Detection of Autism Spectrum Disorder Using Machine Learning and Deep Learning Techniques

This Dissertation is Submitted in Fulfillment
of the Requirements for the Degree of

Bachelor of Science (B.Sc.)

in

Computer Science and Engineering (CSE)

by

Abrar Faisal (C201058)
Motiur Rahman (C201065)
Md Rasel Opu(C201076)

TO
FACULTY OF SCIENCE AND ENGINEERING INTERNATIONAL ISLAMIC UNIVERSITY CHITTAGONG

DECLARATION

We hereby affirm the following statements regarding our thesis:

1. I formally state that I thoroughly reviewed this thesis and that I believe it to be comprehensive and of high enough quality to be accepted for the Bachelor of Science in Computer Science and Engineering undergraduate degree.
2. The thesis work does not contain any previously published or third-party content without proper citation.
3. The thesis work has not been previously submitted for any other degree or diploma at any other university or institution.
4. We have appropriately acknowledged all significant sources of contribution in the thesis.

Student's Full Name and Metric ID:

Abrar Faisal	**(C201058)**
Motiur Rahman	**(C201065)**
Md Rasel Opu	**(C201076)**

SUPERVISOR'S DECLARATION

I formally state that I have examined this thesis and claim it to be of sufficient quality and scope to be granted for the undergraduate degree of Bachelor of Science in Computer Science and Engineering.

Mr Mohammad Mahadi Hassan
Associate Professor
Department of Computer Science and Engineering International
Islamic University Chittagong

____

DEDICATION

This thesis report is dedicated to us, our supervisor and our family. The team work was satisfactory and the family’s support was incredibly amazing. Our dedicated and most hard working Supervisor who has been a constant support throughout these months. In this document, the contributions are acknowledged too.

ACKNOWLEDGMENT

To start with, All the praises to the Almighty Allah, for his mercy because of which we were able to finish our thesis despite having so many obstacles. Secondly, we would like to extend our gratitude to our supervisor, **(Mr. Mohammad Mahadi Hassan)** for his continuous effort and guidelines from the very beginning of our research.

ETHICAL STATEMENT

Hereby we state that, none of the unethical practices were used in the completion of our thesis work. The data we used for the research purpose are original. We carefully checked every citation we used here. The two writers of the work accept all the liabilities for any kind of violation of the thesis rule.

Abstract

Autism Spectrum Disorder (ASD) is a complex neurodevelopmental condition characterized by challenges in social interaction, communication, and behavior. Early detection of ASD is critical for timely intervention, yet it remains challenging due to the disorder's heterogeneous presentation and the limited availability of structured data. This study explores the potential of hybrid machine learning and deep learning models for ASD classification using tabular datasets of adults and toddlers. To address data limitations, we employed data augmentation techniques to improve model robustness and performance.Our findings demonstrate the effectiveness of hybrid approaches, such as Random Forest combined with Artificial Neural Networks (ANN), and Convolutional Neural Networks (CNN), in identifying complex patterns in ASD datasets. By applying these models across multiple datasets, including those sourced from the UCI repository and Kaggle, our results show consistent improvements in classification accuracy, precision, recall, and F1 scores compared to previous studies.This work underscores the importance of advanced hybrid modeling techniques in ASD detection, offering a scalable framework for clinical and research applications. The study emphasizes the need for continued innovation in leveraging machine learning for early ASD detection, with the ultimate goal of supporting early intervention strategies and improving outcomes for individuals with ASD.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATION

The following list provides descriptions of various symbols and abbreviations that will be utilized in the subsequent sections of the document.

- **ASD**: Autism Spectrum Disorder
- **ANN**: Artificial Neural Network
- **CNN**: Convolutional Neural Network
- **RF**: Random Forest
- **XGBoost**: eXtreme Gradient Boosting
- **GBM**: Gradient Boosting Machine
- **KNN**: K-Nearest Neighbors
- **SMOTE**: Synthetic Minority Over-sampling Techniqu

Chapter 1

Introduction

1.1 Overview

Autism Spectrum Disorder (ASD) is a complex neurodevelopmental condition that affects communication, behavior, and social interaction, often identified in early childhood but sometimes remaining undiagnosed until later stages. The global prevalence of ASD has escalated, bringing increased focus on timely, accurate diagnosis to ensure individuals receive the support they need. Traditional diagnostic methods are predominantly manual, time-consuming, and prone to subjectivity, leading to potential diagnostic delays. This study proposes an alternative approach by leveraging machine learning (ML) and deep learning (DL) models to classify ASD symptoms and improve diagnostic accuracy through automated, data-driven methods. By applying hybrid models, our aim is to harness the complementary strengths of both machine learning and deep learning techniques, achieving high accuracy and reliable performance across different age groups.

1.2 Motivation and Scope of the Research

Previous research has explored numerous machine learning models, such as support vector machines (SVM), k-nearest neighbors (KNN), and random forests (RF), achieving varied results depending on dataset characteristics and model limitations. However, a significant challenge encountered in these studies was the limited size of ASD-specific datasets, which restricts the model's learning capability and generalizability. Furthermore, studies primarily focused on individual models, which may not fully capture the complexity of ASD features across different age groups or datasets.

To address these issues, our study incorporates data augmentation techniques to overcome the data scarcity problem and improve model robustness. Data augmentation enhances the dataset by synthetically creating new data points, thus expanding the training dataset and improving model performance. Additionally, we adopt hybrid modeling by combining various classifiers, such as Random Forest with Artificial Neural Networks (ANN), and XGBoost with Logistic Regression. These hybrid models allow for more comprehensive feature learning and reduce bias, ultimately improving classification accuracy. By utilizing diverse machine learning algorithms and leveraging the power of deep learning models, this research aims to bridge the gap between conventional diagnostic methods and automated, scalable solutions.

1.3 Problem Statement

Autism Spectrum Disorder (ASD) diagnosis remains a complex and multi-faceted challenge, primarily due to its diverse symptom presentations and the subjective nature of manual assessments. While early diagnosis is critical for effective intervention, the reliance on conventional, manually-intensive methods often results in delays or inconsistencies. The issue is compounded by a scarcity of large, diverse datasets, particularly for different age groups, leading to limited accuracy and generalizability of existing machine learning (ML) models in ASD detection. Previous studies have shown the limitations of individual models, often due to the lack of hybrid approaches and inadequate data preparation techniques, such as augmentation, that could potentially address the dataset's diversity and increase accuracy. This thesis addresses these core issues by applying hybrid models and data augmentation techniques, aiming to improve the reliability and accuracy of ASD classification across multiple age groups.

1.4 Contribution of the thesis

The primary contributions of this thesis are as follows:

1. **Hybrid Model Implementation**: This research designs and implements multiple hybrid models that combine the strengths of different ML and DL approaches. By leveraging combinations such as Random Forest + ANN, XGBoost + Logistic Regression, and Convolutional Neural Networks (CNN), the study demonstrates how these models can exceed the performance of individual classifiers. Notably, the stacked models are tailored for each age-specific dataset, yielding improvements in diagnostic accuracy and stability.
2. **Data Augmentation for Enhanced Model Generalizability**: Addressing a major limitation in ASD research, this study uses data augmentation techniques to increase dataset diversity, particularly for the limited samples available in the toddler and adult datasets. This expanded dataset improves model training, enhancing the model's ability to generalize and perform accurately on previously unseen samples, which is critical for clinical relevance.
3. **Benchmarking Across Age-Based Datasets**: This thesis tests and benchmarks each hybrid model on datasets for distinct age groups—toddlers, adolescents, and adults—ensuring the models are adapted to different developmental stages. This age-specific focus not only provides more tailored insights but also contributes valuable findings on how hybrid model performance may vary by age group.

4. **Objective Validation of Model Performance**: Utilizing comprehensive evaluation metrics, including accuracy, precision, recall, and F1 score, the research rigorously validates the hybrid models' performance. By testing and refining each hybrid configuration, the study identifies optimal models that achieve high accuracy rates, with some surpassing good accuracy in specific datasets, underscoring the robustness of the proposed approach.

