



الجمهورية الجزائرية الديمقراطية الشعبية
People's Democratic Republic of Algeria



وزارة التعليم العالي والبحث العلمي

Ministry of Higher Education and Scientific Research

جامعة غرداية

University of Ghardaia

كلية العلوم والتكنولوجيا

Faculty of Science and Technology

قسم الرياضيات والإعلام الآلي

Department of Mathematics and Computer Science

مخبر الرياضيات والعلوم التطبيقية

Mathematics and Applied Sciences Laboratory

Registration n°:

...../...../...../...../.....

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

Master

Domain: Mathematics and Computer Science

Field: Computer Science

Specialty: Intelligent Systems for Knowledge Extraction

Topic

Developing a Chatbot for Detecting Autism Spectrum Disorder (ASD) based on Deep Learning

Presented by:

Rania HACHEMI

Publicly defended on July 01, 2024

Jury members:

MR. SLIMANE OULAD-NAOUI	MCB	Univ. Ghardaia	President
MRS. NACERA BRAHIM	MAA	Univ. Ghardaia	Examiner
MR. HOUSSEM EDDINE DEGHA	MCB	Univ. Ghardaia	Supervisor

Academic Year: 2023/2024

Dedication

First and foremost, I thank Allah for granting me the stamina and courage to realize my dreams.

To my grandmother Fatima, I cherish you dearly. Your energy, strength, determination, and success have always been my source of inspiration.

To my beloved parents, whose unwavering love, support, and sacrifices have been the foundation of my journey. Your belief in me, encouragement, and constant presence have propelled my thesis achievement; you were the guiding light on my path. Thank you for being my pillars of strength; I am forever grateful for your love and guidance. May Allah bless you with long life and happiness.

To my sweetly sisters Djihad, Israa, Rawane and the Smart brother Abdelkader thank you for your invaluable support throughout my life.

To the spirit of my late Professor, Abdulnour LONJ, who imparted upon me the essential lessons of life, university experience, and exemplified the epitome of kindness and dedication as an educator, I dedicate this work.

To Dr. Housseem Eddine DEGHA, thank you for your immense assistance and motivation, which empowered me to complete this work. I appreciate the trust you placed in me by supervising and guiding me throughout this journey.

To my dear friends Widad BETTACHE, Meriem BENAJJA, and Nour Elhouda RAMDANI, your guidance and engagement have contributed significantly to my growth as a scientist. Your inspiration pushes me to strive for excellence, continuous learning, and creativity.

Congratulations

Rania HACHEMI



Acknowledgement

To my dear family, who have been my support and strength in every step I have taken. Thank you for the love and care that have always surrounded me, and for the unparalleled dedication and patience.

Dr. Housseem Eddine DEGHA, was an amazing personal advisor, always there for me with his advice and support over the past three years. He has had the greatest impact on my academic life, and whenever I asked for help, he was ready to support and assist me.

To the esteemed committee, for their valuable guidance and direction, which had a significant impact on the completion of this thesis.

To my Professor Bachir MOUHOUB, may God heal him, who spared no effort in teaching and guiding me, he was a role model in knowledge and ethics.

To the esteemed Professor Fatima Zohra LAALLAM , for her continuous support and wise guidance that illuminated my path.

To my dear friends, thank you for standing by my side and for your constant encouragement, especially Fatima BENSALAM and Rawan DEGHA, who have always been a source of inspiration and support.

We appreciate the efforts of DzSoft.

I extend my heartfelt thanks and gratitude to all of you, for you were partners in every success I achieved, and you have a great credit for this accomplishment.

Rania Hachemi

Abstract

In educational settings early diagnosing of Autism Spectrum Disorder (ASD) is crucial since it expedites and in more effective way support is provided to affected children. Educators may deploy effective diagnosis technologies in an effort to make educational sitting inclusive. Although the symptoms may vary and range from a mild to a severe ASD particularly the cognitive, linguistic, communication, and social domains are very much affected. Whilst ASD is a non-cure problem, early detection will always be the first step to intervention. This research utilises messenger chatting in order to detect ASD for an early stage. The chatbot learns from is conversations and compares them with the ones considered typical to people with risky conditions of ASD. Later, an efficient machine learning model is trained by a dataset which includes the chatbot dialogues with both ASD and non-ASD toddlers to differentiate toddlers in the possible risky state by the different features. The results of mentioned research conclude that the proposed chatbot pattern is more precise as compared with traditional types of screening tools to detect toddlers with high ASD risk.

Keywords: Early Childhood Education, Parental Engagement, Autism Spectrum Disorder, Toddlers, Artificial Intelligence, T ASD-Dataset, ASDetect-Bot.

Résumé

Dans les environnements éducatifs, le diagnostic précoce du Trouble du Spectre Autistique (ASD) est crucial car il permet de fournir un soutien plus rapide et plus efficace aux enfants touchés. Les éducateurs peuvent déployer des technologies de diagnostic efficaces pour rendre les séances éducatives inclusives. Bien que les symptômes puissent varier et aller d'un trouble du spectre autistique (ASD) léger à sévère, les domaines cognitifs, linguistiques, de communication et sociaux sont très touchés. Bien que l'ASD soit un problème non curable, la détection précoce sera toujours la première étape de l'intervention. Cette recherche utilise pour la première fois l'intelligence artificielle (IA) pour détecter le trouble du spectre autistique (ASD) à un stade précoce par messagerie et apprentissage automatique. Le chatbot apprend de ses conversations et les compare à celles considérées comme typiques des personnes présentant des conditions à risque d'ASD. Ensuite, un modèle efficace d'apprentissage automatique est entraîné par un ensemble de données qui inclut les dialogues du chatbot avec des tout-petits présentant ou non un ASD pour différencier les tout-petits dans l'état potentiellement à risque par les différentes caractéristiques. Les résultats de cette recherche concluent que le modèle de chatbot proposé est plus précis par rapport aux types traditionnels d'outils de dépistage pour détecter les tout-petits présentant un risque élevé d'ASD.

Mots clés: Éducation de la Petite Enfance, Engagement Parental, Détection Précoce des Troubles du Spectre Autistique, Tout-petits, Intelligence Artificielle, TASD-Dataset, ASDetectBot.

الملخص

في البيئات التعليمية، يعتبر التشخيص المبكر لاضطراب طيف التوحد أمرًا حاسمًا لأنه يسرع ويوفر دعمًا أكثر فعالية للأطفال المصابين. يمكن للمعلمين نشر تقنيات التشخيص الفعالة في محاولة لجعل البيئة التعليمية شاملة. على الرغم من أن الأعراض قد تختلف وتتراوح من توحد خفيف إلى شديد، فإن المجالات الإدراكية، اللغوية، التواصلية، والاجتماعية تتأثر بشكل كبير. في حين أن التوحد هو مشكلة لا علاج لها، فإن الكشف المبكر سيكون دائمًا الخطوة الأولى للتدخل. يستخدم هذا البحث الدردشة عبر المراسلة للكشف عن التوحد في مرحلة مبكرة. يتعلم الروبوت من محادثاته ويقارنها بالمحادثات التي تعتبر نموذجية للأشخاص الذين يعانون من حالات خطر التوحد. لاحقًا، يتم تدريب نموذج فعال للتعلم الآلي بواسطة مجموعة بيانات تتضمن حوارات الروبوت مع الأطفال المصابين بالتوحد وغير المصابين به للتمييز بين الأطفال في حالة الخطر المحتمل بناءً على الميزات المختلفة. نتائج البحث المذكور تستنتج أن نمط الروبوت المقترح أكثر دقة بالمقارنة مع الأنواع التقليدية من أدوات الفحص للكشف عن الأطفال ذوي خطر التوحد العالي.

الكلمات المفتاحية: التعليم في مرحلة الطفولة المبكرة، مشاركة الوالدين، اضطراب طيف التوحد، الأطفال الصغار، الذكاء الاصطناعي.

Contents

List of Figures	vi
List of Tables	viii
List of Acronyms	ix
Introduction	1
1 Basic Concepts	5
1.1 Definition of Early Childhood Education (ECE)	6
1.1.1 Early Childhood	6
1.1.2 Core Principles of Early Childhood Education	7
1.1.3 Approaches to Early Childhood Education	8
1.1.4 Impact of Early Childhood Education	8
1.2 Definition of Parental Engagement	8
1.2.1 Characteristics of Parental Engagement	9
1.2.2 Benefits of Parental Engagement	9
1.3 Impact of Parental Engagement in Early Childhood Education	9
1.3.1 Cognitive and Academic Benefits	10
1.3.2 Social-Emotional Development and Well-being	10
1.3.3 Long-Term Benefits and School Readiness	10
1.4 Application of Artificial Intelligence (AI) in Early Childhood Education	11
1.4.1 Techniques for Data Mining Exploration	13
1.4.2 Recent Advancements and Considerations	13
1.5 Autism Spectrum Disorder: Understanding Diverse Learning Needs in Early Childhood Education	13
1.5.1 Understanding Autism Spectrum Disorder	14

1.5.2	Importance of Early Identification and Intervention	14
2	State Of The Art	15
2.1	Introduction	16
2.2	Related Works	16
2.3	Conclusion	20
3	Methodology and System Design	21
3.1	ASDetectBot Overview	22
3.2	System Architecture Overview	24
3.2.1	User Interface (GUI)	24
3.2.2	Backend Server	25
3.2.3	Database	25
3.2.4	Text Mining and Preprocessing	25
3.2.5	Deep Learning Model for ASD Detection	25
3.2.6	Coordination Model	26
3.2.7	Web Access	26
3.2.8	System Integration and Middleware	26
3.3	System Diagrams	26
3.3.1	Use Case Diagram	27
3.3.2	Activity Diagram	27
3.3.3	Sequence Diagram	28
3.3.4	Class Diagram Description	28
3.4	ASDetectBot Database Design	29
3.4.1	Description of the ASD Analysis Table	29
3.4.2	Description of the ASD Metrics Table	29
3.4.3	Description of the Children Table	30
3.4.4	Description of the Conversations Table	30
3.4.5	Description of the Messages Table	31
3.4.6	Description of the Users Table	31
4	Implementation	33
4.1	Data Description and Embedding Methods	34
4.1.1	Dataset Creation and Description	34

4.1.2	Data Collection and Preprocessing	35
4.1.2.1	Features Extraction	36
4.1.2.2	Generating Data	36
4.1.2.3	Collection	36
4.1.2.4	Pre-Processing	36
4.1.2.5	Final Dataset	36
4.1.2.6	Data Publication and Mendeley Integration	36
4.1.3	Embedding Methods	37
4.1.3.1	Label Encoding	37
4.1.3.2	TF-IDF Vectorization	37
4.2	Data Pre-processing	37
4.2.1	Importing Necessary Libraries	38
4.2.2	Reading Data	38
4.2.3	TF-IDF Vectorization	39
4.2.4	Label Encoding	39
4.2.5	Concatenation and Saving	40
4.3	Data Splitting	41
4.4	Evaluation Metrics for Autism Detection	42
4.4.1	Accuracy	42
4.4.2	Recall (True Positive Rate)	42
4.4.3	Precision (Positive Predictive Value)	42
4.4.4	F1 Score	43
4.5	Comparison of Machine Learning Models	43
4.5.1	SVM Model	43
4.5.2	Artificial Neural Network (ANN) Model	44
4.5.2.1	Model Architecture	44
4.5.2.2	Model Compilation and Training	45
4.5.2.3	Model Evaluation	46
4.5.3	Convolutional Neural Network (CNN) Model	47
4.5.3.1	Data Preprocessing	47
4.5.3.2	Model Architecture	48
4.5.3.3	Model Compilation and Training	48

4.5.3.4	Model Evaluation	49
4.5.4	Recurrent Neural Network (RNN) Model	50
4.5.4.1	Model Architecture	50
4.5.4.2	Model Compilation and Training	50
4.5.4.3	Model Evaluation	50
4.5.5	Long Short-Term Memory (LSTM) Model	51
4.5.5.1	Model Architecture	51
4.5.5.2	Model Compilation and Training	52
4.5.5.3	Model Evaluation	52
4.5.6	Gated Recurrent Unit (GRU) Model	53
4.5.6.1	Model Architecture	53
4.5.6.2	Model Compilation and Training	54
4.5.6.3	Model Evaluation	54
4.6	Chatbot Development	55
4.6.1	Express.js Framework	55
4.6.2	Server.js File Overview	56
4.6.3	signup.js File Overview	60
4.6.4	Script.js File Overview	60
4.6.5	HTML Part	64
4.6.6	CSS Part	65
4.6.7	Dataset.csv	66
4.6.8	Setup Instructions	66
4.7	ASDetectBot APP	67
4.7.1	User Login	67
4.7.2	User Sign Up	68
4.7.3	Chat Interface	68
4.7.4	Chatbot Response Mechanism	68
4.7.5	Parental Controls	69
4.7.6	Child Creation	69
4.7.7	Chat History Access	70
5	Result and Evaluation	72
5.1	Questionnaire's Design	73

5.1.1	Questionnaire Design Rationale	73
5.1.2	Objective	73
5.1.3	Relevance to ASD Assessment in Toddlers	73
5.1.4	Discussion of Validity and Reliability	76
5.2	Interpretation Guidelines for ASD Test Questionnaire Scores	76
5.2.1	Age-Appropriate Variations	76
5.2.2	Scoring Guidelines	77
5.2.3	Scores and ASD Indicators	79
5.2.4	Ethical Considerations in ASD Assessments	80
5.3	Training and Validation Metrics	81
5.3.1	Discussion of Results of ANN Model	81
5.3.2	Discussion of Results of CNN Model	82
5.3.3	Discussion of Results of RNN Model	84
5.3.4	Discussion of Results of LSTM Model	86
5.3.5	Discussion of Results of GRU Model	87
	Conclusion and Perspectives	91
	References	93

List of Figures

1.1 Stages of Early Childhood Development	7
1.2 Core Principles of Early Childhood Education	8
3.1 ASDetectBot App Architecture	24
3.2 Use Case Diagram for ASDetectBot System	27
3.3 Activity Diagram for ASDetectBot System	27
3.4 Sequence Diagram for ASDetectBot System	28
3.5 Class Diagram for ASDetectBot System	28
4.1 Data Process	35
4.2 Dataset Downloads	37
4.3 Reading Data	39
4.4 Data after Label Encoding	40
4.5 Data after excluding the 'Text' column	40
4.6 Data after Pre-processing	41
4.7 Dataset Independents Variables	41
4.8 Dataset Dependent Variable	42
4.9 User Login Interface	67
4.10 User Sign Up Interface	68
4.11 Chat Interface	68
4.12 Response Mechanism	69
4.13 Parental Controls	69
4.14 Child Creation Interface	70
4.15 Chat History Access Interface	70
5.1 Relationship Between Age and ASD Score	80
5.2 Training and Validation Accuracy of ANN Over Epochs	81

