the best parameters of RF with hyperparameter tuning

After applying GridSearchCV, the process identifies the best parameters. These best parameters are the ones that optimize the model's predictive ability on the validation data. These best parameters ensure that the models are finely tuned to deliver the best possible results. With the best parameters, we evaluate the models using stratified 3-fold cross-validation. During each fold, the model is trained and validated using the best parameters. So, the results are in multiple performance scores for each fold. The selected models were trained using the best parameters and calculated performance metrics such as accuracy and f1 score.

The best model is trained on the training dataset and used to make predictions on the test data. Test accuracy and F1 score are calculated to evaluate the model's performance on unseen data.

4.6 Final model and Prediction

We conducted a performance comparison of the selected ML algorithms to determine the best model. This comparison involved evaluating the models using performance metrics such as accuracy and F1 score. Based on the results obtained from this evaluation, we made a selection of the most effective model for predicting Autism Spectrum Disorder (ASD). The best model was identified, and it is used to make predictions for ASD detection.

Based on the comparison, the best model is used to predict ASD. Here the model is predicted with the new, unseen data. Whether a patient has ASD or not has been predicted with the person's input data. If the predicted value is 0, it means that the patient is classified as non-ASD, and if the predicted value is 1, it means that the patient is classified as having ASD. Finally, the result indicates whether the patient is ASD or non-ASD.

Chapter 5

Results and Analysis

This chapter discusses the results obtained in the experiment. The ASD is predicted using the best Ml algorithm, to choose the best algorithm among the selected algorithms, two experiments were conducted. The two experiments procedure was discussed in the method section. The results of two experiments were compared. The following table shows the results of the calculated mean, SD, and accuracy of the SVM and RF models for the two experiments.

5.1 Results for Experiment-1

This section shows the results of experiment-1. In experiment-1, SVM and RF models were built using stratified 3-fold cross-validation on the train set and calculated accuracy and f1 score for the test data. And also provided the mean and SD of each score suchas accuracy, f1 scores for the stratified 3-fold cross validation of each model.

Algorithm	Accuracy		F1 score	
	Mean	Standard deviation	Mean	Standard deviation
SVM	0.9466	0.0126	0.9612	0.0092
RF	0.9383	0.0206	0.9564	0.0108

Table 5.1: Training results of Experiment-1

Algorithm	Accuracy	F1 score
SVM	0.9628	0.9733
RF	0.9431	0.9597

Table 5.2: Testing results of Experiment-1

5.2 Results for Experiment-2

This section shows the results of experiment 2. In experiment 2, SVM and RF models were built using stratified 3-fold cross-validation on the train set and performed parameter tuning using GridSearchCV and calculated accuracy and f1 score for the test data. The mean and SD of each score such as accuracy, and f1 scores is also provided for the stratified 3-fold cross-validation of each model.

Algorithm	Accuracy		F1 score	
	Mean	Standard deviation	Mean	Standard deviation
SVM	0.9608	0.0161	0.9712	0.0120
RF	0.9418	0.0170	0.9516	0.0099

Table 5.3: Training results of Experiment-2

Algorithm	Accuracy	F1 score
SVM	0.9668	0.9768
RF	0.9573	0.9702

Table 5.4: Testing results of Experiment-2

Chapter 6

Discussion

This chapter discusses the research questions and the obtained results for them. In this thesis, we studied which algorithm is best for predicting ASD among the selected algorithms (SVM, RF). To study this, the research is followed by two questions:

- **RQ1.** How well do the chosen algorithms (SVM, RF) recognize ASD?
- **RQ2.** To what extent will parameter tuning affect the performance of the selected algorithms, and how?

In order to answer the questions, we designed two experiments. While experiment one looks at the problem from the perspective of which ML algorithm is best. The second experiment looks at the effect of parameter tuning and identifies whether it can lead to selecting the same algorithm as in RQ1.

In answer to RQ1, a number of important factors were taken into account to evaluate how well the chosen ML algorithm performed in classifying ASD. These components included performance metric, feature selection, hyper-parameter tuning, and data quality.

An analysis of performance measures was carried out to thoroughly assess the effectiveness of the SVM and RF algorithms in identifying ASD. These measurements, such as accuracy and F1 scores, were considered for judging the efficiency of the algorithms. To learn more about performance consistency, the mean and SD of these scores were also evaluated.