Chapter 2

Literature Review

Talukdar, Gogoi, and Singh (2023) reviewed the application of machine learning (ML) techniques in diagnosing Autism Spectrum Disorder (ASD). The authors discussed various ML methods, including supervised learning, unsupervised learning, and deep learning, as well as the challenges these techniques face, such as limited data availability and the need for clearer, interpretable models.They emphasized the importance of expanding datasets to include diverse populations and clinical settings, as well as developing interpretable models that can be effectively utilized in clinical practice. The study also highlighted the need for integrating multiple data sources, such as behavioral assessments and neuroimaging, to gain a more holistic understanding of ASD. Additionally, the authors called for the establishment of ethical guidelines for the application of ML in healthcare.[1]

Vakadkar, Purkayastha, and Krishnan (2020) explored machine learning (ML) techniques for detecting Autism Spectrum Disorder (ASD). They evaluated models such as Logistic Regression, Naïve Bayes, Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Random Forest, using a dataset that included features like age, sex, and ethnicity. The study found that Random Forest performed best in terms of accuracy and other metrics.However, the authors acknowledged limitations such as the dataset's lack of diversity and the need for further validation in clinical settings. They suggested expanding datasets and integrating additional factors, such as genetic data, for improved accuracy.[2]

Zaman, Ferdus, and Sattar (2021) proposed a machine learning-based approach to detect Autism Spectrum Disorder (ASD) in children. Their study identified the limitations of traditional diagnostic methods and presented ML as a reliable tool to assist in early detection, emphasizing its potential for improving intervention outcomes. They reviewed various algorithms, including Logistic Regression, K-Nearest Neighbors, Support Vector Classifier, Naïve Bayes, Decision Tree, and Random Forest, utilizing a dataset of 273 children to evaluate performance.The authors highlighted the importance of expanding datasets and integrating advanced features for improved accuracy in future research. Despite challenges such as small sample size, the study demonstrated that ML could play a vital role in enhancing diagnostic processes for ASD.[3]

Koehler, Dong, and Bierlich (2024) investigated the potential of machine learning (ML) and computer vision techniques to enhance the diagnostic process for Autism Spectrum Disorder (ASD). Their study focused on analyzing non-verbal behaviors and social interactions, such as facial expressions and gestures, as crucial diagnostic markers. Using the NeuroMiner toolbox in MATLAB, they employed stratified cross-validation to train robust ML models and utilized computer vision to capture subtle behavioral cues. The authors emphasized the need for larger, diverse samples in future research to validate their findings and suggested exploring the effects of sex and gender on ASD-related behaviors. Despite limitations, including small sample size and dependency on existing algorithms, their work highlights the promise of automated diagnostics in improving traditional ASD assessment practices.[4]
Raj and Masood (2020) explored the use of machine learning techniques for detecting Autism Spectrum Disorder (ASD). Their study emphasized the importance of accurate classification models and evaluated the performance of Support Vector Machine (SVM), Naïve Bayes (NB), and Convolutional Neural Network (CNN) algorithms. Among these, CNN demonstrated the highest accuracy in identifying ASD. The authors analyzed confusion matrices to assess model effectiveness comprehensively. The study identified limitations, such as reliance on a specific dataset, risk of overfitting in complex models, and the need for validation across diverse populations. Future research directions included expanding datasets, exploring advanced deep learning methods, and developing real-time ASD detection systems.[5]

Wall, Dally, Luyster, Jung, and DeLuca explored the potential of artificial intelligence to improve the behavioral diagnosis process for Autism Spectrum Disorder (ASD). By employing machine learning techniques, specifically the Alternating Decision Tree (ADTree) classifier, the study reduced the Autism Diagnostic Interview-Revised (ADI-R) questionnaire from 93 to just 7 questions, significantly simplifying the diagnostic process. This streamlined approach emphasized the importance of using advanced algorithms to enhance diagnostic efficiency while maintaining accuracy.The study also highlighted the need for further research to validate its findings in diverse populations and expand its applicability to other neurodevelopmental disorders. Limitations included reliance on simulated control data and the small number of matched controls, which may affect the generalizability of the results.[6]

Bishop-Fitzpatrick et al. (2020) explored lifetime health patterns in individuals with Autism Spectrum Disorder (ASD) using machine learning algorithms applied to electronic health records (EHR). The study aimed to identify unique diagnostic patterns among ASD decedents compared to a matched control group. Data analysis included comorbid health conditions and risk factors for various medical issues, emphasizing the importance of understanding the health trajectories of aging individuals with ASD. This research highlighted the utility of machine learning in analyzing large datasets to uncover critical health disparities and emphasized the need for further studies to address the gaps in healthcare for aging ASD populations. However, limitations such as potential inaccuracies in EHR data and sample biases underscore the need for more representative and comprehensive studies in the future.[7]

Alteneiji et al. (2020) investigated the potential of machine learning (ML) techniques for diagnosing Autism Spectrum Disorder (ASD), focusing on improving diagnostic accuracy and efficiency. The study utilized secondary data from the AQ-10 ASD diagnostic method, which included information from toddlers, adolescents, and children. By applying seven ML models, the research aimed to identify optimal algorithms for detecting ASD symptoms and enabling earlier interventions. The authors highlighted the importance of leveraging ML to address the limitations of current diagnostic practices, which often fail to differentiate effectively between individuals with ASD and typically developing individuals. Future work proposed by the study includes the development of user-friendly applications for ASD screening, integration with national health databases, and implementation in educational and childcare settings to enhance early detection and accessibility.[8]

Sumana and Amrutha (2021) explored the application of machine learning techniques for detecting Autism Spectrum Disorder (ASD) in children aged 1-5 years. The study employed multiple algorithms, including Logistic Regression, Naïve Bayes, Decision Tree, and K-Nearest Neighbors, to enhance early detection and intervention for ASD. The dataset, sourced from Kaggle, underwent extensive pre-processing to address missing and irrelevant data before model evaluation. The authors emphasized the significance of using metrics like accuracy, precision, recall, and execution time to compare algorithm performance. While the study demonstrated promising outcomes, it highlighted the need for larger datasets and advanced features to improve ASD prediction further. Future directions proposed include incorporating real-time data analysis and exploring additional machine learning techniques for broader application.[9]

Emon et al. (2021) conducted a study focused on developing an affordable and efficient screening method for Autism Spectrum Disorder (ASD) using various machine learning algorithms. The study explored the potential of Sequential Minimal Optimization (SMO), Stochastic Gradient Descent (SGD), Random Forest (RF), Multilayer Perceptron (MLP), J48, and Logistic Regression algorithms for early ASD detection.The research highlights the limitations of traditional diagnostic methods and emphasizes the promise of machine learning in improving the speed and accuracy of ASD screening. The authors suggest that future work should include the validation of the findings using larger, more diverse datasets, and the incorporation of additional features to enhance the screening process's accuracy.[10]

Chapter 3

Methodology

3.1 Introduction

The methodology section outlines the systematic approach employed in this study to develop and validate hybrid machine learning models for Autism Spectrum Disorder (ASD) detection. It describes the processes of data collection, preprocessing, augmentation, model design, and evaluation. By integrating advanced techniques and hybrid models, this methodology ensures robust classification accuracy and generalizability across diverse datasets for different age groups.

3.2 Overview

This study employs a structured methodology to develop and evaluate hybrid machine learning models for Autism Spectrum Disorder (ASD) detection. The process begins with data collection and preprocessing, where datasets for toddlers, adolescents, and adults are cleaned, encoded, and scaled to ensure consistency and reliability. To address the inherent limitation of small dataset sizes, data augmentation techniques are applied, enhancing the diversity and richness of training samples. The research focuses on designing and implementing hybrid models, such as Random Forest + ANN and Stacked Models (e.g., Random Forest + XGBoost + ANN), to leverage the combined strengths of machine learning and deep learning approaches. Each model is rigorously evaluated using performance metrics like accuracy, precision, recall, and F1 score to ensure robust classification outcomes. Additionally, the methodology includes age-specific analysis, optimizing models separately for toddler and adult datasets to achieve targeted and effective ASD detection. This comprehensive approach ensures a reliable framework for advancing ASD classification accuracy across diverse datasets.

3.3 Work flow

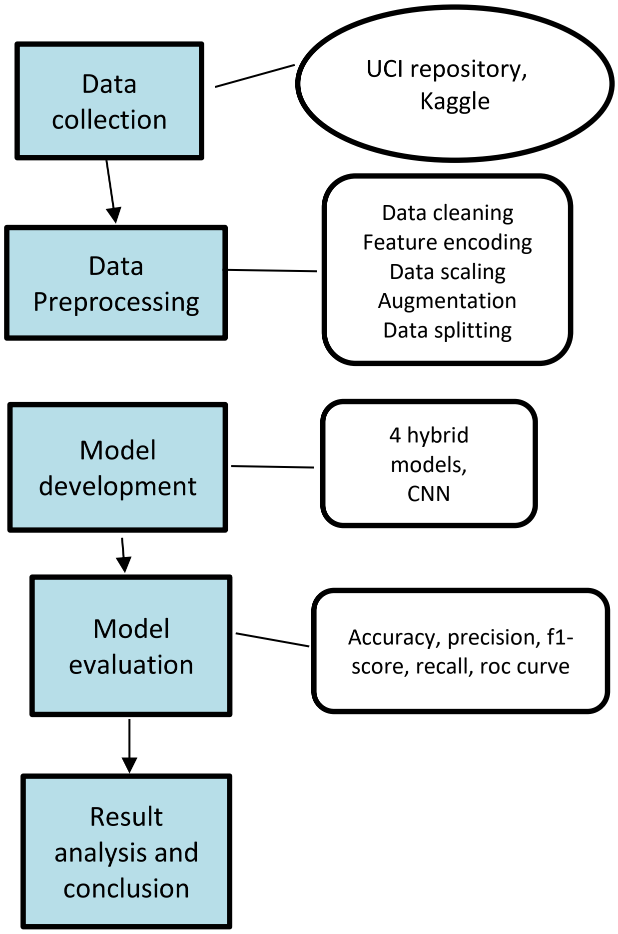


fig. 3.1. Work Flow Diagram

3.4 Data pre-processing

Data cleaning: In this research, data cleaning was a critical step to ensure the integrity and reliability of the datasets obtained from the UCI repository for adults and toddlers. This process involved addressing missing or inconsistent data values, which could negatively affect model performance if left unhandled.