5.3	Training and Validation Loss of ANN over Epochs	81
5.4	Training and Validation Accuracy of CNN Over Epochs	83
5.5	Training and Validation Loss of CNN Over Epochs	83
5.6	Training and Validation Accuracy of RNN Over Epochs	84
5.7	Training and Validation Loss of RNN Over Epochs	85
5.8	Training and Validation Accuracy of LSTM Over Epochs	86
5.9	Training and Validation Loss of LSTM over Epochs	86
5.10	Training and Validation Accuracy of GRU over Epochs	87
5.11	Training and Validation Loss of GRU over Epochs	88

List of Tables

2.1	Comparison of AI Studies in Education and Social Dynamics	18
2.2	Comparison of AI Studies in ASD Management and Detection . . .	19
4.1	Count of ASD Labels	35
4.2	Preprocessing Techniques	38
4.3	Hyperparameters used for training the deeper neural network with dropout	45
4.4	Hyperparameters used for training the CNN model	49
4.5	Hyperparameters used for training the RNN model	50
4.6	Hyperparameters used for training the RNN/LSTM model	52
4.7	Hyperparameters used for training the RNN/GRU model	54
5.1	Questionnaire Data	74
5.2	Questionnaire Data .2	75
5.3	Key Interpretations for ASD Assessment Criteria	78
5.4	Scores and ASD Indicators	79
5.5	ANN Training and Validation Metrics	82
5.6	CNN Training and Validation Metrics	84
5.7	RNN Training and Validation Metrics	85
5.8	LSTM Training and Validation Metrics	87
5.9	GRU Training and Validation Metrics	88
5.10	Performance Metrics of Deep Learning Models on the T ASD Dataset	89

List of Acronyms

AAP *American Academy of Pediatrics*

AI *Artificial Intelligence*

AIED *Artificial Intelligence in Education*

ANN *Artificial Neural Network*

API *Application Programming Interface*

ASD *Autism Spectrum Disorder*

BERT *Bidirectional Encoder Representations from Transformers*

BMC *Business Model Canvas*

CNN *Convolutional Neural Network*

CSS *Cascading Style Sheets*

CSV *Comma-Separated Values*

DBMS *Database Management System*

DOI *Digital Object Identifier*

ECE *Early Childhood Education*

EJS *Embedded JavaScript*

FN *False Negative*

FP *False Positive*

GRU *Gated Recurrent Unit*

GUI *Graphical User Interface*

HDPMM *Hierarchical Dirichlet Process Mixture Model*

HTML *Hypertext Markup Language*

HTTP *Hypertext Transfer Protocol*

IDF *Inverse Document Frequency*

IEEE *Institute of Electrical and Electronics Engineers*

JSON *JavaScript Object Notation*

JWT *JSON Web Token*

KDD *Knowledge Discovery in Databases*

KNN *K-Nearest Neighbors*

LDA *Latent Dirichlet Allocation*

LMS *Learning Management System*

LR *Logistic Regression*

LSTM *Long Short-Term Memory*

ML *Machine Learning*

NLP *Natural Language Processing*

PCA *Principal Component Analysis*

RF *Random Forest*

RNN *Recurrent Neural Network*

SQL *Structured Query Language*

SSL *Secure Sockets Layer*

SVC *Support Vector Classifier*

SVM *Support Vector Machine*

TASD *Text-based Early Autism Spectrum Disorder Detection Dataset for Toddlers*

TF *TensorFlow*

TLS *Transport Layer Security*

TN *True Negative*

TP *True Positive*

TPR *True Positive Rate*

UI *User Interface*

Introduction

The Early Childhood Education (ECE) makes a solid base for the future of a child; research shows that children of actively involved parents have more chance for higher academic performance by 45% [18]. Such studies prove that ECE programs highly contribute to children's mental development, social interactions, and emotional well-being. Parental engagement is not a support to the homework only; rather it is broader to the school activities and creating a learning-supportive home environment [23]. According to the 2021 survey, the attitude "the more involved parents the better the child's academic performance and behavior" is held by 78% of moms and dads. Despite there being hindrances that cause limited time and money parents manage an average of 3 hours of engagement. The study conducted 5 hours per week educational activities with children [38]. Besides health conditions, which according to the CDC chronic health conditions comprise about 13% of students [11], also affect students' learning. For that reason, the schools must be ready and creative in that regard. Thus, a report by Safe Kids Worldwide calls for creating a safe learning environment. The reason for that is the fact that 40% of parents worry about their child's safety at school. While the methods like the homework monitoring and the parent-teacher meetings are still continuing to be used, there is a growing movement towards edtech, the app getting used by 63% of parents in order to support their child's learning.

The stories and role-playing methods, which constituted typical ways of teaching small kids, have been instruct beginning education for many centuries. The private balling which is another traditional method has got to a jump such that with an approximation of 54% of the parents seeking private tutoring in their kids. Educational technology comprising of digital apps and the web platforms have thus transformed the role of education. Statista evidence that world-wide childrens' education app downloads number hit 2 says the data. Projections indicate an increase from 18 billion to 33 billion by the year 2021 as shown on the Statista website

The educational game industry progressively utilizes the gamification phenomenon, with the corresponding games ABCmouse and Khan Academy Kids being the most popular. These platforms discard content-level games that comply with their educational standards. Virtual classrooms are now considered essential elements, especially when the COVID-19 pandemic arises. Media survey by Pew Research Center concluded that 71 percent of parents who oversaw their kids virtual school rated the experience successfully [12]. The digital learning platforms like Google Classroom and Schoology simplify the communication between teachers, parents, and students by facilitating a to and fro interaction [20].

Technology enhancement may have brought in the digital divide interrelation and etc. Per national statistics from the National Center for Education Statistics 14% of households with children do not have fast internet [22]. The advances in AI and ML technologies are driving innovations that personalize education [17]. In summary, the traditional methods are nowadays mutually operating with some modern technologies in the process of early childhood education change. Innovations and fair access to the resources will play a top role in achieving the best outcomes in the learnings experiences. In addition to some of the complications highlighted above, there are some health related problems that accompany early childhood education. Children can have chronic health issues that cause difficulties in the learning process, and it may affect their overall development. According to the Center for Disease Control and Prevention (CDC), about 13% of students

which are mainly diagnosed with chronic health conditions impacts the educational process [11]. It also shows the demand for service staff within education facilities. Along with physical health, mental health is another significant factor that determines how early learning develops. The AAP (American Academy of Pediatrics) underlines the need to solve any mental health problems standing in the way of children's learning abilities [40]. Creating a supportive environment that speaks to emotional health issues is a vital support for mental wellness and academic accomplishment. Autism Spectrum Disorder (ASD) is a neurodevelopmental disorder that impairs people from communicating and socially interacting in a normal manner. Recognition and providing solutions to specific issues that the students with ASD face in the educational system are indispensable [3]. The symptoms of ASD can differ significantly, starting from being very mild up to being quite severe. Symptoms include trouble with communication, repeated behaviors, sensory sensitivity, and difficulty with changing from one activity to another as shown in [3]. A recent figure from CDC indicates that as many as 1 out of 54 kids in the country have a diagnosis of autism spectrum disorder [11]. The history of ASD can be traced to the early 20th century when the original works on autistic disorder such as studies by Leo Kanner and Hans Asperger documented the characteristic behaviors of autism. Their outstanding achievements defined ASD as a separate condition that could be analyzed more accurately. ASD is mostly noticeable as symptoms develop early in children, usually before the age of three. The detection of ASD as early as possible is essential to initialize interventions and to shape therapeutic plans, resulting in prompt supports and improved outcomes for affected individuals [3].

Getting children with ASD integrate into preschools gives you no choice but to adopt special strategies. Individualized education plans (IEPs) which are aligned to different children while the support rendered by professionally trained personnel is unquestionably essential [21]. Along with parents' active participation a key factor for achieving positive result in relation to children who have ASD [24]. Parents, teachers and other stakeholders in ECE face a number of challenges, which highlights the important role of breakthrough ideas. Due to the peculiar nature of Autism Spectrum Disorder (ASD), prompt identification and suitable treatment needs to be developed. An intelligence system with AI and ML technologies can address such hindrances in detecting ASD in children at an early stage within learning environments. Built on data analytics, the system will help the educators and healthcare professionals to detect the early signs of ASD and that way, prompt interventions will be produced with positive outcomes for those children diagnosed with Autism Spectrum Disorder. Parental involvement may be the key factors in early autism spectrum disorder (ASD) detection and treatment. Parents are the vital species of the family members that notice and identify the early indicators in children with autism spectrum disorder(ASD). Early diagnosis enables prompt interventions through the network of support services and allows access to specialized support services. Parents of the children who are the ASD carriers face the size-specific tasks and they need the targeted resources and support. Establishing parent-involvement in their children's education encompassing learning and care systems for children with ASD is key to building a supportive and inclusive learning environment. Parents can be empowered if they are offered information, training, as well as access to support networks for them to cope with Autism Spectrum Disorder and the needs of their kid accordingly.

The thesis delves into the integration of Artificial Intelligence (AI) in early childhood education, with a focus on boosting parental engagement and early detection of Autism Spectrum Disorder (ASD) through the innovative ASDetectBot system. It encompasses an introduction outlining the research problem, objectives, and significance, followed by a comprehensive exploration of early childhood education and AI's pivotal role. The literature review scrutinizes existing AI research in education, particularly in early ASD detection. Detailed insights into the design and development of ASDetectBot, including its architecture, data processing, and machine learning models, are provided. The implementation section elucidates practical steps for system development, while the results and evaluation segment assesses system performance and ASD assessment efficacy. The thesis culminates with key findings, contributions, and future research avenues, emphasizing the transformative potential of AI in early childhood education and ASD detection.

Chapter 1

Basic Concepts

Introduction

The childhood education refers to a stage at which children are in their initial developmental stage that begins from birth up to the age of eight years. This chapter seeks to discuss the early childhood development in terms of physical, cognitive, social and emotional characteristics across the different stages. Under this section, the growth patterns of children have been discussed to give a general understanding of how children develop throughout certain phases of their development – infancy, early childhood, and early school going age. Each of these stages of child development is important for parents, teachers, childcare givers or whoever is involved in the upbringing of the child, this is in a bid to have an insight of proper methods to adopt that will enable these young ones to grow and develop properly.

1.1 Definition of Early Childhood Education (ECE)

Early childhood education (ECE) covers various learning programs and practices that promote the psychological, physical, intellectual, social and emotional development in children over the age of birth to eight. This part will cover the main components of ECE, its many methods and the significance that it holds on a child's lifelong learning quest.

1.1.1 Early Childhood

Early childhood is one of the most important periods of time for the development of the child when all the domains – physical, cognitive, social, and emotional undergo tremendous changes. Although the precise age category may differ between dictionaries, generatly it covers the period from birth through 8 years old. Here, we will take a close look at the stages in early childhood. The section will show the developmental milestones that can be used to distinguish the different phases.

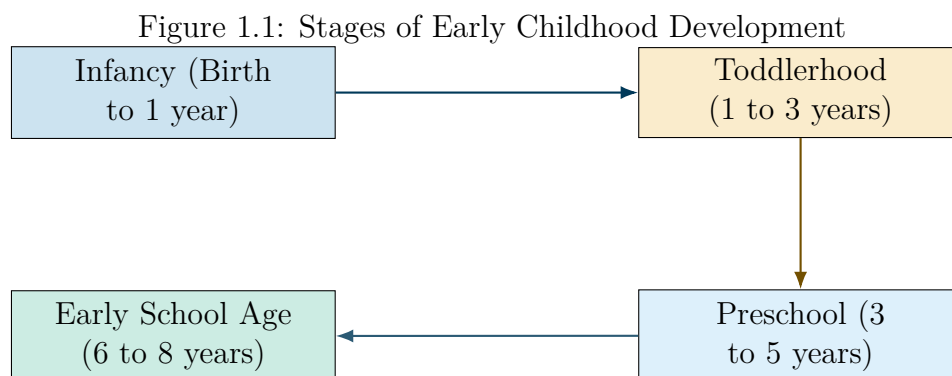
Stages of Early Childhood

The early childhood can be then subdivided into various several stages, each one of them has corresponding developmental milestones. Here's a breakdown of the commonly recognized stages:Here's a breakdown of the commonly recognized stages:

- **Infancy (Birth to 1 year):** These first year-and-a-half are marked by enormous physic growth, as well as the emergence of crucial motor skills like rolling, crawling, and grasping. Besides that, infants will begin to develop early language skills by babbling, and crying, which will, later on, become language acquisition. [32].
- **Toddlerhood (1 to 3 years):** Toddlers see a lot of changes in the same period both physically and mentally. They are progressively mobile and more and more independent while acquiring motor skills like walking and running.

Language development takes a giant leap forward as toddlers begin to use simple sentences and follow basic orders. [43].

- **Preschool (3 to 5 years):** In this period, social and emotional development of a child stands at the central board. They then begin to do plays that are more complicated, showing higher level of cooperation and negotiation. Language faculties hone down from longer sentences they form and a wide spread of emotions they express. [7].
- **Early School Age (6 to 8 years):** This stage, however, is a move from home process to formal schooling. Kids show the most important cognitive progress, being able of easier understand the complex concepts and their problem solving skills are also higher. Human relations are more developed: kids start creating friendships and learning how to handle group situations [8].