After calculating the performance of SVM with an average accuracy of 94.66% and f1 score of 96.12% on the training and average accuracy of 96.28% and f1 score of 97.33% on testing and the performance of RF with an average accuracy of 93.83% and f1 score of 95.64% on the training and average accuracy of 94.31% and f1 score of 95.97% on testing, it is clear that SVM performed better than RF at identifying ASD. This result was reached based on performance metrics average accuracy and f1 score, which together gave a complete view of how well the algorithms handled the ASD detection task.

In answer to the RQ2, as the method section described, how two different sets of experiments were conducted to identify ASD. SVC particularly performed well across both sets of experiments. While the SVC's default parameters achieved good results, the use of hyperparameter tuning improved it even more, leading to an outstanding detection of ASD with an accuracy of 96.68%.

Parameter tuning allows algorithms to better fit in the data. It helps to reduce the risk of overfitting and its ability on unseen data and also improves model accuracy. Tuning parameters can also affect the efficiency of the algorithms and also allows algorithms to adapt to the specific characteristics of the dataset. In this way, parameter tuning affected the performance of the selected algorithms.

Based on the above comparison of the results SVM performed well compared to RF. Using the SVM model the ASD is predicted.

```
input_data = (1, 0, 0, 1, 1, 1, 0, 0, 0, 1)
input_data_as_numpy_array = np.asarray(input_data)
input_data_reshaped = input_data_as_numpy_array.reshape(1, -1)
prediction = grid_search_svm.predict(input_data_reshaped)
if prediction == 0:
    print('Patient is non ASD')
else:
    print('Patient is ASD')
```

```
Patient is ASD
```

Figure 6.1: ASD is predicted

Similarly, Kaushik Vakadkar et al. [31] proposed the detection of ASD in children using machine learning techniques. He applied models such as SVM, RFC, NB, LR, and KNN to the dataset. Predictive models were then created utilizing these algorithms. The study's main objective is to identify young children's possible sensitivity to ASD in order to aid in a more effective diagnostic process.

The results obtained were:

	LR	NB	SVM	KNN	RFC
Accuracy	97.15%	94.79%	93.84%	90.52%	81.52%
Confusion matrix	$\begin{bmatrix} 57 & 5 \\ 1 & 148 \end{bmatrix}$	$\begin{bmatrix} 56 & 6 \\ 5 & 144 \end{bmatrix}$	$\begin{bmatrix} 52 & 10 \\ 3 & 146 \end{bmatrix}$	$\begin{bmatrix} 51 & 11 \\ 9 & 140 \end{bmatrix}$	$\begin{bmatrix} 45 & 17 \\ 14 & 135 \end{bmatrix}$
F1 score	0.98	0.96	0.95	0.93	0.88

Figure 6.2: The obtained results for the related work, we discussed

He concluded that there is presently no diagnostic test that can diagnose ASD fast and accurately, nor is there an optimal and thorough screening method that is specifically designed to detect the onset of ASD. Out of the five models that were applied to our dataset, LR was observed to give the highest accuracy.

A few of the limitations associated with our study are when the dataset is short or highly unbalanced, SVMs may not perform effectively. In the case of ASD prediction, collecting a balanced dataset with enough samples for each class (ASD and non-ASD) can be difficult. Small datasets may result in overfitting, limiting the generalizability of the model.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

ASD is a complex neurodevelopmental disorder with profound impacts on social communication, behavior, and cognitive development. ASD affects a significant portion of the population, with one in 100 children estimated to be affected by it, according to the WHO. Early identification is essential as it can significantly improve the effectiveness of interventions and therapies.

ML algorithms such as SVM and RF, were selected in this thesis to contribute to the early prediction of ASD. The primary objective was to build and train models using these algorithms, comparing their performances helps to identify the best algorithm to predict ASD. To conclude this, the methodology is followed by two experiments: experiment-1 shows that the models were trained with default parameters and experiment-2 shows that the models were trained with hyperparameter tuning. After training the models the performance metrics such as accuracy and f1 score were calculated to select the best algorithm to predict ASD.

Upon careful evaluation, the performance of the SVM algorithm with hyperparameter tuning given accuracy score of 96% and an f1 score of 97% for test data. Based on the results we can conclude that the SVM is better performing algorithm than the RF algorithm to predict ASD in terms of both accuracy and f1 scores.