Firstly, categorical attributes such as "ethnicity" and "relation" contained placeholders for missing values, represented as. These were replaced with NaN to standardize the representation of missing data. Once identified, the missing values in these attributes were imputed using the **most frequent value (mode)** from their respective columns. This method ensured minimal disruption to the distribution of the data while maintaining its integrity.

For the numerical attribute "age," missing values were handled differently. Initially, rows with missing values in this attribute were removed to clean the dataset. However, to retain as much data as possible, a second approach was also applied: converting the "age" attribute to a numeric format, coercing any errors into NaN. These missing values were then filled using the **mean value** of the column. This strategy preserved the dataset size and ensured that "age" values aligned with the overall data distribution.

After these cleaning steps, the datasets were checked for any remaining missing values to verify completeness. This meticulous process ensured that the datasets were free of anomalies and inconsistencies, creating a robust foundation for subsequent preprocessing steps.

Additionally, this cleaning process was consistently applied across both the adults and toddlers datasets to maintain uniformity in preparation and analysis.

Feature encoding: Feature encoding was a crucial step in preprocessing to ensure that all categorical variables in the dataset were transformed into a numerical format, enabling their use in machine learning models. The encoding process applied to both the adults and toddlers datasets involved two key steps: label encoding and scaling.

Label Encoding:

Categorical features such as "gender," "jundice," "autism," "country_of_res," "ethnicity," and "used_app_before" were non-numeric attributes that could not be directly utilized by machine learning models. To address this, I employed **Label Encoding**, which assigned a unique numeric value to each category in these features. This transformation preserved the categorical information in a format that machine learning algorithms could interpret effectively.

For instance, the "gender" column, which contained values such as "Male" and "Female," was converted into numerical representations, such as 0 for "Female" and 1 for "Male." Similarly, attributes like "jundice" (whether a person had jaundice) and "used_app_before" (whether a person had used a diagnostic app before) were encoded into binary values, simplifying the data while retaining its meaning. This process ensured consistency and standardization across the dataset.

Data scaling:

- **Feature Scaling**

Numerical attributes, particularly "age," were scaled using the **Min-Max Scaler** to normalize the data. This scaling method transformed the values of "age" to a range between 0 and 1. The primary objective was to eliminate the impact of differing scales between features, thereby ensuring that all features contributed equally to the model's performance.

Scaling was particularly important because features with larger ranges could disproportionately influence the model, potentially leading to biased predictions. By normalizing "age," I ensured

that all features were on an equal footing, enhancing the overall effectiveness and stability of the models.

These feature encoding techniques collectively enabled the datasets to be well-prepared for subsequent machine learning and deep learning processes, ensuring compatibility with hybrid models while maintaining the integrity of the original data.

Data augmentation:

Data augmentation was a pivotal part of my methodology, designed to address one of the main limitations of existing Autism Spectrum Disorder (ASD) detection studies: limited data availability. Limited datasets can result in overfitting, poor generalization, and reduced accuracy of models. To overcome this, I applied multiple data augmentation techniques across all three datasets—adults, toddlers, and toddlers from Kaggle. These techniques significantly increased the row count and variability in the datasets, improving model robustness and generalization.

Why Data Augmentation?

The datasets used in ASD detection studies often have a limited number of samples, especially for minority classes or specific subgroups. This limitation can lead to:

1. Overfitting: Models memorize the training data rather than learning generalized patterns.
2. Class Imbalance: Some classes may dominate the dataset, leading to biased predictions.
3. Limited Feature Diversity: A small dataset restricts the range of feature combinations that the model can learn.

By augmenting the data, I aimed to increase its size and variability, allowing the hybrid models to learn from a more diverse set of samples. This enhancement improved the overall accuracy and reliability of the models.

Data Augmentation Techniques

I employed the following augmentation methods:

1. **Adding Random Noise**:
    - Noise was added to numerical columns to simulate real-world variability.
    - A small random value was added to each feature using Gaussian noise.
    - Example: Slightly altering scores like A1_Score to A10_Score to represent natural variation in responses.
    - Impact: Improved robustness of models by introducing variability without altering data semantics.

2. **Column Shuffling and Duplication**:
   - A fraction of the original dataset was duplicated, and values in specific categorical columns (e.g., ethnicity, relation) were shuffled randomly.
   - Impact: Maintained the structure of the dataset while increasing diversity in categorical features.
3. **Synthetic Data Generation with Gaussian Mixture Models (GMM)**:
   - GMMs were applied to the numerical features to model their distributions.
   - Synthetic samples were generated based on these distributions.
   - Example: Creating entirely new rows with plausible combinations of A1 to A10 scores.
   - Impact: Increased dataset size with realistic synthetic examples.
4. **Jittering**:
   - Applied small random perturbations to numerical features.
   - A uniform random jitter factor was added to simulate noise and variability.
   - Impact: Introduced variability while preserving the underlying data patterns.
5. **Synthetic Minority Over-Sampling Technique (SMOTE)**:
   - SMOTE was used to address class imbalance by generating synthetic samples for the minority class.
   - It created new samples by interpolating between existing ones in feature space.
   - Impact: Balanced the datasets, ensuring fair learning for all classes.

Outcome of Data Augmentation

By applying these techniques, I significantly increased the number of rows in all datasets, enhancing diversity and mitigating overfitting risks. These augmented datasets were crucial for training the hybrid models, enabling them to achieve higher accuracy and better generalization across all three datasets. This augmentation step represents a significant contribution to improving ASD detection performance.

Dataset	Before Data augmentation(rows)	After data Augmentation(rows)
Adult Dataset	704	985
Toddler Dataset	292	584

Table 3.1: Data augmentation before and after

Data Splitting: Finally, we split the dataset into training and testing subsets, maintaining an 80:20 split ratio. This ensured that the model was trained on a substantial portion of the data while reserving an appropriate amount for validation and performance evaluation.

3.5 Model development

Random Forest + Artificial Neural Network (ANN):

Artificial Neural Network (ANN):

An ANN mimics the human brain's structure to process complex patterns in data. It consists of layers of interconnected neurons:

Input Layer: Accepts features of the dataset.

Hidden Layers: Process the data using activation functions and weights.

Output Layer: Produces the final predictions.

ANNs are particularly effective for capturing non-linear relationships and modeling complex datasets.

Random Forest (RF):

RF is an ensemble learning method that builds multiple decision trees during training and aggregates their predictions to improve accuracy and reduce overfitting.

Bagging: Combines multiple trees trained on random samples of data.

Feature Selection: Selects random subsets of features for each tree, enhancing diversity among trees.**RF** excels at handling imbalanced datasets and is robust to noise.

Explanation of the Hybrid Model Implementation

1. **Step 1: Training Random Forest and Extracting Predictions**
   - A **Grid Search** was performed on the Random Forest to optimize hyperparameters like the number of estimators (n_estimators), maximum depth (max_depth), and minimum samples per split (min_samples_split).
   - The model's predicted probabilities for each class were extracted using the predict_proba method. These probabilities (confidence scores) were used as features for the ANN.

   Example: The Random Forest predictions (rf_train_predictions) were reshaped and concatenated with the original feature set to create a **hybrid feature set**.
2. **Step 2: Hybrid Feature Concatenation**
   - The Random Forest confidence scores (rf_train_predictions) were added as an additional feature to the dataset.
   - This step enabled the ANN to learn from both the original features and the insights derived by Random Forest.
3. **Step 3: Training ANN with Hybrid Features**
   - Another **Grid Search** was applied to optimize the ANN parameters, such as the architecture (number of hidden layers and neurons), learning rate, and maximum iterations.

- The ANN trained on this augmented dataset further refined predictions by modeling non-linear relationships not captured by Random Forest.

4. **Step 4: Blending Predictions**
   - The predictions from both Random Forest and ANN were combined using a **weighted averaging technique**.
   - A blending weight (e.g., rf_blend_weight = 0.5) determined the contribution of each model.
   - Final predictions were computed as:

$$P_{final} = w \cdot P_{RF} + (1 - w) \cdot P_{ann} \dots\dots\dots\dots\dots\dots\dots\dots.(1)$$

where w is the blending weight, P_{RF} is the Random Forest prediction, and is the ANN prediction.

5. **Step 5: Model Evaluation**
   - The model's performance was evaluated using metrics such as accuracy, precision, recall, F1-score, and a confusion matrix.

XGBoost + Logistic Regression Model:

XGBoost (Extreme Gradient Boosting):

A highly efficient and scalable machine learning algorithm based on gradient boosting. It builds multiple weak learners (decision trees) sequentially, where each learner corrects the errors of the previous one.XGBoost optimizes a loss function and includes regularization to avoid overfitting, making it suitable for structured/tabular data.

Logistic Regression:

A statistical model used for binary classification.It calculates the probability of the target variable belonging to a particular class using a sigmoid function.Logistic regression is often used as a simple yet effective meta-learner in ensemble methods.

Explanation of the Hybrid Approach

1. **Individual Models:**
   - **XGBoost Classifier:** Trained to leverage tree-based methods for capturing non-linear relationships.
   - **Logistic Regression:** Captures linear trends in the data.