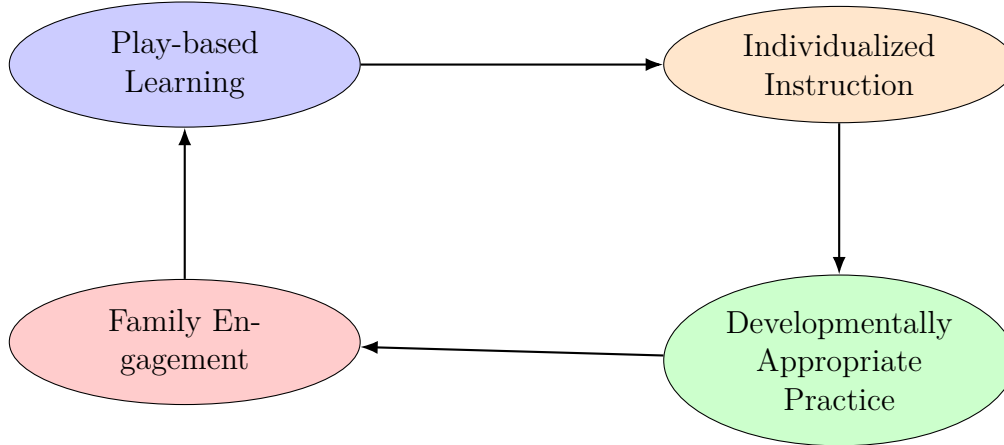


1.1.2 Core Principles of Early Childhood Education

High-quality ECE programs adhere to several fundamental principles:

- **Play-based Learning:** The play is perceived as an indispensable means of schooling in ECE. Play helps children learn as they assume different roles, interact with others and generate new ideas [41].
- **Individualized Instruction:** Efficient ECE is catered to the different requirements and ways of learning suited by every child, delivering an exceptional and rewarding journey [36].
- **Developmentally Appropriate Practice:** Every step of the way, learning resources are intentionally picking what would be most suitable to child's developmental stages, at the same time optimizing the most productive outcome [9].
- **Family Engagement:** School staff and family partnership is a cornerstone for success of the child's scholastic and psychological health [31].

Figure 1.2: Core Principles of Early Childhood Education



1.1.3 Approaches to Early Childhood Education

There are various approaches to ECE, each with its own philosophical underpinnings:

- **Montessori Method:** First, this method stresses the self-directed process on materials of open-ended nature provided in the carefully designed room.
- **Waldorf Education:** This holistic approach focuses on fostering creativity, imagination, and social-emotional development through artistic expression and play.
- **Reggio Emilia Approach:** For one thing, this approach takes the child as the core of the operation, and browsing, inquiring, and team work are its chief components with project-based learning on the peak.

1.1.4 Impact of Early Childhood Education

Studies for many years demonstrate the impact of the great quality of the ECE on the development of children. ECE programs can [31]:

- Promote social and emotional well-being and emotional regulation.
- Increase the level of school readiness and academic achievement.
- Shrink an achievement gap and contribute to the more level playing field in education.

1.2 Definition of Parental Engagement

Parental involvement or parental engagement, is referred to as the dynamic two-way communication or relationship between the educators/caregivers and parents for the proper support of learning-development of children. It goes beyond been present, and includes action and ownership among stakeholders in ensuring the

child advances in life. This section is where we will look more closely at parental engagement traits, benefits to children, and how to promote parental engagement in early childhood education centers. [27].

1.2.1 Characteristics of Parental Engagement

Effective parental engagement is characterized by several key elements:

- Parents should actively involve themselves in their child’s education not only by participating in their school activities but also by helping them with homework and moral support when needed.
- Communication and Collaboration: Open and regular communication between parents and educators is crucial for building trust and ensuring a coordinated approach to supporting the child.
- SuSupport for Learning at Home: In addition to parents taking part in activities at home which support learning at home, the environment at home becomes stimulating and the importance of studying is demonstrated through the way that parents conduct their business.
- Advocacy: Parents, themselves become parents for the well-being of their children and schooling and help them to receive proper support and resources.

1.2.2 Benefits of Parental Engagement

Parental engagement in early childhood education has been shown to have numerous benefits for children:

- Improved academic achievement : Strong parental engagement is linked to improved academic performance, stronger literacy skills, and higher test scores.
- Enhanced social-emotional development: Parental involvement fosters better social-emotional well-being, increased self-confidence, and improved emotional regulation in children.

1.3 Impact of Parental Engagement in Early Childhood Education

There has been a significant rise in the importance of parental involvement as pertaining to early childhood education (ECE) in the recent years. This factor is now considered a pivotal factor in the overall development of a child, as well as in shaping his or her academic path. Here we will link the various facets of parental involvement and its effect to children’s learning and emotional well-being, giving summary of the most recent empirical studies.

1.3.1 Cognitive and Academic Benefits

An ever-increasing number of studies show the correlation between a parent's involvement in their children's education and their cognitive development and success in school. Liao et al. 's study on home activities 2023 showed that parents who stimulated cognitive development of the family mainly for low-income families had significant effects on the school readiness and cognitive skills of the children. This highlights the importance of tailoring engagement strategies to address potential inequalities.

Moreover, a 2022 meta-analysis by Castro et al. revealed that parAdditionally, in a 2022 meta-analysis by Castro et al. proved that parental involvement in home-based literacy activities was associated with language development or early reading skills were better in toddlers. This emphasizes the importance of fostering a love for reading and creating a print-rich environment at home.

1.3.2 Social-Emotional Development and Well-being

Another way in which the parental involvement affects the social - emotional development and mental health of children is also worth noting. A study that came out in 2023 titled Kaščák et al. looked at parental participation in ECE and how it influences youngsters' emotions. The research results demonstrated that good parent-teacher relationships led to the development of emotional competence among toddlers [34]. This implies that the cooperation of educators and families will surely give children what they need to meaningfully handle their emotions.

In addition, Park et al. (2022) in their research made a link between parental engagement in the upbringing of the child and their social skills development. The investigators found that the involvement of parents in school extracurriculars and their being willing to communicate with teachers had positive implications on children's social interactions and peer relationships according to their research [39].

1.3.3 Long-Term Benefits and School Readiness

Parental engagement is not only beneficial for the early years but is also a positive influence on a child's later years. Researchers believe that the parent-school relationship which is established during the early childhood education period impacts children's likelihood of success in future academic life. Alper et al. (2021) study showed that parents who are involved with their preschooler students in the long-term have more positive effects on the achievements of their students. As a result of their study, they found out that the performance of children in the elementary school was shaped by high level of parental involvement in their ECE [2].

1.4 Application of Artificial Intelligence (AI) in Early Childhood Education

Defining Artificial Intelligence

Artificial Intelligence (AI) encompasses a broad spectrum of technologies, but its core principle lies in enabling machines to exhibit intelligent behavior [44]. This includes capabilities such as:

- **Learning:** AI systems can learn from data, identify patterns, and adapt their behavior accordingly.
- **Problem-solving:** AI algorithms can analyze complex situations, propose solutions, and make informed decisions.
- **Reasoning:** AI systems can process information, draw logical inferences, and justify their conclusions.

Artificial Intelligence in Early Childhood Education - A Comprehensive Handbook

The introduction of AI in early childhood education raises several questions that should be keenly considered by educators and learners at large. Here, we explore the profound implications and possibilities of AI in ECE whereas we discussed its benefits and problems.

Benefits of Artificial Intelligence in Early Childhood Education

AI technologies offer numerous benefits for ECE:

- **Enhanced Learning:** Children exposed to AI learning platforms show significant improvements in their understanding of AI, machine learning, and robotics concepts.
- **Development of Broader Skills:** AI learning impacts skills like creativity, emotional control, literacy, and computational thinking.
- **Increased Engagement:** AI-driven teaching platforms effectively boost children's enthusiasm for learning, creating dynamic and interactive educational experiences.

Challenges and Opportunities

While AI presents vast opportunities, it also comes with challenges:

- **Teacher Training:** Many educators face barriers in using AI due to limited training and familiarity with the technology.

- **Curriculum Development:** Creating age-appropriate curriculums that leverage AI's strengths is crucial for effective integration.
- **Ethical Considerations:** Ensuring data privacy and fair usage of AI tools remains a paramount concern in ECE.
- **Technology Accessibility:** Bridging the gap in access to AI tools among educators is essential for equitable learning opportunities.

Personalized Learning with Artificial Intelligence

AI facilitates personalized learning experiences tailored to each child's unique needs and learning style:

- **Adaptive Learning:** AI analyzes individual performance, identifies strengths and weaknesses, and adjusts learning conditions to optimize engagement.
- **Early Intervention:** AI detects potential learning difficulties early on, enabling timely support and intervention strategies.
- **Customized Feedback:** AI provides personalized feedback, highlighting areas of excellence and suggesting actionable improvements.

Empowering Educators with Artificial Intelligence Tools

AI tools empower educators with innovative solutions:

- **Learning Management Systems (LMS):** AI-based software creates customize lessons plans that monitor the progress, and furnishes insights derived from big data.
- **Language Learning Apps:** AI-based apps make personalised learning fun and interactive through activities available that help learners acquire and use language.
- **Assessment Tools:** AI assessments bar the path for cumbersome assessment processes by serving accurate metrics, insights, and actionable recommendations.
- **Teaching Assistants:** AI teaching assistants have an essential role in supporting classroom activities, providing real-time educational content, and optimizing communication.

Data Mining: Unveiling Insights for Early Childhood Education

Data mining, that is a subdomain under knowledge discovery in databases (KDD) is all about deriving hidden patterns and information from big volumes of data. In early childhood learning perspective, data mining provides a powerful

instrument that helps educators understand how children’s learning journeys, detect problems, and work on creating educational practice that improves the process. The data mining basics, techniques, and its within the ECE principle will be discussed in this section, dependent upon the recent research.

1.4.1 Techniques for Data Mining Exploration

Data mining utilizes a range of techniques to extract valuable knowledge from datasets:

- **Classification:** Such a method segments data into predefined groups that match with the properties of points. It has the ability to describe the trend of a child who is prone to developmental disabilities or accommodates learning styles.
- **Clustering:** This graphical method draws classes that contain similar data without prior defined categories. It could turn out to identify learners’ sub-groups having similar emotional status or learning styles.
- **Association Rule Learning:** Through this method it is possible to find web connections between the entities on the data elements. That can be a very useful tool, which aims to find those points that simultaneously contribute to children’s motivation or academic success..

1.4.2 Recent Advancements and Considerations

The field of data mining is not static as emergence of machine learning and artificial intelligence leads to new developments that enable the mining of even more complex data. On the other hand, ethics issues related to data security, privacy, and system biases demand sophisticated consideration. Making responsible data collection, storing, and analyzes are needed for using data mining to provide positive impacts for children and educators within the ECE setting.

1.5 Autism Spectrum Disorder: Understanding Diverse Learning Needs in Early Childhood Education

Autism spectrum disorder (ASD) is a complex neurodevelopmental condition that affects how an individual processes information and interprets the world. The core features of autism are persistent deficits in social interaction and communication, as well as restricted, repetitive or stereotyped patterns of behavior, interests or activities. Each individual with ASD displays a unique combination of characteristics, ranging from mild to severe, requiring individually determined educational and treatment programming.

1.5.1 Understanding Autism Spectrum Disorder

ASD comprises a range of peculiarities, and however people can display drastic differences in social interaction, repetitive behaviors and special interests. Here are some key characteristics of ASD:

- **Social Communication Difficulties:** Individuals with ASD may experience challenges with nonverbal communication, social interaction, and understanding social cues.
- **Restricted Interests and Repetitive Behaviors:** Children on the spectrum may have intense interests in specific topics and engage in repetitive behaviors or routines.
- **Sensory Sensitivities:** Some individuals with ASD may exhibit heightened sensitivities to sensory stimuli such as sound, touch, or light.

1.5.2 Importance of Early Identification and Intervention

Diagnosis and intervention in the early stages of ASD play a great role in enabling effective adjustment to the situation. Teachers are indeed an active party of the team in locating autistic symptoms in children during their development.

- **Promotes Brain Development:** Early intervention programs can capitalize on the brain's plasticity during early childhood, fostering the development of communication, social, and learning skills [19].
- **Improves Quality of Life:** Early intervention can equip children with ASD with the tools and strategies they need to navigate social interactions, manage sensory sensitivities, and thrive in educational settings.

Conclusion

In conclusion, early childhood development is a complex process where several vital developmental domains within the child are experienced. This first five stages of child development, from infancy to early school going age, is a useful guide towards understanding children's growth. This knowledge is imperative in developing appropriate child care settings that help grow healthy children and equip them with the ability to face future academic and other challenges. When one understands the importance and complication of early childhood, it is now possible then reverse the destinies of children.

Chapter 2

State Of The Art

2.1 Introduction

In the last years, there has been an increase in the researchers interest about how to integrate artificial intelligence into education's levels that are from primary school to higher education. That focus has mainly been on AI's potential applications in traditional areas of education, for instance, intelligent tutoring systems and collaborate learning applications. On the contrary, there is an emerging field of study that focuses on the implication of AI in early childhood learning which is pre-school. Building on these technologies, AI in education strives to create great learning environment and steps to AI literacy for children. Teachers may make use of AI in ECE, to some extent, and this might allow them to tailor learning, to determine learning gaps early and to come up with targeted interventions, in order to support every child's individual needs in learning. A significant obstacle that we need to take into account during the application of AI in ECE is making sure that the AI-technologies and devices are developmental and in line with the core principles of early childhood development. Furthermore, problems such as data privacy, security and ethical aspects should be approached cautiously when adopting AI technologies into ECE facilities. On the other hand, there are a plethora of exceptional openings that AI has to offer the field of early childhood education. AI help educational personnel in processing large volumes of data to detect trends of learner's patterns, preferences as well as their strong and weak areas. Furthermore, AI tools can help to realize interactive and engaging learning, develop creativity and critical thinking skills, and promote collaboration among young people through application of AI powered tools. The positive impact of AI in ECE can be seen beyond the classroom as it can enhance parents and caregivers by equipping them with helpful information about their child's progress in learning and provide customized feedback to them in assisting their young one's learning at home.

This chapter presents a systematic literature review on Artificial Intelligence in Education (AIED) to explore the primary categories of AI applications, research topics, and major research design elements.

2.2 Related Works

Artificial Intelligence (AI), a significant subfield of computer science, focuses on the creation of intelligent systems that mimic human cognitive processes such as learning, reasoning, and problem-solving. These systems include technologies like expert systems and neural networks[26].

Prior review studies in AI in Education (AIED) have targeted various application areas such as chatbots, precision education, mathematics education, and student assessment, utilizing methodologies like bibliometric, systematic, or narrative reviews. Theoretical frameworks often referenced in AIED research include constructivist learning theory, learning style theory, cognitive theories of learning, and item response theory. This research employs bibliometric and content analysis to explore elements like AIED applications, research objectives, methods, guiding theories, and research contexts, contributing to a comprehensive understanding of the AIED landscape. Recent major reviews have addressed the effectiveness of intelligent tutoring systems, the classification of AIED applications, and learning

analytics methods [13].

The field of education has seen significant transformation with the integration of AI across various educational levels, from primary education to higher learning institutions. This shift has prompted extensive research into the potential of AI to revolutionize teaching and learning paradigms. Researchers are actively exploring how AI applications can reshape the educational landscape, aiding educators in administrative tasks and enhancing students' learning experiences through tools like intelligent tutoring systems and robotic kits [14]. The emergence of "AI literacy" as a crucial competency reflects the need for effective engagement with AI technologies across different contexts, particularly in early childhood education. This early introduction is aimed at equipping young learners with the basic competencies needed in today's digital era and fostering curiosity and problem-solving skills through well-designed AI educational tools [10].