7.2 Future work

The following are some of the ways to develop this research further:

- We can research ensemble learning techniques that combine multiple ML models, such as bagging, boosting, or stacking. Ensemble methods also lead to improved predictive accuracy.

- Validate the models on external datasets, preferably from diverse sources or populations. This ensures the adaptability of the model beyond the training dataset.

- Collaborate with healthcare professionals to integrate the predictive model into clinical practice. Explore how the model can assist in early diagnosis or support treatment decisions.

- Create user-friendly interfaces or applications that enable people with ASD and their families to interact with the predictive model to understand its predictions.

References

[1] "Autism screening data for toddlers — kaggle.com," https://www.kaggle.com/datasets/fabdelja/autism-screening-for-toddlers?resource=download, [Accessed 08-10-2023].

[2] K. A. Al Mamun, S. Bardhan, M. A. Ullah, E. Anagnostou, J. Brian, S. Akhter, and M. G. Rabbani, "Smart autism—a mobile, interactive and integrated framework for screening and confirmation of autism," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2016, pp. 5989–5992.

[3] Z. Alizadeh-Sani, P. P. Martínez, G. H. González, A. González-Briones, P. Chamoso, and J. M. Corchado, "A hybrid supervised/unsupervised machine learning approach to classify web services," in *Highlights in Practical Applications of Agents, Multi-Agent Systems, and Social Good. The PAAMS Collection: International Workshops of PAAMS 2021, Salamanca, Spain, October 6–9, 2021, Proceedings 19*. Springer, 2021, pp. 93–103.

[4] S. Aljahdali, S. N. Hussain *et al.*, "Comparative prediction performance with support vector machine and random forest classification techniques," *International journal of computer applications*, vol. 69, no. 11, 2013.

[5] M. Bala, A. A. Prova, and M. H. Ali, "Prediction of autism spectrum disorder using feature selection and machine learning algorithms," in *Proceedings of International Conference on Computational Intelligence and Emerging Power System: ICCIPS 2021*. Springer, 2022, pp. 133–148.

[6] E. Bisong and E. Bisong, "More supervised machine learning techniques with scikit-learn," *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*, pp. 287–308, 2019.

[7] ——, "Numpy," *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*, pp. 91–113, 2019.

[8] D. Bone, S. L. Bishop, M. P. Black, M. S. Goodwin, C. Lord, and S. S. Narayanan, "Use of machine learning to improve autism screening and diagnostic instruments: effectiveness, efficiency, and multi-instrument fusion," *Journal of Child Psychology and Psychiatry*, vol. 57, no. 8, pp. 927–937, 2016.

[9] I. El Naqa and M. J. Murphy, *What is machine learning?* Springer, 2015.

[10] J. H. Elder, C. M. Kreider, S. N. Brasher, and M. Ansell, "Clinical impact of early diagnosis of autism on the prognosis and parent–child relationships," *Psychology research and behavior management*, pp. 283–292, 2017.

[11] L. Gigović, H. R. Pourghasemi, S. Drobnjak, and S. Bai, "Testing a new ensemble model based on svm and random forest in forest fire susceptibility assessment and its mapping in serbia's tara national park," *Forests*, vol. 10, no. 5, p. 408, 2019.

[12] G. Hackeling, *Mastering Machine Learning with scikit-learn*. Packt Publishing Ltd, 2017.

[13] M. F. A. Hady and F. Schwenker, "Semi-supervised learning," *Handbook on Neural Information Processing*, pp. 215–239, 2013.

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

[15] C. Huang, S.-X. Li, C. Caraballo, F. A. Masoudi, J. S. Rumsfeld, J. A. Spertus, S.-L. T. Normand, B. J. Mortazavi, and H. M. Krumholz, "Performance metrics for the comparative analysis of clinical risk prediction models employing machine learning," *Circulation: Cardiovascular Quality and Outcomes*, vol. 14, no. 10, p. e007526, 2021.

[16] V. Jakkula, "Tutorial on support vector machine (svm)," *School of EECS, Washington State University*, vol. 37, no. 2.5, p. 3, 2006.

[17] J. Kosmicki, V. Sochat, M. Duda, and D. Wall, "Searching for a minimal set of behaviors for autism detection through feature selection-based machine learning," *Translational psychiatry*, vol. 5, no. 2, pp. e514–e514, 2015.