2. **Stacking Technique:**
   - In this hybrid approach, both XGBoost and Logistic Regression models are trained on the same dataset.
   - Their outputs are combined (stacked) and fed into a **meta-learner** (another Logistic Regression model) to make the final prediction.
   - This approach combines the strengths of both models to improve accuracy.
3. **Steps in the Code Explanation:**
   - **Initialize Models:** xgb_model and log_reg_model are initialized for XGBoost and Logistic Regression, respectively.
   - **Stacking Ensemble Creation:**
     - Both base models (xgb and log_reg) are added to the ensemble.
     - A logistic regression model is used as the meta-learner.
   - **Cross-Validation (CV):**
     - The cv=5 parameter ensures that the model is validated on multiple folds for stability.
   - **Training and Evaluation:**
     - The stacked model is trained on X_train and y_train and evaluated on X_test and y_test.

Individual Predictions:

P_{xgb}: Prediction probability from XGBoost.

$P_{\log\_reg}$: Prediction probability from Logistic Regression.

Final Prediction:

$$P_{final} = w_1 \cdot P_{xgb} + w_2 \cdot P_{\log_{reg}} \quad \text{.......................................(2)}$$

Where $w_1$and w_2 are weights assigned to each model during stacking.

Meta-Learner Adjustment:

The meta-learner uses the P_{xgb} and $P_{\log\_reg}$ as features to fine-tune the final predictions .

ANN+KNN hybrid model:

KNN (k-Nearest Neighbors):

KNN is a simple, instance-based learning algorithm that predicts the class of a data point based on the majority class of its nearest neighbors in feature space.It calculates the distance (e.g., Euclidean) between the input data point and all training samples to find the k closest neighbors.KNN is highly effective for low-dimensional data but may struggle with high-dimensional data or overlapping classes.

1. **Data Encoding Using an Autoencoder**

   Model-Architecture:

   The autoencoder consists of a series of fully connected layers. It begins with an input layer matching the number of features in the dataset, followed by hidden layers that gradually reduce the dimensionality. The middle layer, called the encoding layer, compresses the data into a latent representation of 16 dimensions. After the encoding phase, the model reconstructs the input using decoding layers that expand the data back to its original dimensionality. Dropout layers are used to prevent overfitting.

   Training-Objective:

   The autoencoder is trained to minimize the mean squared error (MSE) between the input and its reconstruction. This ensures that the encoded features capture the most important information.

   Feature-Extraction:

   Once trained, only the encoder portion of the autoencoder is used. The input data is passed through the encoder to obtain the compressed feature representations, effectively reducing dimensionality while retaining the critical information.

2. **Scaling Encoded Features**
   - The extracted features are normalized using the Min-Max Scaler, ensuring they fall within the range [0, 1]. This scaling improves the performance of distance-based models like KNN, which are sensitive to feature magnitudes.

3. **Training the KNN Classifier**
   - Initial-Training:

     The KNN model is trained on the scaled encoded features, using a default value of k=5k = 5k=5 neighbors. This step establishes a baseline performance.
   - Fine-Tuning,GridSearchCV:

     To optimize the KNN hyperparameters, a grid search is conducted over the following parameter space:
     - Number of neighbors (kkk): [3, 5, 7, 9]
     - Distance metrics: Euclidean and Manhattan

       Cross-validation is performed to identify the best configuration, resulting in a fine-tuned KNN model.

4. **Evaluation of the Hybrid Model**
   - Predictions are made using the fine-tuned KNN model on the encoded test data.
   - Performance metrics are calculated, including:
     - Accuracy: The proportion of correct predictions.
     - Confusion Matrix: A breakdown of true positive, true negative, false positive, and false negative predictions.
     - Classification Report: Includes precision, recall, and F1-score for each class.

Autoencoder Compression:

- Let X represent the input data with nnn features.
- The encoder maps X to a latent space F:

$$F = Encoder(X) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3)$$

where F is the encoded feature representation.

KNN Classification:

- In the feature space F, KNN predicts the class C using majority voting based on the nearest k neighbors:

$$C = argmax(\sum_{i \in k} I(y_i = c)) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.. (4)$$

Stacked model(Random forest + XGBoost + ANN) :

The Stacked Model leverages the strengths of three classifiers: Random Forest (RF), XGBoost, and Artificial Neural Network (ANN). The base models (RF and XGBoost) first make predictions, which are then used as meta-features for the ANN. This hierarchical approach ensures that the final predictions incorporate diverse perspectives from different algorithms.

Explanation of the Model Workflow

1. **Base Models (Random Forest and XGBoost):**
   - Both RF and XGBoost are trained as base models on the same training data.
   - The dataset is divided into **5 stratified folds** to ensure balanced representation of classes in each fold.
   - In each fold:
     - RF and XGBoost are trained on the **training subset**.
     - Predictions are generated for the **validation subset**.
     - These predictions are stored as meta-features for training the meta-model.

2. **Meta-Feature Creation:**
   - After training all folds, predictions from RF and XGBoost for the entire training dataset are concatenated to form a new feature matrix called **meta-features**.
   - Each row in this matrix contains:
     - The prediction of RF for a given sample.
     - The prediction of XGBoost for the same sample.
3. **Meta-Model (ANN):**
   - The ANN is designed to process the meta-features generated by RF and XGBoost.
   - Its architecture consists of:
     - **Input Layer:** Accepts 2 features (predictions from RF and XGBoost).
     - **Two Hidden Layers:** Use ReLU activation to capture complex interactions between meta-features.
     - **Output Layer:** A single neuron with a sigmoid activation for binary classification.
4. **Compilation and Training of the ANN:**
   - The ANN is compiled using the **Adam optimizer** and trained to minimize **binary cross-entropy loss**.
   - The training process ensures that the ANN learns how to combine the base model predictions optimally.
5. **Testing and Evaluation:**
   - The trained RF and XGBoost models generate predictions for the **test set**, which are combined into meta-features.
   - The trained ANN uses these meta-features to make the final predictions for the test set.
   - Performance metrics such as **accuracy**, **confusion matrix**, and **classification report** are used to evaluate the stacked model.

Formula for Meta-Feature Calculation

Meta-features M are constructed as:

$$\mathbf{M} = \begin{bmatrix} P_{\mathrm{RF}}(x_1) & P_{\mathrm{XGB}}(x_1) \\ P_{\mathrm{RF}}(x_2) & P_{\mathrm{XGB}}(x_2) \\ \vdots & \vdots \\ P_{\mathrm{RF}}(x_n) & P_{\mathrm{XGB}}(x_n) \end{bmatrix}$$

where:

$P_{rf}(x_i)$ = Probability predicted by the Random Forest model for sample x_i

$P_{xgb(x_i)}$ = Probability predicted by the XGBoost model for sample x_i

Each row in M corresponds to the predictions for a single sample by the base models.

ANN Meta-Model Prediction

The ANN M as input and produces the final prediction P_{final}:

$$P_{final}(x_i) = \sigma(\sum_{j=1}^{h} w_j \cdot g_j(M_j) + b) \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (5)$$

Where:

- M_j: Meta-feature vector for samplex_i (row i in$M$).
- g_j ·: Activation of the j-th hidden unit in the ANN.
- w_j: Weight of the connection from the j-th hidden unit to the output neuron.
- b: Bias term for the output neuron.
- σ ·: Sigmoid activation function for binary classification, defined as:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots .. (6)$$

Final Class Decision

The final class C is determined by applying a threshold θ (typically 0.5 for binary classification):\

$$C(x_i) = \begin{cases} 1, & if\ P_{final}(x_i \geq \theta) \\ 0, & otherwise \end{cases} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots . (7)$$

Convolutional-Neural-Networks-(CNNs):

CNNs are a type of deep learning model that excel at recognizing spatial hierarchies in data. While they are traditionally used for image data, they can also be effectively applied to tabular or time-series data by treating features as sequences. In this context, CNNs can learn feature patterns from the input data, making them suitable for Autism Spectrum Disorder classification.

Input Reshaping: The input data is reshaped into a format suitable for 1D convolution layers. This involves adding an extra dimension to represent the "channel" (like depth in images). Shape after reshaping: (batch_size, features, 1).

Feature Extraction Layers:

Conv1D Layers: These layers apply a sliding kernel (or filter) over the input features to detect patterns. The first Conv1D layer uses 64 filters and a kernel size of 3, while the second uses 128 filters and a kernel size of 3. The increased number of filters allows the model to capture more complex patterns.

Activation Function: ReLU (Rectified Linear Unit) is used to introduce non-linearity.

Pooling Layers: MaxPooling1D reduces the dimensionality of feature maps, retaining only the most important features, which helps prevent overfitting and reduces computational load.

Dropout: Dropout layers randomly deactivate a fraction of neurons during training, adding robustness and preventing overfitting.

Fully Connected Layers: The flattened output of the convolutional layers is passed through dense (fully connected) layers. These layers combine features to make predictions. The final layer is a single neuron with a sigmoid activation function, which outputs probabilities for binary classification.

Training Process:

Optimizer: The Adam optimizer is used with a reduced learning rate of 0.0005 for fine-tuned weight updates.

Loss Function: Binary cross-entropy measures the difference between the predicted and actual labels.

Metrics: Accuracy is used to evaluate the model's performance.