In specialized areas such as understanding social dynamics within communities, advanced AI techniques like Latent Dirichlet Allocation (LDA) and Hierarchical Dirichlet Process Mixture Model (HDPMM) are employed [45]. Content analysis of online platforms offers insights into the social support dynamics, sentiment analysis, and experiences of individuals affected by Autism Spectrum Disorder (ASD) [16]. Additionally, numerous studies have investigated significant factors and techniques for ASD detection and analysis, utilizing feature selection techniques and classifiers such as Support Vector Machine (SVM), Random Forest (RF), and Logistic Regression (LR) to detect ASD in various age groups [5].

Many studies have focused on investigating, classifying, and exploring significant factors of ASD. For instance, researchers have used several feature selection techniques and classifiers to detect ASD in toddlers, children, adolescents, and adults, finding that SVM performed the best, achieving high accuracies [5].

Researchers have also utilized various machine learning approaches for ASD investigation and detection. For example, samples collected from persons aged 16–30 years were analyzed using tree-based classifiers to investigate ASD, extracting several rules for distinguishing between normal and autism conditions. Different datasets have been analyzed using methods like KNN, SVM, and RF, with RF demonstrating the best performance in identifying ASD.

Lastly, the identification of autism subtypes and discriminatory factors among different groups has been achieved using clustering algorithms like k-means. These efforts have significantly contributed to the advancement of ASD detection and analysis using AI and machine learning techniques, highlighting the diverse applications of AI in addressing educational and developmental challenges [1].

Table 2.1: Comparison of AI Studies in Education and Social Dynamics

Study	Focus Area	Methods Used	Key Findings/Applications
Gao et al. (2023) [26], Giattino et al. (2023) [28]	AI systems, Expert systems, Neural networks	Bibliometric analysis, Content analysis	Understanding AI applications in cognitive processes
Gao et al. (2019) [25], Chaudhry et al. (2021) [13]	AI in Education (AIED)	Systematic review, Narrative review	Effectiveness of intelligent tutoring systems, Classification of AIED applications
Chen et al. (2020) [14], Su et al. (2022) [42]	AI in various educational levels	Review	Impact of AI on teaching, AI literacy in early education
Burgsteiner et al. (2016) [10], Kandlhofer et al. (2016) [33], Long et al. (2020) [35]	AI literacy, Early childhood education	Case studies	Development of AI educational tools for young learners
Teh et al. (2006) [45]	Social dynamics, ASD	Latent Dirichlet Allocation (LDA), Hierarchical Dirichlet Process Mixture Model (HDPMM)	Social support dynamics, Sentiment analysis for ASD

Research conducted on the viability of employing the T ASD dataset for the early screening of ASD

The proposed study [37] is to leverage the BERT model to create a classification model to detect early signs of ASD with the participants being children below 2 years of age. Considering this purpose, the T ASD-Dataset for Toddlers which we published was employed by these researchers.

Dataset: The T ASD-Dataset V1 [29] presents texts described as script sequences referring to the utilization of interventions, interactions, and activities by toddlers with ASD and a control group. The elements characteristic of ASD assessment are presented, therefore you will notice Attention Response, Change Reaction, Word Repetition, Eye Contact, Emotional Empathy, Finger Movements, Focused Attention, Follow Pointing, Repetitive Behavior, and Toy Arranging as the key components. The specific record set is made up of 297 entries, and 163 of them are subjected to the ASD group while the rest, 134, are placed in the non-ASD category.

Model: The behavioral traits are first annotated with the help of the BERT model, a deep learning architecture used for such NLP problems. Cross-validation is not employed herein to prevent overfitting; instead, the given model goes for

positive training and testing with an 80% train and 20% test split.

Results: Specifically, in the experiment performed on the TAsD-Dataset, BERT achieved an 88% accuracy rate for identifying behavioral traits associated with the condition, demonstrating its aptitude for early diagnosis of ASD. This performance is better than previous research results in the identification of ASD, as the present work has developed a novel model with better robustness and stability to resolve real-time issues of the global mean of the behavioral traits in the dataset.

The presented work contributes to the argument on how the effectiveness of an AI system can be used to detect behavioral patterns in ASD individuals. The information provided in the TAsD-Dataset enables researchers to capture relevant behavioral signs likely to appear at an early stage of ASD. Detailed observations from parents offer a rich source of information that, when combined, form the best model. More research should focus on the fusion of other datasets and explore additional modalities in the fusion process to achieve even more accurate results.

Table 2.2: Comparison of AI Studies in ASD Management and Detection

Study	Focus Area	Methods Used	Key Findings/Applications
Pal et al. (2024) [37]	AI for ASD management	Innovative approaches	Development of solutions for ASD challenges
Parsloe et al. (2015) [16], Bellon et al. (2020) [6]	Online platforms, ASD	Content analysis	Insights into experiences of individuals with ASD
Bala et al. (2022) [5], Akter et al. (Various years) [1]	ASD detection	Feature selection, Classifiers (SVM, RF, LR)	High accuracy in detecting ASD
Chowdhury et al. (2020) [15]	ASD investigation	Tree-based classifiers, KNN, SVM, RF	Rule extraction, RF performance in identifying ASD
Akter et al. (2021) [1]	ASD subtypes	Clustering algorithms (k-means)	Identification of autism subtypes

2.3 Conclusion

The introduction of AI in ECE will entail a lot of work but the end result will be very positive and pupils will be able to learn how to use AI right from when they are at an early age. This transformatory course of the mentioned AI applications in this chapter is revealed by their variety of intelligent tutoring systems that track down individual strengths and weaknesses and powerful sentiment analysis softwares that can be applied in autism spectrum disorder (ASD) intervention and development of behavioral outlines specifically for ASD. The multifaceted possibileness of artificial intelligence provides a glimpse into the future, where the educational and technological strands are that tightly tangled that every child's learning trajectory enriched by the technology. Firstly, sustained focus on the Artificial Intelligence in the early childhood milieu will be a cornerstone of developing this field.

This means the need to build AI powered systems which are not only factual but also designed in a way that are appropriate for young learners depending on the current development needs. Moreover, wise educational integration is the key factor that will guarantee the creation of the interface between the AI and human beings, not between the AI and machines. Effective collaboration between teachers who have education background and technology field experts, with the highest role of policy-makers, will definitely be important to harness the maximum power of artificial intelligence to create abilities and skills in young minds propagating in the digital world. Moreover, education of AI in class enable go beyond the traditional classroom. Immersion is centered on the formation of environments in which AI interventions are all-inclusive creating a learning space that supports diverse learning methods and the learners with special educational needs. One the hand, integrating artificial intelligence to confront problems such as early detection and intervention of developmental disorders such as autism spectrum disorder also demonstrate how it has the capability to be an ultimate partner in fostering inclusive education as a well as in offering the best in child development.

Chapter 3

Methodology and System Design

Introduction

This chapter provides information with regard to the actions taken for the project and details concerning the ASDetectBot system. Automated Screening Detection and Analysis using Chatbot – an experimental system with parents and healthcare practitioners suggesting enhanced likelihood of identifying ASD. Thus, ASDetectBot, which applies AI and data analysis methods, suggests a solution fit for both early developmental milestones and parental anxiety.

This chapter aims to discuss the generalized methods of the system architecture, the graphical user interface design, database system design, and deep learning model for ASD detection on the proposed system. Diagnostic generalization of ASD, especially in the toddlers, prove difficult due to the asymptomatic form of the symptoms and difficult nature of behavioral assessment in young children. ASDetectBot accomplishes enhanced data analysis and analysis mechanisms for the early indicator of ASD, which is helpful for the doctors and parents.

3.1 ASDetectBot Overview

The “ASDetectBot”, intelligent system that can assist parents in tracking the physical and cognitive progression of the child, especially for the purpose of identifying when the child may have symptoms of a condition like Autism Spectrum Disorder (ASD). This system is based on the usage of the advanced methods of artificial intelligence to optimize the processes of early childcare and parental involvement in the process, the application is designed to be a fully usable tool for aiding parents in the early diagnostic of ASD signs in their children. Connecting in harmony with one another, the elements of ASDetectBot’s system include a graphical shell, server, database, and, most importantly, machine learning algorithms. The primary operations occurring in the system include text mining and deep learning analysis to analyze the data and ascertain characteristics associated with ASD.

Objectives and Functionalities

System Objectives

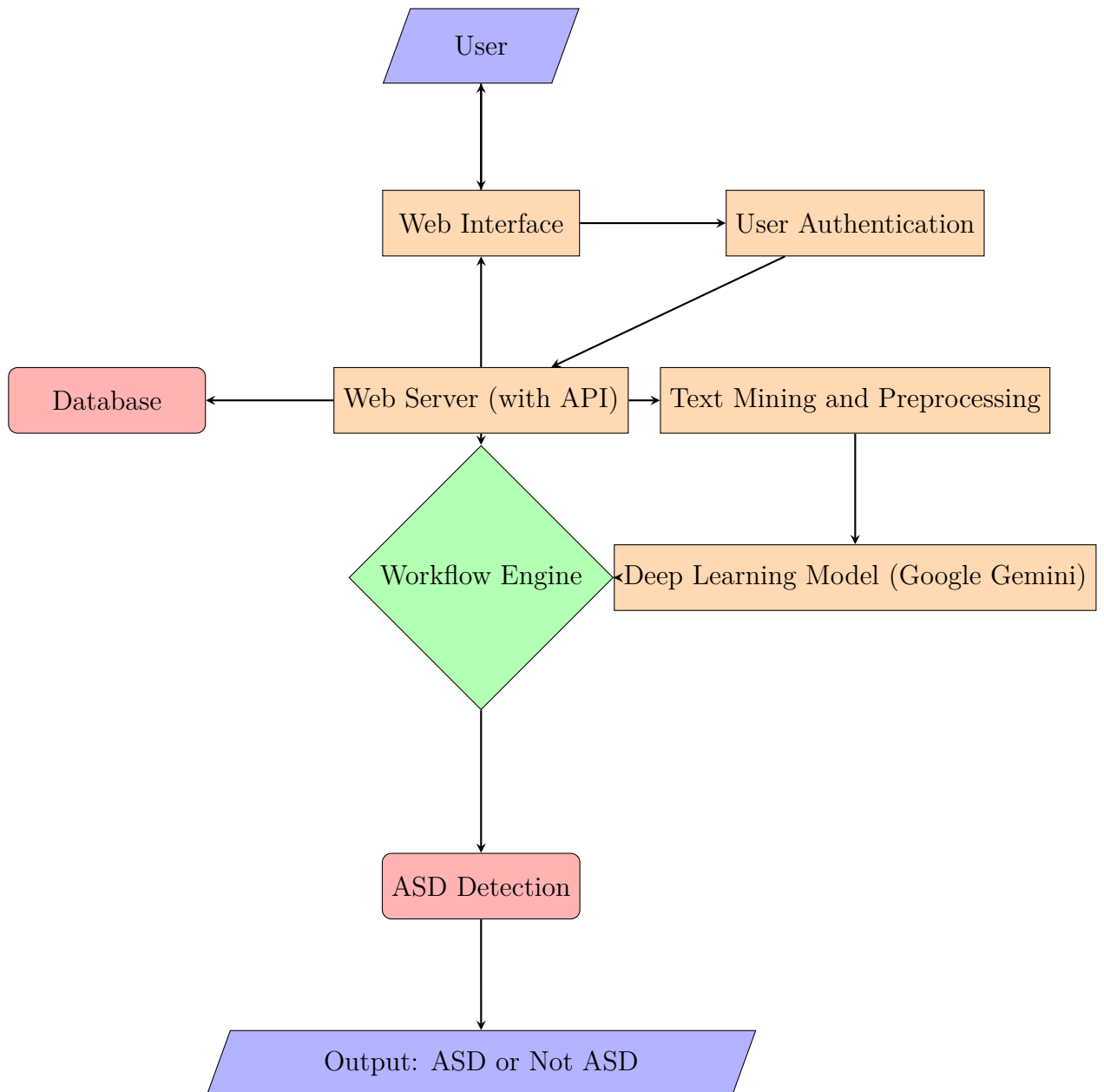
The objectives of the ASDetectBot are aimed solely and directly at increasing early identification, parental involvement, and parental enablement. The ASDetectBot seeks to detect early indications of ASD using Chatbot-based techniques in the process. The system also enables parents to have a direct play in monitoring and assessment process, giving them the information and tools with which to effectively aid their child. ASDetectBot thus helps parents to make the proper decisions on how to take care of the child or where to seek assistance from a specialist.

Key Functionalities

ASDetectBot offers a range of functionalities designed to support parents and healthcare professionals in the early detection of ASD. It ensures secure access to the system through user authentication, allowing parents to input detailed information about their child's developmental milestones, behavioral observations, and responses to specific questionnaires. The system utilizes deep learning algorithms to analyze the input data and provide an initial assessment of the child's risk for ASD. Based on the assessment results, ASDetectBot provides personalized feedback and recommendations, guiding parents on the next steps. It also enables ongoing monitoring of the child's development over time, allowing parents to track improvements or emerging concerns. Furthermore, the system creates detailed reports that can be shared with healthcare professionals, facilitating informed discussions and decision-making. Each functionality is designed to enhance the early detection of ASD by providing parents with the tools and information they need to take proactive steps in their child's development.

3.2 System Architecture Overview

Figure 3.1: ASDetectBot App Architecture



3.2.1 User Interface (GUI)

The user interface (GUI) of the Early ASDetect system is meticulously crafted to ensure usability and accessibility across a wide range of devices, including desktops, tablets, and smartphones. Utilizing modern front-end frameworks "React", the GUI supports dynamic, interactive elements that enhance the user experience. The design is both intuitive and responsive, adapting seamlessly to different screen sizes and orientations. Secure authentication mechanisms, including OAuth and JSON Web Tokens (JWT), are implemented to manage user sessions and protect

sensitive information. This ensures that user data remains secure and that only authorized individuals can access specific functionalities of the system.

3.2.2 Backend Server

The backend server is the core component of the Early ASDetect system, responsible for handling client requests, processing application logic, and managing data flow. Developed using Node.js, the server exposes a RESTful API that acts as the interface between the frontend and the server. This API handles all client requests, processes data, and returns appropriate responses. Secure token-based authentication is employed to manage user sessions, ensuring secure and efficient tracking of user interactions. The backend server is designed for scalability and performance, capable of handling multiple concurrent requests with minimal latency.