[18] B. Mahesh, "Machine learning algorithms-a review," *International Journal of Science and Research (IJSR).[Internet]*, vol. 9, pp. 381–386, 2020.

[19] C. A. Mazefsky, D. P. Oswald, T. N. Day, S. M. Eack, N. J. Minshew, and J. E. Lainhart, "Asd, a psychiatric disorder, or both? psychiatric diagnoses in adolescents with high-functioning asd," *Journal of Clinical Child & Adolescent Psychology*, vol. 41, no. 4, pp. 516–523, 2012.

[20] W. McKinney *et al.*, "pandas: a foundational python library for data analysis and statistics," *Python for high performance and scientific computing*, vol. 14, no. 9, pp. 1–9, 2011.

[21] S. J. Moon, J. Hwang, R. Kana, J. Torous, and J. W. Kim, "Accuracy of machine learning algorithms for the diagnosis of autism spectrum disorder: Systematic review and meta-analysis of brain magnetic resonance imaging studies," *JMIR mental health*, vol. 6, no. 12, p. e14108, 2019.

[22] V. Nasteski, "An overview of the supervised machine learning methods," *Horizons. b*, vol. 4, pp. 51–62, 2017.

[23] S. Pattanayak and T. Singh, "Cardiovascular disease classification based on machine learning algorithms using gridsearchcv, cross validation and stacked ensemble methods," in *Advances in Computing and Data Sciences: 6th International Conference, ICACDS 2022, Kurnool, India, April 22–23, 2022, Revised Selected Papers, Part I*. Springer, 2022, pp. 219–230.

[24] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[25] S. Prusty, S. Patnaik, and S. K. Dash, "Skcv: Stratified k-fold cross-validation on ml classifiers for predicting cervical cancer," *Frontiers in Nanotechnology*, vol. 4, p. 972421, 2022.

[26] S. Raj and S. Masood, "Analysis and detection of autism spectrum disorder using machine learning techniques," *Procedia Computer Science*, vol. 167, pp. 994–1004, 2020.

[27] J. Reilly and Akkio, 2022. [Online]. Available: https://www.akkio.com/post/precision-vs-recall-how-to-use-precision-and-recall-in-machine-learning-complete-guide#:~:text=They%20are%20both%20related%20to,correctly%20identified%20by%20the%20model.

[28] f. score, "F1 score in machine learning: Intro amp; calculation," 2023. [Online]. Available: https://www.v7labs.com/blog/f1-score-guide#:~:text=for%20Machine%20Learning-,What%20is%20F1%20score%3F,prediction%20across%20the%20entire%20dataset.

[29] D.-Y. Song, S. Y. Kim, G. Bong, J. M. Kim, and H. J. Yoo, "The use of artificial intelligence in screening and diagnosis of autism spectrum disorder: a literature review," *Journal of the Korean Academy of Child and Adolescent Psychiatry*, vol. 30, no. 4, p. 145, 2019.

[30] R. Vaishali and R. Sasikala, "A machine learning based approach to classify autism with optimum behaviour sets," *Int. J. Eng. Technol*, vol. 7, no. 4, p. 18, 2018.

[31] K. Vakadkar, D. Purkayastha, and D. Krishnan, "Detection of autism spectrum disorder in children using machine learning techniques," *SN Computer Science*, vol. 2, pp. 1–9, 2021.

[32] B. van den Bekerom, "Using machine learning for detection of autism spectrum disorder," in *Proc. 20th Student Conf. IT*, 2017, pp. 1–7.

[33] D. P. Wall, R. Dally, R. Luyster, J.-Y. Jung, and T. F. DeLuca, "Use of artificial intelligence to shorten the behavioral diagnosis of autism," 2012.

[34] A. M. Wetherby, N. Watt, L. Morgan, and S. Shumway, "Social communication profiles of children with autism spectrum disorders late in the second year of life," *Journal of autism and developmental disorders*, vol. 37, pp. 960–975, 2007.

[35] M. A. Wiering and M. Van Otterlo, "Reinforcement learning," *Adaptation, learning, and optimization*, vol. 12, no. 3, p. 729, 2012.

Faculty of Faculty, Blekinge Institute of Technology, 371 79 Karlskrona, Sweden