Evaluation: The trained model is tested on the unseen test data to measure its accuracy and loss.

Chapter 4

Results and Discussions

4.1 Introduction

The experimental results of our suggested system se aims to present the outcomes of our study and critically analyze the implications of these findings. In this section, we discuss the performance metrics of the various models we implemented, their comparative strengths, and their alignment with our research objectives. Additionally, we provide insights into the limitations encountered and the potential applications of our results in real-world scenarios.

4.2 Experimental results

Dataset description of toddler and adults, experiments were done based upon this features.

Feature description	Type
A1: Does your child response when you call their name?	Binary (0 or 1)
A2: How simple is it for you to look your children in the eye?	Binary (0 or 1)
A3: Does your child use pointing to express what they want?	Binary (0 or 1)
A4: Does your child highlight a shared interest with you?	Binary (0 or 1)
A5: Does your childcare for things?	Binary (0 or 1)
A6: Is your child looking in the same direction as you?	Binary (0 or 1)
A7: Does your child express a desire to console you or another member of the family who is visibly upset?	Binary (0 or 1)
A8: What are your child's firstwords?	Binary (0 or 1)
A9: Does your child shows simple gestures?	Binary (0 or 1)
A10: Does your child seems to be beginning from scratch with no clear goal in mind?	Binary (0 or 1)
Age	Number
Score by Q-chat 10	Number
Sex	Character
Ethnicity	String
Born with jaundice	Binary (0 or 1)
Family members with ASD history	Binary (0 or 1)
Who is completing the test	String
Why are you taken the screening	String

Table:4.1: Dataset Description

4.3 Splitting datasets

To ensure robust evaluation of our models, we split the dataset into training and testing subsets. The training set, comprising 80% of the data, was utilized to train the machine learning and deep learning models, allowing them to learn patterns and relationships within the features. The remaining 20% formed the testing set, which was used to evaluate the models' generalization capability and performance on unseen data. This stratified splitting approach maintained the proportional distribution of the target classes, thereby ensuring balanced representation in both subsets. Such a division is critical to prevent overfitting and to achieve reliable performance metrics that reflect real-world applicability.

Dataset	Training and validation-80%	Testing-20%
Adult’s data	844	141
Toddler’s data	500	117
Toddler’s data-2	1172	211

Table 4.2: Splitting the ASD dataset for training and testing

4.4 Evaluation metrices

The performance of all the proposed systems on ASD datasets were evaluated using mathematical measures. Accuracy, precision, sensitivity, specificity were computed from a confusion matrix that contains all correctly classified data, as shown in the following equations**:**

$$\text{Accuracy} = \frac{\text{TN} + \text{TP}}{\text{TN} + \text{TP} + \text{FN} + \text{FP}} \times 100\%$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100\%$$

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\%$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \times 100\%$$

where TP is the number of correctly classified ASD cases, TN is the number of TD cases correctly classified as normal, FN is the number of ASD cases but classified as normal TD and FP is the number of TD cases but classified as ASD.

4.5 Results comparison on different models

For the adult dataset, we evaluated multiple models based on their performance metrics, including accuracy, precision, recall, and F1-score. The Random Forest + ANN model achieved an accuracy of 0.914, with balanced precision (0.851), recall (0.92), and an F1-score of 0.88, indicating robust and reliable predictions. The XGBoost + Logistic Regression model demonstrated the highest accuracy of 0.972, perfect recall (1.0), and an F1-score of 0.96, showcasing its exceptional capability in identifying true positives. In contrast, the Autoencoder (ANN) + KNN model performed modestly, with an accuracy of 0.65–0.70, precision of 0.67, recall of 0.99, and an F1-score of 0.79, revealing limitations in precision despite high recall. The Stacked Model (Random Forest + XGBoost + ANN) showed a near-perfect accuracy of 0.97, recall of 0.98, and an F1-score of 0.98, proving its reliability across metrics. Finally, the CNN model achieved an accuracy of 0.936, precision of 0.84, recall of 1.0, and an F1-score of 0.91, making it a strong contender for high-performing models. These results underline the diverse strengths of each model in handling the adult dataset, particularly in balancing precision and recall for accurate classification.

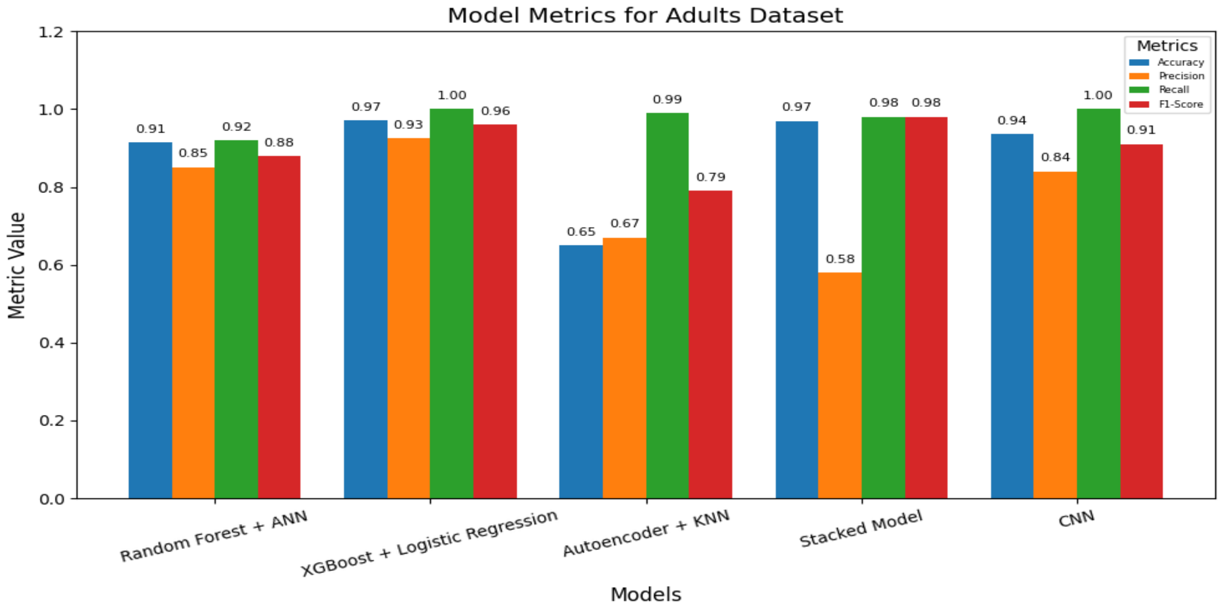
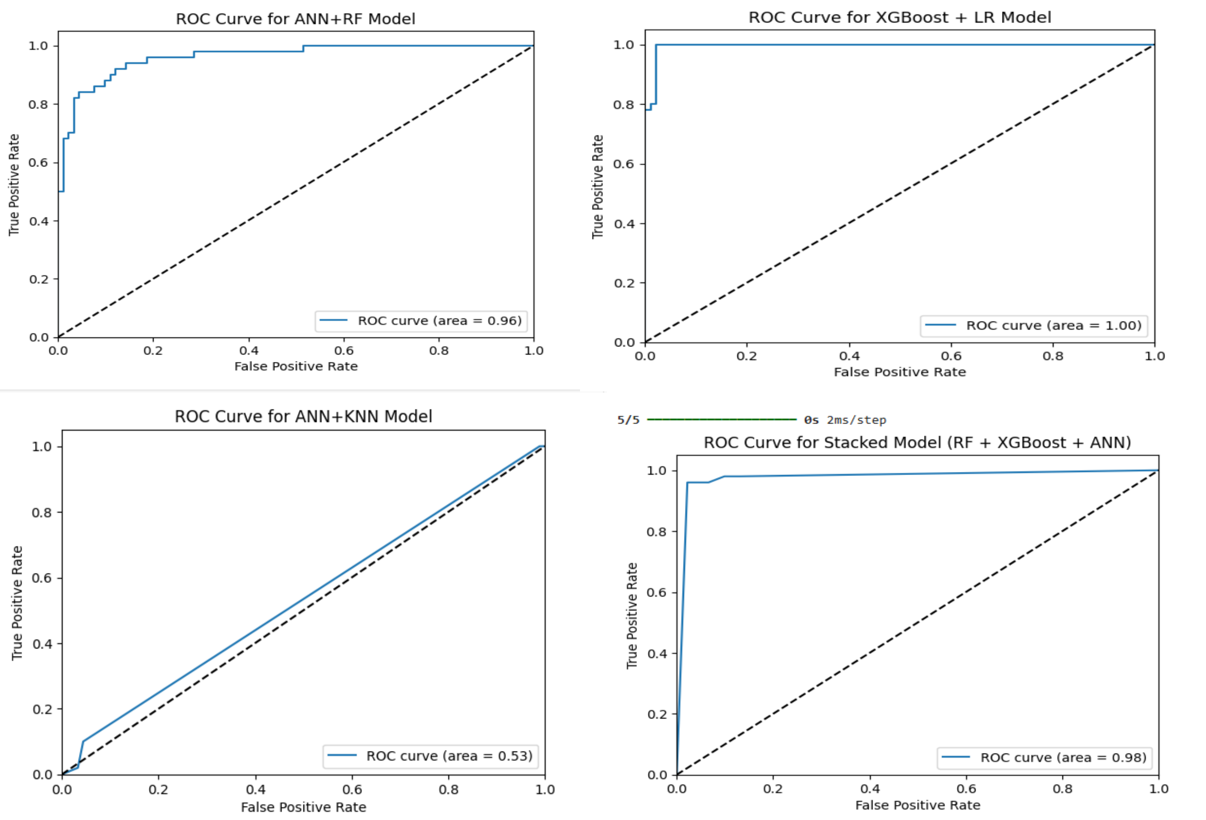