3.2.3 Database

The database management system (DBMS) underpinning the Early ASDetect system is MySQL, chosen for its robustness and reliability. The database schema is meticulously designed to include well-defined tables for various entities such as Users, Children, Conversations, and Messages. This structured approach not only ensures data integrity but also facilitates efficient querying and data management. Relationships between tables are carefully modeled to support complex queries and data retrieval operations. Regular backups and data redundancy mechanisms are in place to protect against data loss and ensure high availability.

3.2.4 Text Mining and Preprocessing

Text mining and preprocessing are pivotal steps in preparing input data for analysis within the Early ASDetect system. Advanced data cleaning scripts are utilized to normalize the data, removing noise and irrelevant information to enhance the quality of the input. Techniques such as tokenization, lemmatization, and stop-word removal are employed to convert raw text into structured features suitable for machine learning models. These preprocessing steps are crucial for transforming unstructured text data into a format that can be effectively analyzed to detect patterns indicative of ASD.

3.2.5 Deep Learning Model for ASD Detection

At the heart of the Early ASDetect system is a sophisticated deep learning model designed to analyze sequential conversation data for signs of ASD. Long Short-Term Memory (LSTM) networks are employed due to their ability to capture long-term dependencies and patterns within sequences. The model is trained on an extensive dataset of labeled ASD-related conversations, enabling it to learn subtle cues and indicators of ASD. Real-time integration of the model into the system allows for immediate analysis and predictions during user interactions, providing timely and accurate assessments.

3.2.6 Coordination Model

The coordination model orchestrates the interaction sequence between users and the chatbot, ensuring a coherent and logical flow of conversation. A workflow engine underpins this model, dynamically adapting the conversation path based on user inputs and model predictions. Decision points within the workflow allow for real-time adjustments, providing a personalized and context-aware user experience. This adaptive approach ensures that each user interaction is meaningful and tailored to their specific needs and responses.

3.2.7 Web Access

The web server, hosting the Early ASDetect application, plays a crucial role in managing HTTP requests from clients. Security is a paramount concern, with SSL/TLS encryption implemented to protect data communication between the client and server. This encryption ensures that all transmitted data is secure, preventing unauthorized access and safeguarding user information. The web server is optimized for performance and reliability, ensuring that users experience fast and responsive interactions with the system.

3.2.8 System Integration and Middleware

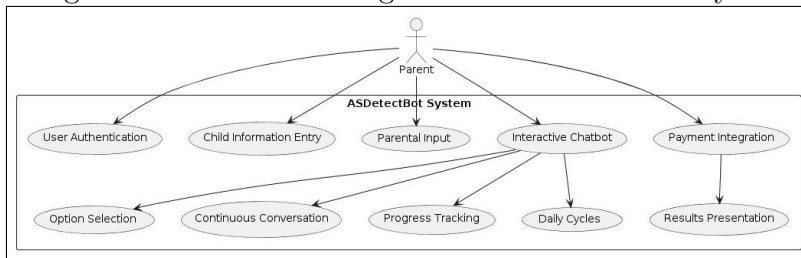
System integration within the Early ASDetect system is facilitated by a suite of middleware components designed to handle various tasks such as error logging, request processing, and communication between different modules. An API gateway is employed to route requests, enforce rate limits, and manage API versioning. This middleware infrastructure ensures that the system remains scalable, maintainable, and resilient. By providing seamless integration of all components, the middleware supports the overall functionality and reliability of the Early ASDetect system, enabling it to deliver a robust and efficient user experience.

3.3 System Diagrams

This section includes various system diagrams that provide visual representations of the ASDetectBot's architecture, including sequence diagrams, use case diagrams, and activity diagrams. Each diagram is accompanied by a detailed description to aid in understanding the system's design and functionality.

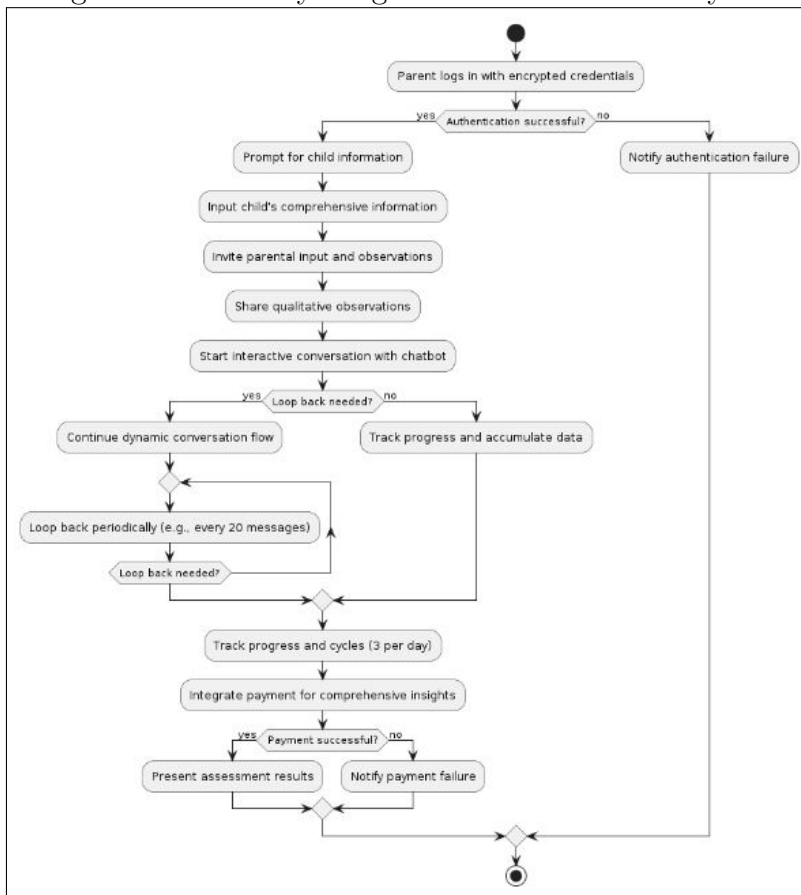
3.3.1 Use Case Diagram

Figure 3.2: Use Case Diagram for ASDetectBot System



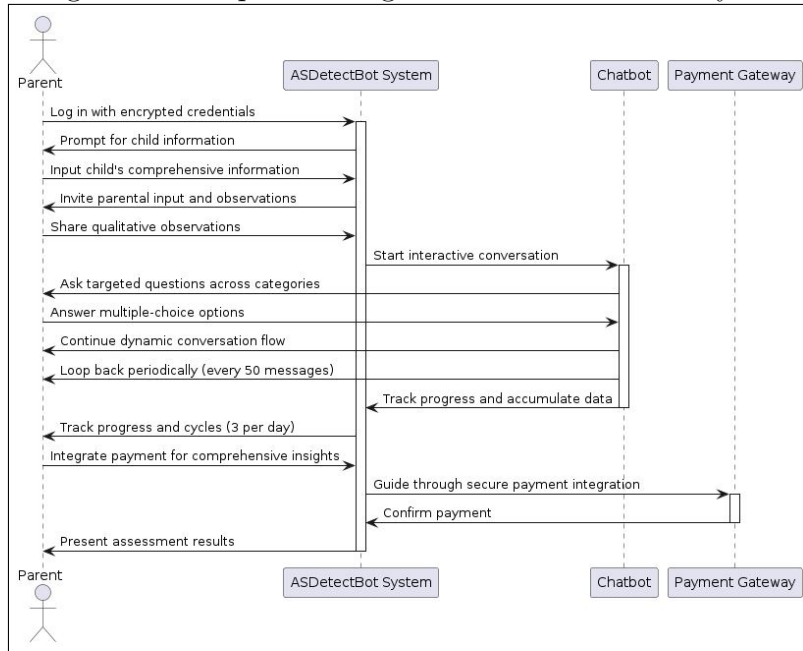
3.3.2 Activity Diagram

Figure 3.3: Activity Diagram for ASDetectBot System



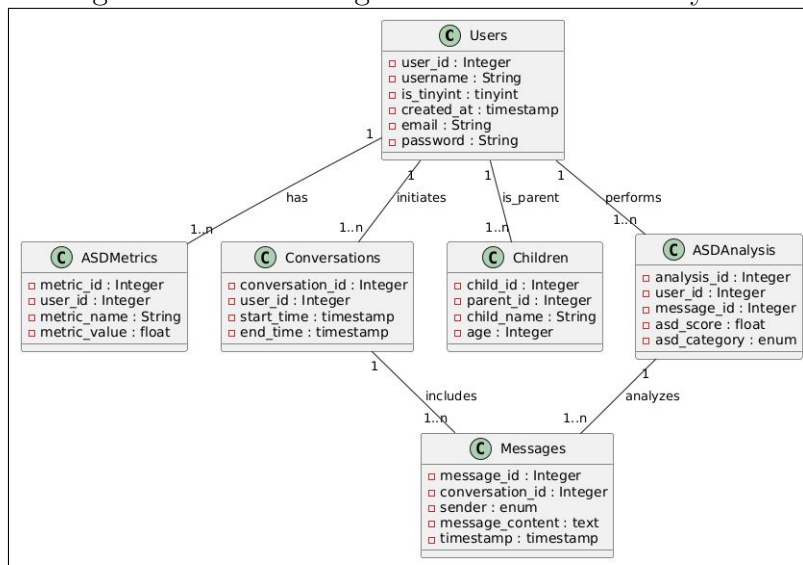
3.3.3 Sequence Diagram

Figure 3.4: Sequence Diagram for ASDetectBot System



3.3.4 Class Diagram Description

Figure 3.5: Class Diagram for ASDetectBot System



Relationships

- **Users** has a one-to-many relationship with **ASDMetrics**
- **Users** has a one-to-many relationship with **Conversations**
- **Users** has a one-to-many relationship with **Children**

- **Users** has a one-to-many relationship with **ASDAnalysis**
- **Conversations** has a one-to-many relationship with **Messages**
- **ASDAnalysis** has a one-to-many relationship with **Messages**

3.4 ASDetectBot Database Design

The ASDetect Database is designed to manage information related to Autism Spectrum Disorder (ASD), including data about users, children, conversations, analysis metrics, and more. It serves as a platform for tracking interactions, assessments, and user profiles within the context of ASD detection and support.

3.4.1 Description of the ASD Analysis Table

The `asd_analysis` table is designed to store analysis results related to Autism Spectrum Disorder (ASD). Below is the SQL code to create the `asd_analysis` table:

```

1 CREATE TABLE `asd_analysis` (
2   `analysis_id` int(11) NOT NULL,
3   `user_id` int(11) DEFAULT NULL,
4   `message_id` int(11) DEFAULT NULL,
5   `asd_score` float DEFAULT NULL,
6   `asd_category` enum('low','medium','high') DEFAULT 'low'
7 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```

Listing 3.1: SQL code to create the asd analysis table

The table includes the following columns:

- **analysis_id**: An integer identifier for each analysis result (primary key).
- **user_id**: An integer identifier for the user associated with the analysis result (foreign key to another table).
- **message_id**: An integer identifier for the message related to the analysis result (foreign key to another table).
- **asd_score**: A floating-point number representing the ASD score.
- **asd_category**: An enumeration indicating the ASD category (low, medium, high) with a default value of 'low'.

3.4.2 Description of the ASD Metrics Table

The `asd_metrics` table is designed to store various metrics related to Autism Spectrum Disorder (ASD). Below is the SQL code to create the `asd_metrics` table:

```

1 CREATE TABLE `asd_metrics` (
2   `metric_id` int(11) NOT NULL,
3   `user_id` int(11) DEFAULT NULL,
4   `metric_name` varchar(255) NOT NULL,
5   `metric_value` float DEFAULT NULL
6 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```

Listing 3.2: SQL code to create the asd metrics table

The table includes the following columns:

- **metric_id**: An integer identifier for each metric (primary key).
- **user_id**: An integer identifier for the user associated with the metric (foreign key to another table).
- **metric_name**: A varchar field for the name of the metric.
- **metric_value**: A floating-point number representing the value of the metric.

3.4.3 Description of the Children Table

The `children` table is designed to store information about children. Below is the SQL code to create the `children` table:

```
1 CREATE TABLE `children` (  
2   `child_id` int(11) NOT NULL,  
3   `parent_id` int(11) DEFAULT NULL,  
4   `child_name` varchar(255) NOT NULL,  
5   `age` int(11) DEFAULT NULL  
6 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Listing 3.3: SQL code to create the children table

The table includes the following columns:

- **child_id**: An integer identifier for each child (primary key).
- **parent_id**: An integer identifier for the parent of the child (foreign key to another table).
- **child_name**: A varchar field for the name of the child.
- **age**: An integer representing the age of the child.

3.4.4 Description of the Conversations Table

The `conversations` table is designed to store information about conversations. Below is the SQL code to create the `conversations` table:

```
1 CREATE TABLE `conversations` (  
2   `conversation_id` int(11) NOT NULL,  
3   `user_id` int(11) DEFAULT NULL,  
4   `start_time` timestamp NOT NULL DEFAULT current_timestamp(),  
5   `end_time` timestamp NULL DEFAULT NULL,  
6   `favorite` tinyint(1) DEFAULT NULL,  
7   `child_id` int(11) DEFAULT NULL  
8 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Listing 3.4: SQL code to create the conversations table

The table includes the following columns:

- **conversation_id**: An integer identifier for each conversation (primary key).

- **user_id**: An integer identifier for the user associated with the conversation (foreign key to another table).
- **start_time**: A timestamp indicating the start time of the conversation (defaults to the current timestamp).
- **end_time**: A timestamp indicating the end time of the conversation (nullable).
- **favorite**: A tinyint field (0 or 1) indicating if the conversation is marked as a favorite.
- **child_id**: An integer identifier for the child associated with the conversation (foreign key to another table).

3.4.5 Description of the Messages Table

The `messages` table is designed to store communication messages within a system. Below is the SQL code to create the `messages` table:

```

1 CREATE TABLE `messages` (
2   `message_id` int(11) NOT NULL,
3   `conversation_id` int(11) DEFAULT NULL,
4   `sender` enum('user','bot') NOT NULL,
5   `message_content` text DEFAULT NULL,
6   `timestamp` timestamp NOT NULL DEFAULT current_timestamp()
7 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```

Listing 3.5: SQL code to create the messages table

The table includes the following columns:

- **message_id**: An integer identifier for each message (primary key).
- **conversation_id**: An integer identifier for the conversation to which the message belongs (foreign key to another table).
- **sender**: An enumeration indicating whether the message was sent by a user or a bot.
- **message_content**: Text content of the message.
- **timestamp**: Timestamp indicating when the message was sent (defaults to the current timestamp).

3.4.6 Description of the Users Table

The `users` table is designed to store information about users. Below is the SQL code to create the `users` table:

```

1 CREATE TABLE `users` (
2   `user_id` int(11) NOT NULL,
3   `username` varchar(255) NOT NULL,
4   `is_parent` tinyint(1) DEFAULT 0,
5   `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
6   `email` varchar(255) NOT NULL,
7   `password` varchar(255) NOT NULL

```

```
8 | ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

Listing 3.6: SQL code to create the users table

The table includes the following columns:

- **user_id**: An integer identifier for each user (primary key).
- **username**: A varchar field for the username.
- **is_parent**: A tinyint field (0 or 1) indicating if the user is a parent (defaults to 0).
- **created_at**: A timestamp indicating the creation time of the user (defaults to the current timestamp).
- **email**: A varchar field for the email address of the user.
- **password**: A varchar field for the password of the user.

Conclusion

In summary, this chapter has described the strong evidence of how the ASDetectBot system works in this study and its importance in early autism detection as well as its effective interaction with parents and doctors. Thus, through the implementation of an organized ASDetect database, a clear and coherent user interface, and a high-capacity deep learning model, ASDetectBot aspires to minimize the detection period and promptly assist families that have been impacted by the disorder. Every aspect of the system's features is developed to respond to the ASD detection challenges; it reflects the system's architecture and various functions aimed at being capable of enhancing early intervention. In this chapter, the reader is provided with backgrounds about how exactly the ASDetectBot works, in addition to its possibility in effecting early childhood development and parental involvement.

Chapter 4

Implementation

Introduction

This chapter provides a detailed description of the T ASD-Dataset for early toddler’s ASD detection as well as important assessment features associated with toddler behaviors. It includes an account of the generation of the dataset and how the data was acquired, processed, and transformed for machine learning. It includes 326 texts from 14 classes, all captured in early childhood to provide a comprehensive depiction of the ASD markers. By preprocessing the T ASD-Dataset, it becomes well-prepared for further analysis and model training, ensuring the effectiveness of machine learning algorithms in capturing distinguishing characteristics and facilitating accurate classification. The selection of deep learning models, including Artificial Neural Network (ANN), Convolutional Neural Networks (CNN), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU), offers a comprehensive exploration of various architectures’ effectiveness in capturing the nuances of ASD-related behaviors present in the T ASD dataset. This comparative analysis aims to provide insights into the performance and suitability of each model for early ASD detection, guiding future research and clinical applications in this critical domain. This study follows a structured approach to address the topic of autism detection. The application of artificial intelligence by the help of the chatbot can be extremely helpful in collecting a wide variety of information regarding the specific behaviors and communication development of toddlers with ASD.

4.1 Data Description and Embedding Methods

4.1.1 Dataset Creation and Description

The T ASD-Dataset is designed to facilitate early ASD detection in toddlers. It includes text sequences depicting scenarios of toddlers with and without ASD, incorporating essential ASD assessment features linked to distinct toddler behaviors. With 326 sentences distributed across 14 classes, it offers a comprehensive depiction of ASD markers in early childhood, significantly contributing to detection (see Table 4.1).

Table 4.1: Count of ASD Labels

ASD Label	Count
Attention Response	29
Change Reaction	48
Word Repetition	28
Repetitive Behavior	17
Follow Pointing	28
Focused Attention	30
Finger Movements	48
Eye Contact	18
Emotional Empathy	36
Toy Arranging	15
Noise Sensitivity	25
Sign Communication	12
Tiptoe Flapping	8
Sharing Interest	15

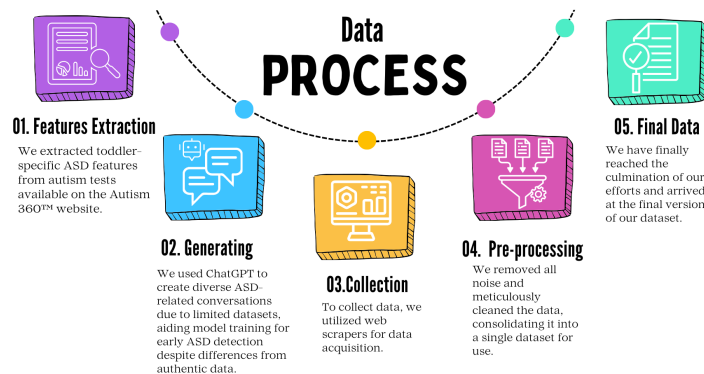
The dataset’s creation was motivated by the need to enhance early detection and intervention strategies for ASD. Leveraging artificial intelligence techniques, particularly through chatbot interactions with parents, was grounded in the belief that these interactions could yield nuanced insights into ASD-related behaviors and communication patterns in toddlers.

4.1.2 Data Collection and Preprocessing

Data collection involved utilizing web scrapers to gather necessary information from online sources. This process systematically extracted data from conversations with ChatGPT, enabling aggregation and organization of relevant information. Preprocessing steps included noise removal, duplicate data elimination, and data integrity checks to ensure a polished dataset suitable for analysis. see Figure

4.1

Figure 4.1: Data Process



4.1.2.1 Features Extraction

Features were extracted using the Autism 360™[\[4\]](#) website, which provided tools and assessments tailored for ASD-related traits in toddlers. This allowed the derivation of crucial features for analysis from specialized tests.

4.1.2.2 Generating Data

ChatGPT was utilized to generate additional data, compensating for the limited real-world data. Although generated data may differ from authentic data, this strategy broadened the dataset's diversity, aiding in model enhancement for early ASD identification.

4.1.2.3 Collection

In our data collection process, we used web scrapers to gather necessary information from online sources. These tools systematically extracted data from our conversations with ChatGPT, allowing us to aggregate and organize the relevant information for our project. Leveraging web scraping enabled us to efficiently access and compile essential data for our research and analysis, streamlining the otherwise cumbersome process of manual data collection.

4.1.2.4 Pre-Processing

We carefully removed all extra noises and duplicate data, ensuring data integrity and consistency. This thorough process involved eliminating unwanted artifacts and irrelevant information, resulting in a polished dataset. We then combined this cleaned data into a comprehensive dataset for analysis and project utilization.

4.1.2.5 Final Dataset

After meticulous curation and extensive refinement, we have arrived at the conclusive version of our dataset. This final version[\[30\]](#) encapsulates the culmination of our efforts in gathering, cleaning, and organizing the data, representing a comprehensive and refined collection ready for analysis and utilization in our project.

4.1.2.6 Data Publication and Mendeley Integration

The final dataset, after curation and refinement, was published on Mendeley. This platform enables data sharing and collaboration within the scientific community, enhancing accessibility and visibility of the dataset. The dataset's DOI (Digital Object Identifier) "Doi V1:[10.17632/87s2br3ptb.1](#)" and "Doi V2:[10.17632/87s2br3ptb.2](#)" ensures proper citation and acknowledgment in research publications.

This dataset has been viewed by more than 175 researchers and used by more than 40 researchers.

Figure 4.2: Dataset Downloads

Mendeley Data

T ASD-Dataset: Text-based Early Autism Spectrum Disorder Detection Dataset for Toddlers

Published: 12 February 2024 | Version 2 | DOI: 10.17632/87s2br3ptb.2
Contributors: Rania HACHEMI, Housseem Eddine degha

Description

The T ASD-Dataset is tailored for early ASD detection in toddlers, providing detailed textual sequences that describe the daily situations of toddlers both with and without ASD. It incorporates essential ASD assessment features such as Attention Response, Word Repetition, Emotional Empathy, and introduces new features including Noise Sensitivity, Sharing Interest, Sign Communication, and Tiptoe Flapping. Each feature is intricately linked with specific toddler behaviors, offering nuanced insights for parents and aiding in the identification of ASD-related signals. By emphasizing these features, the dataset facilitates the development of machine learning models to explore behavioral markers crucial for early ASD risk detection.

Here's a summary of the key features:

Dataset metrics

Usage

Views:	177
Downloads:	42

Latest version

Version 2	
Published:	12 Feb 2024
DOI:	10.17632/87s2br3ptb.2

[Sign In / Register](#)

4.1.3 Embedding Methods

Embedding methods were employed to convert text data into numerical formats suitable for machine learning algorithms.

4.1.3.1 Label Encoding

Categorical variables like 'Class', 'Sign', and 'ASD' were encoded using LabelEncoder from Scikit-Learn, essential for processing categorical data in machine learning models.

4.1.3.2 TF-IDF Vectorization

TF-IDF Vectorization using the TfidfVectorizer from Scikit-Learn converted text data into numerical vectors. This technique assigns weights to words based on their frequency and importance in the corpus, serving as a form of word embedding for machine learning tasks.

The integration of these embedding methods into the dataset preparation ensures compatibility with machine learning algorithms for early ASD detection in toddlers.

4.2 Data Pre-processing

In Table [4.2](#), we describe the detailed steps taken for data pre-processing before training the model.

Table 4.2: Preprocessing Techniques

Technique	Description
Tokenization	Splitting text into tokens (words, phrases, symbols)
Lemmatization	Reducing words to their base or root form
Stop-word Removal	Removing common words that do not add significant meaning to the text
TF-IDF Vectorization	Converting text data into numerical vectors
Label Encoding	Converting categorical labels into numerical format
Standardization	Scaling numerical features to have a mean of 0 and a variance of 1

4.2.1 Importing Necessary Libraries

First, we import the required libraries for data manipulation, feature extraction, model training, and evaluation.

```

1 import pandas as pd
2 from sklearn.feature_extraction.text import
   TfidfVectorizer
3 from scipy.sparse import hstack
4 import random
5 from sklearn.model_selection import train_test_split
6 from sklearn.decomposition import PCA
7 from sklearn.ensemble import RandomForestClassifier
8 from sklearn.metrics import confusion_matrix,
   accuracy_score
9 from sklearn.preprocessing import LabelEncoder
10 import matplotlib.pyplot as plt
11 import seaborn as sns

```

4.2.2 Reading Data

Next, we read the CSV file containing our dataset into a DataFrame.

```

1 file_path = 'Dataset-v2.csv'
2 df = pd.read_csv(file_path)
3
4 df = pd.DataFrame(df)

```

Figure 4.3: Reading Data

	Text	Class	Sign	ASD
0	There's societal pressure to express emotions ...	Emotional Empathy	-	ASD
1	It's fascinating to see how certain words or p...	Word Repetition	+	Non-ASD
2	Their reaction to change might impact their so...	Change Reaction	-	ASD
3	I'm concerned about our toddler's repetitive b...	Repetitive Behavior	-	ASD
4	I'm concerned about how easily our toddler get...	Attention Response	-	ASD
...
321	Their reaction to change might affect their ab...	Change Reaction	-	ASD
322	Avoiding eye contact might help my toddler fee...	Eye Contact	-	ASD
323	I've come to realize that word repetition is a...	Word Repetition	+	Non-ASD
324	Trouble in following pointing might limit thei...	Follow Pointing	-	ASD
325	I've noticed that my toddler's interest in arr...	Toy Arranging	+	Non-ASD

326 rows x 4 columns

4.2.3 TF-IDF Vectorization

TF-IDF Vectorization using the `TfidfVectorizer` from Scikit-Learn converted text data into numerical vectors. This technique assigns weights to words based on their frequency and importance in the corpus, serving as a form of word embedding for machine learning tasks.

```

1 tfidf_vectorizer = TfidfVectorizer()
2 tfidf_matrix = tfidf_vectorizer.fit_transform(df['Text '
3 ])
4 tfidf_df = pd.DataFrame(tfidf_matrix.toarray(), columns
5 =tfidf_vectorizer.get_feature_names_out())

```

4.2.4 Label Encoding

We apply label encoding to the categorical columns ('Class', 'Sign', 'ASD') to convert them into numerical format.

```

1 label_encoder = LabelEncoder()
2 df['Class'] = label_encoder.fit_transform(df['Class'])
3 df['Sign'] = label_encoder.fit_transform(df['Sign'])
4 df['ASD'] = label_encoder.fit_transform(df['ASD'])

```

Figure 4.4: Data after Label Encoding

	Text	Class	Sign	ASD
0	Changes can disrupt our established routines. ...	1	1	0
1	Repetitive behaviors might interfere with lear...	8	1	0
2	I'm concerned about how repetitive behaviors m...	8	1	0
3	I'm concerned that this repetition might inter...	13	1	0
4	I worry about my toddler's reaction to sudden ...	1	1	0
...
321	My toddler might feel isolated or excluded in ...	10	1	0
322	I worry about not knowing how my toddler might...	1	1	0
323	Their attention span for certain activities or...	0	1	0
324	My toddler gets easily distracted. making it c...	0	1	0
325	I'm concerned about my toddler's struggle in f...	6	1	0

326 rows x 4 columns

4.2.5 Concatenation and Saving

Finally, we concatenate the TF-IDF DataFrame with the original DataFrame, excluding the 'Text' column, and save the processed data to a new CSV file.

```

1 df = df.drop('Text', axis=1)
2 df = pd.concat([tfidf_df, df ], axis=1)
3 output_file_path = 'tfidf_df3.csv'
4 df.to_csv(output_file_path, index=False)
5 print(f"DataFrame has been saved to {output_file_path}"
      )

```

Figure 4.5: Data after excluding the 'Text' column

	Class	Sign	ASD
0	1	1	0
1	8	1	0
2	8	1	0
3	13	1	0
4	1	1	0
...

These steps ensure that our data is properly pre-processed and ready for training the model.

Figure 4.6: Data after Pre-processing

	abilities	ability	able	about	accept	accepting	accepts	access	\
0	0.000000	0.000000	0.0	0.000000	0.0	0.0	0.0	0.0	0.0
1	0.000000	0.000000	0.0	0.000000	0.0	0.0	0.0	0.0	0.0
2	0.000000	0.000000	0.0	0.201285	0.0	0.0	0.0	0.0	0.0
3	0.264926	0.000000	0.0	0.000000	0.0	0.0	0.0	0.0	0.0
4	0.000000	0.000000	0.0	0.164641	0.0	0.0	0.0	0.0	0.0
..
321	0.000000	0.000000	0.0	0.000000	0.0	0.0	0.0	0.0	0.0
322	0.000000	0.000000	0.0	0.195900	0.0	0.0	0.0	0.0	0.0
323	0.000000	0.153480	0.0	0.000000	0.0	0.0	0.0	0.0	0.0
324	0.000000	0.000000	0.0	0.000000	0.0	0.0	0.0	0.0	0.0
325	0.000000	0.215651	0.0	0.218841	0.0	0.0	0.0	0.0	0.0

	accessibility	accidents	...	word	words	world	worried	worry	\
0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.000000	0.0
1	0.0	0.0	...	0.0	0.0	0.0	0.0	0.000000	0.0
2	0.0	0.0	...	0.0	0.0	0.0	0.0	0.000000	0.0
3	0.0	0.0	...	0.0	0.0	0.0	0.0	0.000000	0.0
4	0.0	0.0	...	0.0	0.0	0.0	0.0	0.204135	0.0
..
321	0.0	0.0	...	0.0	0.0	0.0	0.0	0.000000	0.0
322	0.0	0.0	...	0.0	0.0	0.0	0.0	0.242891	0.0
323	0.0	0.0	...	0.0	0.0	0.0	0.0	0.000000	0.0
324	0.0	0.0	...	0.0	0.0	0.0	0.0	0.000000	0.0
325	0.0	0.0	...	0.0	0.0	0.0	0.0	0.000000	0.0

	wrongly	yet	Class	Sign	ASD
0	0.0	0.0	1	1	0
1	0.0	0.0	8	1	0
2	0.0	0.0	8	1	0
3	0.0	0.0	13	1	0
4	0.0	0.0	1	1	0
..