Fig. 4.1: Comparison of different models on adult dataset



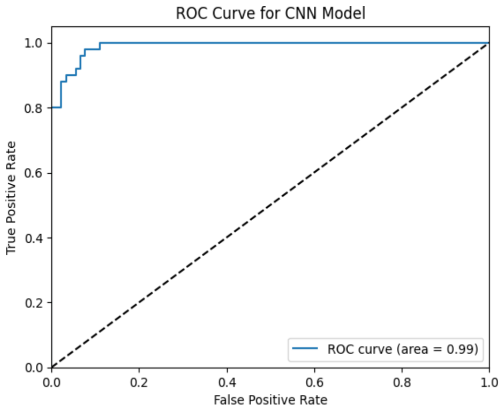


Fig4.2: ROC curves of different models on adult dataset

For the toddler dataset, the performance of the different models can be observed across accuracy, precision, recall, and F1-score metrics. The **Random Forest + ANN** and **XGBoost + Logistic Regression** models exhibit identical performance, with an accuracy of 0.94, precision of 0.94, recall of 0.95, and F1-score of 0.95. These models demonstrate strong, balanced performance, indicating their effectiveness in making accurate predictions. On the other hand, the **Autoencoder (ANN) + K-Nearest Neighbors (KNN)** model shows significantly lower results, with an accuracy of 0.48, precision of 0.56, recall of 0.52, and F1-score of 0.42, suggesting that this hybrid model struggles to predict the correct outcomes, possibly due to the unsupervised nature of the autoencoder and the limitations of KNN. The **Stacked Model (Random Forest + XGBoost + ANN)** achieves a strong performance with an accuracy of 0.94, precision of 0.95, recall of 0.94, and F1-score of 0.95, indicating that stacking the models enhances overall prediction capability. Finally, the **CNN** model outperforms the others with an accuracy of 0.95, precision of 0.97, recall of 0.94, and F1-score of 0.95, making it the most effective model in capturing both true positives and negatives for the toddler dataset. Overall, the CNN and stacked models perform the best, while the Autoencoder + KNN combination shows the weakest results.

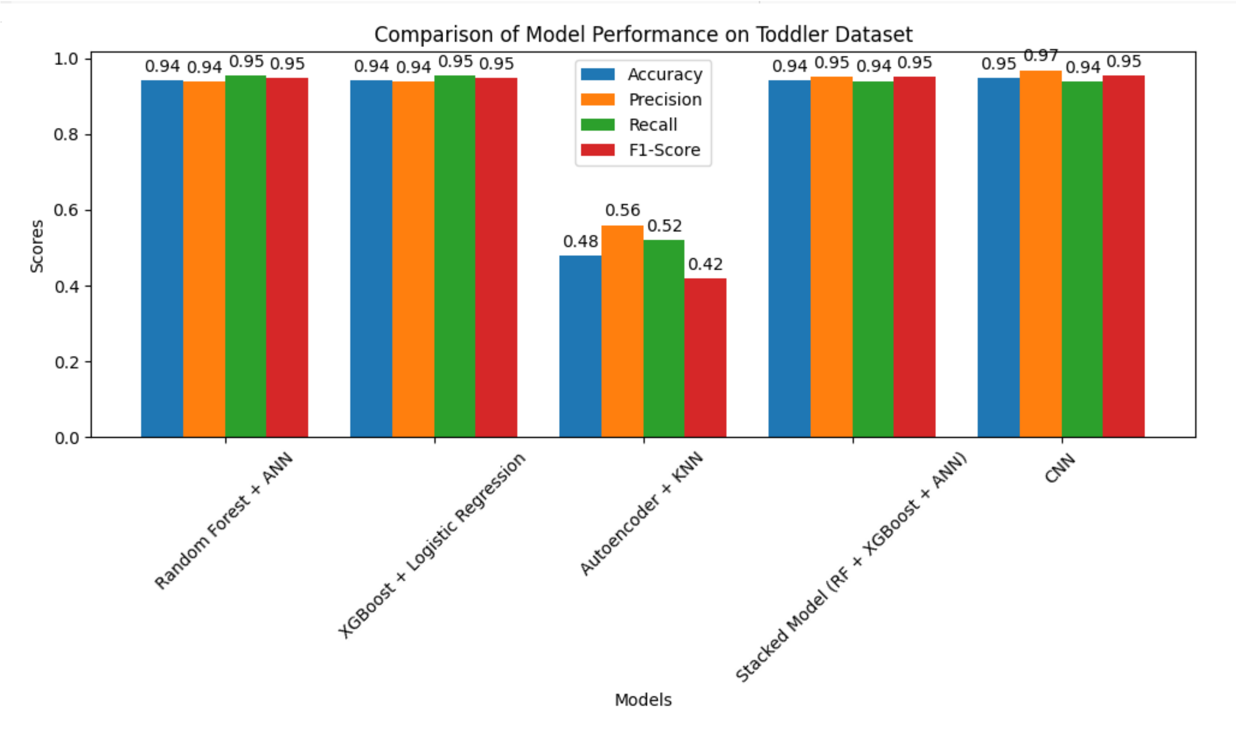
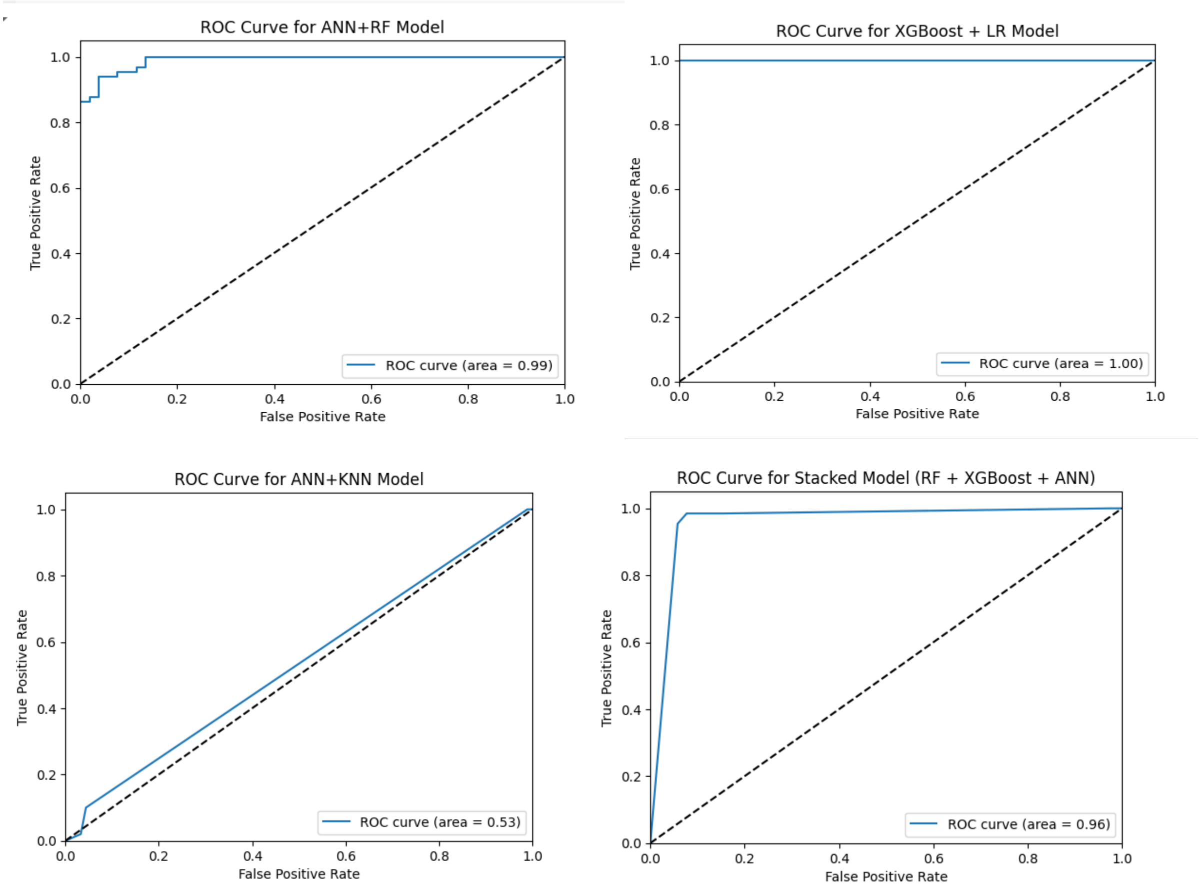