4.3 Data Splitting

Data Splitting: We split the data into training and testing sets using the `train_test_split` function from `sklearn.model_selection`.

```

1 from sklearn.model_selection import train_test_split
2 X_train, X_test, y_train, y_test = train_test_split(X,
  Y, test_size=0.2, random_state=42)

```

Figure 4.7: Dataset Independents Variables

able	about	accept	accepting	accepts	access	accessibility	accidents	...	witness	word	words	world	worried	worry	wrongly	yet	Class	Sign
0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	1	1
0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	8	1
0.0	0.201285	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	8	1
0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	13	1
0.0	0.164641	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.204135	0.0	0.0	1	1
...
0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	10	1
0.0	0.195900	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.242891	0.0	0.0	1	1
0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0	1
0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0	1
0.0	0.218841	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	6	1

Figure 4.8: Dataset Dependent Variable

```
0      0
1      0
2      0
3      0
4      0
..
321    0
322    0
323    0
324    0
325    0
Name: ASD, Length: 326, dtype: int64
```

4.4 Evaluation Metrics for Autism Detection

When evaluating models for the detection of autism, several key evaluation metrics are commonly used to assess their performance:

4.4.1 Accuracy

Accuracy measures the overall correctness of the model's predictions, calculated as the ratio of correctly predicted cases to the total number of cases:

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

4.4.2 Recall (True Positive Rate)

Recall, also known as Sensitivity or true positive rate (TPR), measures the proportion of actual positive cases that the model correctly identifies:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

4.4.3 Precision (Positive Predictive Value)

Precision measures the proportion of true positive predictions among all positive predictions made by the model:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

4.4.4 F1 Score

The F1 score is the harmonic mean of precision and recall, providing a balanced measure of a model's performance:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Where:

- TP: True Positives
- TN: True Negatives
- FP: False Positives
- FN: False Negatives

When evaluating models for the detection of autism, several key evaluation metrics are commonly used to assess their performance. These metrics include accuracy, recall, precision, and F1-score.

4.5 Comparison of Machine Learning Models

Before selecting the best model for our task, we compared several machine learning models, including Support Vector Machine (SVM), Artificial Neural Network (ANN), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU). This section describes the data pre-processing steps and the code for each model.

4.5.1 SVM Model

The SVM model is trained using the following steps:

1. **Model Training:** We create an SVM classifier using the `SVC` class from `sklearn.svm` and train the model using the training data.
2. **Model Evaluation:** After training the model, we make predictions on the test set and evaluate the model's performance using the `accuracy_score` function from `sklearn.metrics`.

```
1 from sklearn.svm import SVC
2 from sklearn.metrics import accuracy_score
3 # Create an SVM classifier
4 svm_model = SVC(kernel='linear', C=1.0)
5 # Train the model
6 svm_model.fit(X_train, y_train)
7 # Make predictions on the test set
```

```

8 y_pred = svm_model.predict(X_test)
9 # Evaluate the model
10 accuracy = accuracy_score(y_test, y_pred)
11 print(f"Accuracy: {accuracy}")

```

4.5.2 Artificial Neural Network (ANN) Model

In this section, we describe the architecture and training process of the Artificial Neural Network (ANN) model used for classification tasks. The ANN model consists of multiple layers of neurons that can learn complex patterns from the input data.

4.5.2.1 Model Architecture

The ANN model architecture is defined using the `Sequential` model from `tensorflow.keras.models`. It comprises several dense layers with different activation functions and dropout regularization to prevent overfitting.

```

1 from tensorflow.keras.models import Sequential
2 from tensorflow.keras.layers import Dense, Dropout
3 from tensorflow.keras.optimizers import Adam
4 # Create a deeper neural network with dropout
5 modelANN = Sequential()
6 modelANN.add(Dense(128, activation='relu',
7 input_shape=(X_train.shape[1],)))
8 modelANN.add(Dropout(0.5))
9 modelANN.add(Dense(64, activation='relu'))
10 modelANN.add(Dropout(0.3))
11 modelANN.add(Dense(264, activation='relu'))
12 modelANN.add(Dense(32, activation='relu'))
13 modelANN.add(Dense(1, activation='sigmoid'))

```

- `Sequential`: Initializes a sequential model where layers are added sequentially.
- `Dense`: Adds a fully connected layer with the specified number of neurons and activation function.
- `Dropout`: Adds dropout regularization to the network to reduce overfitting by randomly dropping neurons during training.
- `activation='relu'`: Sets the Rectified Linear Unit (ReLU) activation function for hidden layers, which introduces non-linearity.
- `input_shape=(X_train.shape[1],)`: Specifies the input shape for the first layer based on the number of features in the training data.
- `activation='sigmoid'`: Sets the Sigmoid activation function for the output layer, suitable for binary classification tasks.

4.5.2.2 Model Compilation and Training

After defining the model architecture, it needs to be compiled with an optimizer, loss function, and evaluation metrics. Then, it is trained on the training data using the `fit` method.

Table 4.3: Hyperparameters used for training the deeper neural network with dropout

Hyperparameter	Value
Learning Rate	0.0003
Batch Size	32
Number of Epochs	20
Dense Layer 1 Units	128
Dropout Rate (Layer 1)	0.5
Dense Layer 2 Units	64
Dropout Rate (Layer 2)	0.3
Dense Layer 3 Units	264
Dense Layer 4 Units	32
Output Layer Units	1
Activation Functions	ReLU (Dense), Sigmoid (Output)
Loss Function	Binary Crossentropy
Optimizer	Adam

```
1 # Compile the modelANN
2 modelANN.compile(optimizer=Adam(learning_rate=0.0003),
3                 loss='binary_crossentropy', metrics=['accuracy'])
4 # Train the modelANN
5 history = modelANN.fit(X_train_scaled, y_train, epochs
6                       =20, batch_size=32, validation_split=0.1, verbose=1)
```

- `compile`: Configures the model for training by specifying the optimizer, loss function, and evaluation metrics.
- `optimizer=Adam(learning_rate=0.0003)`: Uses the Adam optimizer with a specific learning rate for optimizing the model's weights during training.
- `loss='binary_crossentropy'`: Sets the binary cross-entropy loss function, suitable for binary classification tasks.
- `metrics=['accuracy']`: Specifies the evaluation metric as accuracy to monitor the model's performance during training.
- `epochs=20`: Defines the number of training epochs, i.e., the number of times the model will be exposed to the entire training dataset.
- `batch_size=32`: Sets the batch size for mini-batch training, where the model updates its weights after processing each batch of training data.
- `validation_split=0.1`: Splits a fraction of the training data for validation during training to monitor validation metrics and prevent overfitting.
- `verbose=1`: Displays training progress and metrics during training.

4.5.2.3 Model Evaluation

After training the ANN model, it's essential to evaluate its performance on the test set using various metrics such as accuracy, recall, precision, and F1 score.

```
1 from sklearn.metrics import recall_score, f1_score,
  precision_score
2 # Make predictions on the test set
3 y_pred = modelANN.predict(X_test_scaled)
4 # Convert continuous labels to binary using a threshold
5 threshold = 0.5 # threshold for binary classification
6 y_pred = [1 if pred >= threshold else 0 for pred in
  y_pred]
7 # Evaluate the model on the test set
8 loss, accuracy = modelANN.evaluate(X_test_scaled,
  y_test)
9 recall = recall_score(y_test, y_pred)
10 f1 = f1_score(y_test, y_pred)
11 precision = precision_score(y_test, y_pred)
12 # Print the evaluation metrics
13 print("Recall: {:.2f}".format(recall))
14 print("F1 Score: {:.2f}".format(f1))
15 print("Precision: {:.2f}".format(precision))
16 print(f'Accuracy on the test set: {accuracy * 100:.2f}%
  ')
```

- `recall_score`, `f1_score`, `precision_score`: Importing functions to calculate recall, F1 score, and precision metrics.
- `modelANN.predict`: Making predictions on the scaled test set using the trained ANN model.
- `threshold`: Setting a threshold for converting continuous prediction probabilities to binary labels.
- `loss`, `accuracy`: Calculating the loss and accuracy of the model on the test set using `modelANN.evaluate`.
- `recall`, `f1`, `precision`: Computing recall, F1 score, and precision metrics based on the true and predicted labels.
- **Confusion Matrix Visualization** The confusion matrix is a useful visualization tool to assess the performance of a classification model like the ANN. It provides insights into the model's ability to correctly classify instances into true positives, true negatives, false positives, and false negatives.

```
1 from sklearn.metrics import confusion_matrix
2 import matplotlib.pyplot as plt
3 import seaborn as sns
4 # Compute the confusion matrix
5 conf_matrix = confusion_matrix(y_test, y_pred)
6 # Plot the confusion matrix as a heatmap
```

```

7 plt.figure(figsize=(8, 6))
8 sns.heatmap(conf_matrix, annot=True, cmap='Blues', fmt
9             = 'g',
10             xticklabels=['Predicted Negative', '
11                          Predicted Positive'],
12             yticklabels=['Actual Negative', 'Actual
13                          Positive'])
14 plt.xlabel('Predicted Label')
15 plt.ylabel('True Label')
16 plt.title('Confusion Matrix')
17 plt.show()

```

- `confusion_matrix`: Computing the confusion matrix using `sklearn.metrics.confusion_matrix` based on the true and predicted labels.
- `plt.figure`, `sns.heatmap`: Creating a heatmap visualization of the confusion matrix using Matplotlib and Seaborn libraries.
- `annot=True`: Displaying the actual values in the heatmap cells for better interpretation.
- `cmap='Blues'`: Setting the color map of the heatmap to 'Blues' for a visually appealing representation.
- `xticklabels`, `yticklabels`: Setting the labels for x-axis (predicted labels) and y-axis (true labels) of the heatmap.
- `plt.xlabel`, `plt.ylabel`, `plt.title`: Adding labels and a title to the heatmap plot for clarity.

4.5.3 Convolutional Neural Network (CNN) Model

Convolutional Neural Networks (CNNs) are powerful deep learning models commonly used for image classification tasks due to their ability to capture spatial patterns in data.

4.5.3.1 Data Preprocessing

First, the features are standardized using the `StandardScaler` to ensure that they have a mean of 0 and a standard deviation of 1, which can improve the training process and convergence of the model.

```

1 from tensorflow.keras.layers import Conv1D,
2   MaxPooling1D,
3   Flatten, Dense, Dropout
4 from tensorflow.keras.optimizers import Adam
5 # Standardize the features
6 scaler = StandardScaler()
7 X_train_scaled = scaler.fit_transform(X_train)
8 X_test_scaled = scaler.transform(X_test)
9 # Reshape the data for CNN input

```

```

9 X_train_reshaped = X_train_scaled.reshape(
    X_train_scaled.shape[0], X_train_scaled.shape[1], 1)
10 X_test_reshaped = X_test_scaled.reshape(X_test_scaled.
    shape[0], X_test_scaled.shape[1], 1)

```

4.5.3.2 Model Architecture

Next, we construct the CNN model architecture using layers such as Conv1D, MaxPooling1D, Flatten, Dense, and Dropout. These layers help in extracting features from the input data and learning hierarchical representations.

```

1 # Create a CNN model
2 modelCNN = Sequential()
3 modelCNN.add(Conv1D(filters=64, kernel_size=3,
    activation='relu', input_shape=(X_train_scaled.shape
    [1], 1)))
4 modelCNN.add(MaxPooling1D(pool_size=2))
5 modelCNN.add(Conv1D(filters=32, kernel_size=3,
    activation='relu'))
6 modelCNN.add(MaxPooling1D(pool_size=2))
7 modelCNN.add(Flatten())
8 modelCNN.add(Dense(128, activation='relu'))
9 modelCNN.add(Dropout(0.5))
10 modelCNN.add(Dense(1, activation='sigmoid'))
11 # Compile the modelCNN
12 modelCNN.compile(optimizer=Adam(learning_rate=0.0003),
    loss='binary_crossentropy', metrics=['accuracy'])

```

4.5.3.3 Model Compilation and Training

We then train the CNN model using the training data and evaluate its performance on the test set to measure its accuracy.

```

1 # Train the modelCNN
2 history = modelCNN.fit(X_train_reshaped, y_train,
    epochs=20, batch_size=32, validation_split=0.1,
    verbose=1)
3 # Evaluate the modelCNN on the test set
4 loss, accuracy = modelCNN.evaluate(X_test_reshaped,
    y_test)
5 print(f'Accuracy on the test set: {accuracy * 100:.2f}%
    ')

```

Table 4.4: Hyperparameters used for training the CNN model

Hyperparameter	Value
Learning Rate	0.0003
Batch Size	32
Number of Epochs	20
Filters (Layer 1)	64
Kernel Size (Layer 1)	3
Activation (Conv Layers)	ReLU
Pooling Size (Layer 1)	2
Filters (Layer 2)	32
Kernel Size (Layer 2)	3
Pooling Size (Layer 2)	2
Dense Layer Units	128
Dropout Rate	0.5
Output Layer Units	1
Output Activation	Sigmoid
Loss Function	Binary Crossentropy
Optimizer	Adam