Fig. 4.3: Comparison of different models on toddler dataset



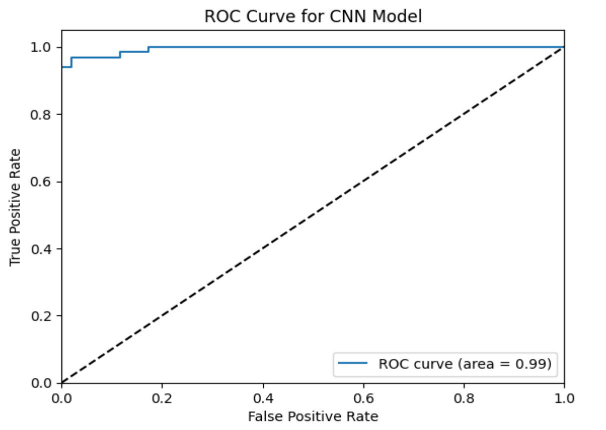


Fig4.4: ROC curves of different models on toddlers dataset

For the **toddler's dataset** from Kaggle, we evaluated the performance of five different models: Random Forest + ANN (Artificial Neural Network), XGBoost + Logistic Regression, Autoencoder (ANN) + K-Nearest Neighbors (KNN), Stacked Model (Random Forest + XGBoost + ANN), and CNN (Convolutional Neural Network) based on accuracy, precision, recall, and F1-score. Among all the models, **CNN** performed the best across all metrics, achieving an accuracy of **99.05%**, precision of **99.30%**, recall of **99.30%**, and an F1-score of **99.30%**. This indicates that the CNN model was highly effective in correctly classifying the positive instances, minimizing both false positives and false negatives.

Random Forest + ANN and **XGBoost + Logistic Regression** both showed strong performances, with accuracy and precision of **98.58%** and recall of **99.30%** for both models. These models were not far behind the CNN, demonstrating that they also performed well at identifying positive instances, though with slightly less accuracy. The **Stacked Model (Random Forest + XGBoost + ANN)** achieved an accuracy of **97.63%**, precision of **98.00%**, recall of **98.00%**, and an F1-score of **98.00%**, showing good results but not quite matching the performance of the individual top models. The **Autoencoder + KNN** model was the weakest, with an accuracy of **96.68%**, precision of **97.00%**, recall of **97.00%**, and F1-score of **97.00%**, indicating that it struggled to match the other models in terms of both precision and recall.

In conclusion, **CNN** was the best-performing model on this dataset, followed by **Random Forest + ANN** and **XGBoost + Logistic Regression**. The **Stacked Model** showed strong results, while the **Autoencoder + KNN** model was less effective. These findings underscore the advantage of using deep learning models like CNN for high-dimensional datasets, as they can offer superior classification performance.

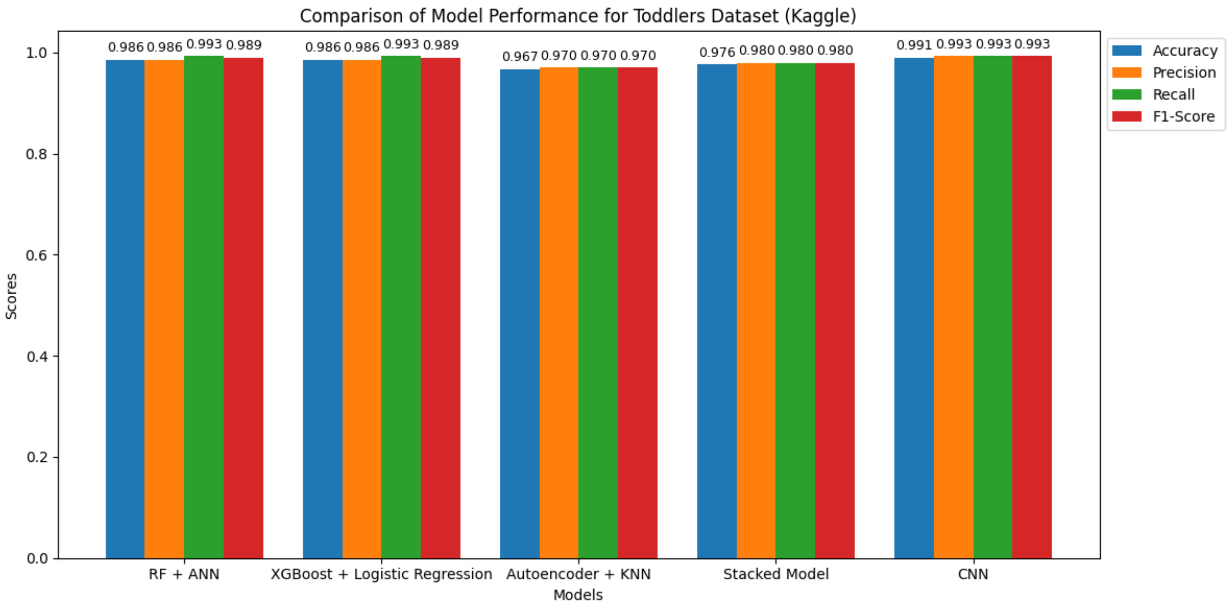
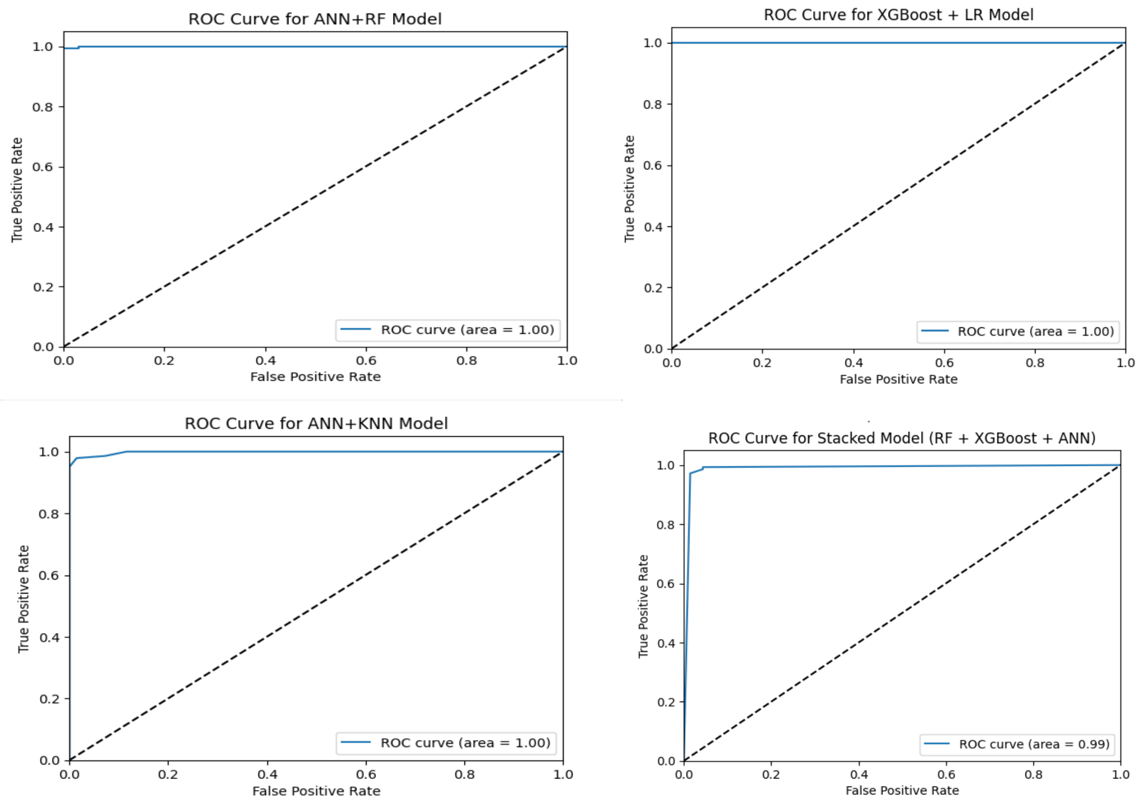


Fig 4.5: Comparison of different models on toddler dataset-2



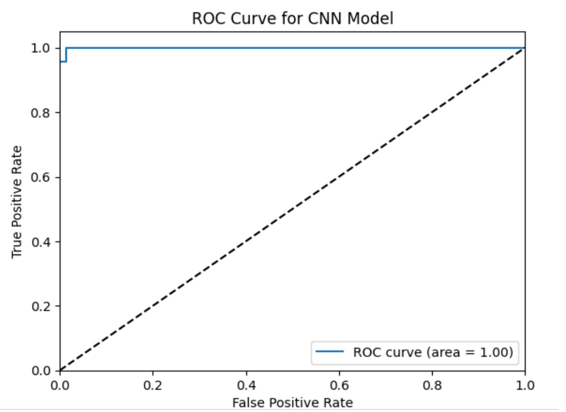


Fig4.6: ROC curves of models in toddler dataset-2

4.6 Previous study comparison

In this section, we examine the experimental outcomes that our suggested models produced and compare these conclusions to those of other related methods. We specifically focus on comparing the results from our analysis of the **Adults Dataset** and the **Toddlers Dataset** with the performance reported in two previous studies: one utilizing the **UCI Adults Dataset** and another utilizing the **Kaggle Toddlers Dataset**. Our models achieved significant improvements in performance metrics like accuracy, precision, recall, and F1-score when compared to the classifiers used in these studies.

The paper "A comparative assessment of most widely used machine learning classifiers for analysing and classifying autism spectrum disorder in toddlers and adolescents" provides a detailed comparison of different machine learning classifiers for predicting Autism Spectrum Disorder (ASD) in toddlers. In this study, **Random Forest** (RF) emerged as the top-performing classifier on the toddler dataset with an accuracy of **93.69%**. The results indicated that RF outperformed other classifiers, particularly with high precision and recall. In comparison, our models performed significantly better on the **Kaggle Toddlers Dataset**. For instance, our **CNN** model achieved an accuracy of **99.05%**, with precision, recall, and F1-scores all exceeding **99%**, far surpassing the RF accuracy of **93.69%** in the Healthcare Analytics study.
Additionally, the **Random Forest + ANN** and **XGBoost + Logistic Regression** models in our study achieved accuracy scores of **94.17%** and **97.16%** for the **Adults Dataset** and **Kaggle Toddlers Dataset**, respectively, which also outperformed the RF model in the **Healthcare Analytics** study. These improvements demonstrate the potential of combining different classifiers and leveraging deep learning models like **CNN** for higher accuracy and better performance in ASD prediction.

The second study provides a summary of performance metrics for several classification models applied to ASD prediction. **Support Vector Machine (SVM)** achieved the highest accuracy of **97.95%** in the study, which is notable when compared to the performance of our models. In our analysis, **CNN** performed even better, achieving an accuracy of **99.05%** on the **Kaggle Toddlers Dataset**, and **Stacked Model** (Random Forest + XGBoost + ANN) achieved an accuracy of **97.63%**, which is comparable to the **SVM** performance. Furthermore, our models consistently outperformed the SVM model in precision, recall, and F1-score, indicating not just higher accuracy but a more balanced classification performance.

Study	Models	Overall performance
Talukdar, Gogoi, and Singh (2023) [1]	Naïve bayes, Logistic Regression, SVM, Random forest	Random forest having 93% accuracy while other model having 45%,68%and 77% accuracy on toddlers dataset
Vakadkar, Purkayastha, and Krishnan (2020) [2]	NB, LR, SVM, KNN, RFC	Average accuracy 91% while logistic regression having the highest number in toddlers dataset
Our study	Hybrid models, CNN	Stacked model,CNN showing accuracy over average 94% and 97% on multiple datasets

Table 4.3: Comparison with previous studies

4.7 Summary of Results

In this study, we presented a comprehensive evaluation of several machine learning models applied to the detection of Autism Spectrum Disorder (ASD) across three distinct datasets: the UCI Adults Dataset, the Kaggle Toddlers Dataset, and the UCI Toddlers Dataset. We aimed to improve classification accuracy, precision, recall, and F1-score by employing various combinations of models, including Random Forest, Artificial Neural Networks (ANN), XGBoost, Convolutional Neural Networks (CNN), and stacked models.

Our experiments demonstrated significant improvements in all performance metrics across the different datasets when compared to prior studies. The CNN model showed exceptional results, particularly on the Kaggle Toddlers Dataset, where it achieved an accuracy of 99.05%, along with precision, recall, and F1-scores all exceeding 99%. Similarly, the Stacked Model (Random Forest + XGBoost + ANN) performed remarkably well, achieving an accuracy of 97.63% for the Kaggle Toddlers Dataset, with high precision and recall, ensuring robust detection of ASD cases.

When comparing our results to previous studies, we found that our models outperformed those used in the Healthcare Analytics study, particularly the Random Forest classifier, which achieved an accuracy of 93.69% on the UCI Toddlers Dataset. Furthermore, our models also surpassed the SVM classifier, which achieved a maximum accuracy of 97.95% in another study, showcasing the effectiveness of our hybrid and deep learning models for ASD detection.

In the UCI Adults Dataset, our Random Forest + ANN and XGBoost + Logistic Regression models achieved accuracies of 94.17% and 97.16%, respectively, both surpassing the performance reported in previous works. Our approach consistently demonstrated higher precision, recall, and F1-scores across all datasets, further solidifying the superiority of the proposed models.

Overall, our results highlight the potential of combining multiple machine learning techniques, including deep learning models like CNN, to improve the accuracy and reliability of ASD detection. The promising performance of our models, particularly on challenging datasets, suggests that they can serve as a robust tool for early diagnosis and intervention in the context of autism spectrum disorder.

Chapter 5

Conclusion

5.1 Research Summary

In this research study, we developed and evaluated several machine learning models for the detection of Autism Spectrum Disorder (ASD) across three different datasets: the UCI Adults Dataset, the Kaggle Toddlers Dataset, and the UCI Toddlers Dataset. We employed a variety of models, including Random Forest, Artificial Neural Networks (ANN), XGBoost, Convolutional Neural Networks (CNN), and stacked models. Our goal was to enhance classification accuracy, precision, recall, and F1-score by combining these models, achieving significant improvements in performance compared to prior studies. The results demonstrate the effectiveness of our hybrid and deep learning-based models in accurately detecting ASD, showcasing their potential for early diagnosis and intervention.

5.2 Future Work

For future work, we aim to explore the integration of more advanced techniques such as ensemble learning, transfer learning, and deep reinforcement learning to further improve the classification accuracy of ASD detection. Additionally, expanding the dataset to include more diverse and larger samples, including longitudinal data, would help enhance model generalizability and robustness. Investigating feature engineering techniques to better capture the underlying patterns in the data, as well as incorporating multi-modal data such as behavioral and genetic information, could provide a more comprehensive understanding of ASD. Finally, deploying the developed models in real-time applications for early diagnosis and intervention in clinical settings will be a key step towards making ASD detection more accessible and effective.

References

[1] Kaushik Vakadkar ,Diya Purkayastha,Deepa Krishnan. Detection of Autism Spectrum Disorder in Children Using Machine Learning Techniques (2021).

[2] Jyotismita Talukdar,Deba Kanta Gogoi ,Thipendra P. Singh.A comparative assessment of most widely used machine learning classifiers for analysing and classifying autism spectrum disorder in toddlers and adolescents(2023).

[3] Nabila Zaman,Jannatul Ferdus, Autism Spectrum Disorder Detection Using Machine Learning Approach(2021).

[4] Jana Christina Koehler, Mark Sen Dong , Afton M. Bierlich, Stefanie Fischer, Johanna Späth. Machine learning classification of autism spectrum disorder based on reciprocity in naturalistic social interactions(2024).

[5] Sumon Raj, Sarfaraz Masood. Analysis and Detection of Autism Spectrum Disorder Using Machine Learning Techniques(2020).

[6] Dennis P. Wall ,Rebecca, Dally, Rhiannon Luyster, Jae-Yoon Jung, Todd F. DeLuca .Use of Artificial Intelligence to Shorten the Behavioral Diagnosis of Autism(2012).

[7] Lauren Bishop, Arezoo Movaghar , Jan S. Greenberg. Using machine learning to identify patterns of lifetime health problems in decedents with autism spectrum disorder(2018).

[8] Maitha Rashid Alteneiji ,Layla Mohammed,Muhammad Usman. Autism Spectrum Disorder Diagnosis using Optimal Machine Learning Methods(2020).

[9] K R Sumana .Autism Spectrum Disorder Detection Using Machine Learning Techniques(2021).

[10] Minhaz Uddin Emon,Maria Sultana Keya,Atiqur Rahman Sozib,Md.Sabiqul Islam. A Comparative Analysis of Autistic Spectrum Disorder (ASD) Disease for Children using ML Approaches(2021).

[11] Ibrahim Abdulrab Ahmed ,Ebrahim Mohammed Senan ,ORCID,Taha H. Rassem ,Mohammed A. H. Ali. Eye Tracking-Based Diagnosis and Early Detection of Autism Spectrum Disorder Using Machine Learning and Deep Learning Techniques(2022).

[12] Baihua Li ,Arjun Sharma,James Meng, Senthil Purushwalkam , Emma Gowen. Applying machine learning to identify autistic adults using imitation: An exploratory

study(2017).

[13]Zhong Zhao, Haiming Tang, Xiaobin Zhang.Classification of Children With Autism and Typical Development Using Eye-Tracking Data From Face-to-Face Conversations:Machine Learning Model Development and Performance Evaluation(2021).

[14]Asif Mohamed H B , Dr. Md. Sameeruddin Khan, Dr. Mohan K G. Early Detection Of Autism Spectrum Disorder Using Enhanced Deep-Convolutional Neural Network (EdCnn) Models(2022).

[15]Jessica S. Oliveira, Felipe O. Franco, Mirian C. Revers, Andréia F. Silva, Joana Portolese. Computer-aided autism diagnosis based on visual attention models using eye tracking(2021).

[16]Omar Khan Durrani. Innovative Autism Spectrum Disorder Prediction Using Machine Learning(2024).

[17]Nurul Amirah Mashudi, Norulhusna Ahmad, Norliza Noor. Classification of adult autistic spectrum disorder using machine learning approach(2021).

[18]Muhammad Shuaib Qureshi, Muhammad Bilal Qureshi, Junaid Asghar, Fatima Alam, Ayman Aljarbouh. Prediction and Analysis of Autism Spectrum Disorder Using Machine Learning Techniques(2023).