4.5.3.4 Model Evaluation

Finally, we visualize the training and validation accuracy, as well as the training and validation loss over epochs to analyze the model's performance and convergence.

```

1 # Plot training history
2 plt.figure(figsize=(12, 5))
3 # Plot Accuracy
4 plt.subplot(1, 2, 1)
5 plt.plot(history.history['accuracy'], label='Training
   Accuracy')
6 plt.plot(history.history['val_accuracy'], label='
   Validation Accuracy')
7 plt.title('Training and Validation Accuracy for CNN')
8 plt.xlabel('Epochs')
9 plt.ylabel('Accuracy')
10 plt.legend()
11 # Plot Loss
12 plt.subplot(1, 2, 2)
13 plt.plot(history.history['loss'], label='Training Loss'
   )
14 plt.plot(history.history['val_loss'], label='Validation
   Loss')
15 plt.title('Training and Validation Loss for CNN')
16 plt.xlabel('Epochs')
17 plt.ylabel('Loss')
18 plt.legend()
19 plt.tight_layout()
20 plt.show()

```

4.5.4 Recurrent Neural Network (RNN) Model

4.5.4.1 Model Architecture

Next, we construct the RNN model architecture using layers such as `SimpleRNN`, `Dropout`, and `Dense`. These layers are designed for sequential data processing, capturing temporal dependencies and patterns in the input sequence.

```
1 # Create an RNN model
2 model = Sequential()
3 model.add(SimpleRNN(units=64, activation='relu',
4               input_shape=(1, X_train_scaled.shape[1])))
5 model.add(Dropout(0.5))
6 model.add(Dense(32, activation='relu'))
7 model.add(Dense(1, activation='sigmoid'))
8 # Compile the model
9 model.compile(optimizer=Adam(learning_rate=0.0003),
10              loss='binary_crossentropy', metrics=['accuracy'])
```

4.5.4.2 Model Compilation and Training

Table 4.5: Hyperparameters used for training the RNN model

Hyperparameter	Value
Learning Rate	0.0003
Batch Size	32
Number of Epochs	20
RNN Units	64
Dropout Rate	0.5
Dense Layer Units	32
Output Layer Units	1
Activation Functions	ReLU (RNN, Dense), Sigmoid (Output)
Loss Function	Binary Crossentropy
Optimizer	Adam

```
1 # Compile the modelRNN
2 modelRNN.compile(optimizer=Adam(learning_rate=0.0003),
3                  loss='binary_crossentropy', metrics=['accuracy'])
4 # Train the modelRNN
5 history = modelRNN.fit(X_train_scaled, y_train, epochs
6                       =20, batch_size=32, validation_split=0.1, verbose=1)
```

4.5.4.3 Model Evaluation

Finally, we visualize the training and validation accuracy, as well as the training and validation loss over epochs to analyze the model's performance and convergence.

```
1 # Plot training history
2 plt.figure(figsize=(12, 5))
```

```

3 # Plot Accuracy
4 plt.subplot(1, 2, 1)
5 plt.plot(history.history['accuracy'], label='Training
  Accuracy')
6 plt.plot(history.history['val_accuracy'], label='
  Validation Accuracy')
7 plt.title('Training and Validation Accuracy for RNN')
8 plt.xlabel('Epochs')
9 plt.ylabel('Accuracy')
10 plt.legend()
11 # Plot Loss
12 plt.subplot(1, 2, 2)
13 plt.plot(history.history['loss'], label='Training Loss'
  )
14 plt.plot(history.history['val_loss'], label='Validation
  Loss')
15 plt.title('Training and Validation Loss for RNN')
16 plt.xlabel('Epochs')
17 plt.ylabel('Loss')
18 plt.legend()
19 plt.tight_layout()
20 plt.show()

```

4.5.5 Long Short-Term Memory (LSTM) Model

4.5.5.1 Model Architecture

Next, we create a deeper RNN/LSTM model with layers such as SimpleRNN, Dropout, LSTM, and Dense to capture long-term dependencies and improve model performance.

```

1 # Create a deeper RNN/LSTM model
2 model = Sequential()
3 model.add(SimpleRNN(units=64, activation='relu',
  input_shape=(1, X_train_scaled.shape[1]),
  return_sequences=True))
4 model.add(Dropout(0.3))
5 model.add(LSTM(units=64, activation='relu',
  return_sequences=True))
6 model.add(Dropout(0.3))
7 model.add(LSTM(units=32, activation='relu'))
8 model.add(Dropout(0.3))
9 model.add(Dense(32, activation='relu'))
10 model.add(Dense(1, activation='sigmoid'))
11
12 # Compile the model
13 model.compile(optimizer=Adam(learning_rate=0.0003),
  loss='binary_crossentropy', metrics=['accuracy'])

```

4.5.5.2 Model Compilation and Training

Table 4.6: Hyperparameters used for training the RNN/LSTM model

Hyperparameter	Value
Learning Rate	0.0003
Batch Size	32
Number of Epochs	20
RNN Units	64
LSTM Units (Layer 1)	64
LSTM Units (Layer 2)	32
Dropout Rate	0.3
Dense Layer Units	32
Activation Functions	ReLU (RNN, LSTM, Dense)
Output Activation	Sigmoid
Loss Function	Binary Crossentropy
Optimizer	Adam

```
1 # Compile the modelLSTM
2 modelLSTM.compile(optimizer=Adam(learning_rate=0.0003),
3                   loss='binary_crossentropy', metrics=['accuracy'])
4 # Train the modelLSTM
5 history = modelLSTM.fit(X_train_scaled, y_train, epochs
6                         =20, batch_size=32, validation_split=0.1, verbose=1)
```

4.5.5.3 Model Evaluation

Finally, we evaluate the LSTM model's performance by visualizing the training and validation accuracy, as well as the training and validation loss over epochs.

```
1 # Plot training history
2 plt.figure(figsize=(12, 5))
3
4 # Plot Accuracy
5 plt.subplot(1, 2, 1)
6 plt.plot(history.history['accuracy'], label='Training
7           Accuracy')
8 plt.plot(history.history['val_accuracy'], label='
9           Validation Accuracy')
10 plt.title('Training and Validation Accuracy')
11 plt.xlabel('Epochs')
12 plt.ylabel('Accuracy')
13 plt.legend()
14
15 # Plot Loss
16 plt.subplot(1, 2, 2)
17 plt.plot(history.history['loss'], label='Training Loss '
18       )
```

```

16 plt.plot(history.history['val_loss'], label='Validation
    Loss')
17 plt.title('Training and Validation Loss')
18 plt.xlabel('Epochs')
19 plt.ylabel('Loss')
20 plt.legend()
21
22 plt.tight_layout()
23 plt.show()

```

4.5.6 Gated Recurrent Unit (GRU) Model

4.5.6.1 Model Architecture

Next, we create a deeper RNN/GRU model with layers such as SimpleRNN, Dropout, GRU, and Dense to capture long-term dependencies and improve model performance.

```

1 # Create a deeper RNN/GRU model
2 modelGRU = Sequential()
3 modelGRU.add(SimpleRNN(units=64, activation='relu',
    input_shape=(1, X_train_scaled.shape[1]),
    return_sequences=True))
4 modelGRU.add(Dropout(0.3))
5 modelGRU.add(GRU(units=64, activation='relu',
    return_sequences=True))
6 modelGRU.add(Dropout(0.3))
7 modelGRU.add(GRU(units=32, activation='relu'))
8 modelGRU.add(Dropout(0.3))
9 modelGRU.add(Dense(32, activation='relu'))
10 modelGRU.add(Dense(1, activation='sigmoid'))
11
12 # Compile the modelGRU
13 modelGRU.compile(optimizer=Adam(learning_rate=0.0003),
    loss='binary_crossentropy', metrics=['accuracy'])

```

4.5.6.2 Model Compilation and Training

Table 4.7: Hyperparameters used for training the RNN/GRU model

Hyperparameter	Value
Learning Rate	0.0003
Batch Size	32
Number of Epochs	20
RNN Units (Layer 1)	64
Dropout Rate (Layer 1)	0.3
GRU Units (Layer 2)	64
Dropout Rate (Layer 2)	0.3
GRU Units (Layer 3)	32
Dropout Rate (Layer 3)	0.3
Dense Layer Units	32
Output Layer Units	1
Activation Functions	ReLU (RNN, GRU, Dense), Sigmoid (Output)
Loss Function	Binary Crossentropy
Optimizer	Adam

```
1 # Compile the modelGRU
2 modelGRU.compile(optimizer=Adam(learning_rate=0.0003),
3                 loss='binary_crossentropy', metrics=['accuracy'])
4 # Train the modelGRU
5 history = modelGRU.fit(X_train_scaled, y_train, epochs
6                       =20, batch_size=32, validation_split=0.1, verbose=1)
```

4.5.6.3 Model Evaluation

Finally, we evaluate the GRU model's performance by visualizing the training and validation accuracy, as well as the training and validation loss over epochs.

```
1 # Plot training history
2 plt.figure(figsize=(12, 5))
3
4 # Plot Accuracy
5 plt.subplot(1, 2, 1)
6 plt.plot(history.history['accuracy'], label='Training
7         Accuracy')
8 plt.plot(history.history['val_accuracy'], label='
9         Validation Accuracy')
10 plt.title('Training and Validation Accuracy for GRU')
11 plt.xlabel('Epochs')
12 plt.ylabel('Accuracy')
13 plt.legend()
14
15 # Plot Loss
16 plt.subplot(1, 2, 2)
17 plt.plot(history.history['loss'], label='Training Loss')
18 plt.legend()
```

```
16 plt.plot(history.history['val_loss'], label='Validation
    Loss')
17 plt.title('Training and Validation Loss for GRU')
18 plt.xlabel('Epochs')
19 plt.ylabel('Loss')
20 plt.legend()
21
22 plt.tight_layout()
23 plt.show()
```

4.6 Chatbot Development

4.6.1 Express.js Framework

Express.js is a popular web application framework for Node.js, designed to build web applications and APIs quickly and easily. It provides a robust set of features for web and mobile applications.

Express.js follows a modular and minimalist architecture, allowing developers to build scalable and maintainable applications. The framework structure typically includes the following components:

- **Routing:** Express.js simplifies the process of defining routes for handling different HTTP requests (GET, POST, PUT, DELETE, etc.) and mapping them to specific controller functions.
- **Middleware:** Middleware functions are a core feature of Express.js, enabling developers to execute code at various points during the request-response cycle. Middleware can be used for tasks such as authentication, error handling, logging, and more.
- **Templates:** Express.js supports various template engines such as EJS, Handlebars, Pug (formerly Jade), and Mustache for rendering dynamic HTML content on the server-side.
- **Static Files:** Express.js allows serving static files such as HTML, CSS, JavaScript, images, and other assets directly from the server's file system.
- **Error Handling:** Express.js provides mechanisms for handling errors gracefully, including built-in error-handling middleware and the ability to define custom error-handling middleware.

Important Files

When working with Express.js, several files and directories play crucial roles in the development and deployment process:

- **node_modules:** This directory contains all the external dependencies (libraries, frameworks, etc.) required by the Express.js application. Dependencies are managed using npm (Node Package Manager).
- **.vscode:** This directory may contain configuration files specific to the Visual Studio Code (VSCode) editor, such as launch configurations, settings, and extensions.
- **package.json:** The package.json file is the manifest file for the Node.js project. It includes metadata about the project, such as name, version, dependencies, scripts, and other configuration settings. Developers use package.json to manage project dependencies and define custom scripts.
- **package-lock.json:** The package-lock.json file is automatically generated by npm and serves as a lockfile to ensure deterministic dependency installation. It provides information about the exact versions of dependencies installed in the node_modules directory, along with their transitive dependencies.
- **app.js** (or **index.js**): This is the main entry point of the Express.js application. It typically initializes the Express app, sets up middleware, defines routes, and starts the server.

4.6.2 Server.js File Overview

The `server.js` file serves as the main entry point for the web application. It is responsible for configuring and starting the server, defining routes, handling HTTP requests, and interacting with various modules and services. Below is an overview of its functionality in the project:

- **Express Configuration:** The file imports the Express.js framework and sets up the application by creating an Express app instance.
- **Static File Serving:** It configures Express to serve static files, such as HTML, CSS, and JavaScript, from the specified directory (`public`).
- **Route Handling:** Various routes are defined to handle different types of requests, including:
 - Route for serving the index page (`/index`).
 - Route for serving the sign-up page (`/signup`).
 - Route for handling user login (`/login`).
 - Route for fetching child names (`/getnames`).
 - Route for inserting conversation data (`/insertConversation`).
 - Route for terminating a chat session (`/terminateChat`).
 - Route for handling chat messages (`/chat`).
 - Route for user sign-up (`/signup`).
 - Route for checking if a user is a parent (`/isparent`).
 - Route for fetching conversations (`/conversations`).

- Route for fetching messages (`/messages`).
 - Route for saving child information (`/saveChild`).
 - Route for creating a conversation for a child (`/createConversation`).
 - Route for handling child chat messages (`/chatchild`).
 - Route for inserting metric data (`/insertmetric`).
 - Route for adding favorites (`/addFavorite`).
- **Database Interaction:** The file establishes a connection to the MySQL database and executes queries to perform operations such as user authentication, data insertion, and data retrieval.
 - **Google Generative AI Integration:** It includes a function (`runChat`) to interact with the Google Generative AI model for generating chat responses based on user input.
 - **Server Initialization:** Finally, the server is started to listen for incoming HTTP requests on the specified port (either from the environment variable or defaulting to port 4000).

The `server.js` file acts as the backbone of the web application, orchestrating the flow of data and interactions between the client, server, and external services.

Explanation of Methods in `server.js`

Method: `app.post('/signup')`

This method handles user sign-up requests. It receives user data such as username, email, password, and whether the user is a parent from the request body. It then checks if the provided email already exists in the database. If not, it inserts the user data into the database and initiates a new conversation for the user.

Method: `app.post('/login')`

This method handles user login requests. It verifies the provided email and password against the database records. If the login credentials are valid, it extracts the user ID and email, initiates a new conversation for the user, and retrieves additional information such as whether the user is a parent and the names of associated children.

Method: `app.post('/isparent')`

This method checks whether a given user is a parent based on their user ID. It queries the database to retrieve the user's parent status and sends a response with the boolean value indicating whether the user is a parent or not.

Method: `app.get('/conversations')`

This method retrieves conversation data for a specific user based on their user ID. It queries the database to fetch all conversations associated with the user and returns the conversation data in the response.

Method: `app.post('/chat')`

This method handles incoming chat messages from the user. It receives the user input, generates a response using the `runChat` function, and stores both the user input and the bot response in the database with the corresponding conversation ID. It ensures that the conversation ID is set and properly initialized before storing the messages.

Method: `app.post('/terminateChat')`

This method terminates a chat session by updating the end time of the conversation in the database. It receives the end time from the request body and updates the corresponding record in the conversations table.

Method: `app.get('/messages')`

This method retrieves messages for a specific conversation based on the conversation ID. It retrieves all messages associated with the conversation from the database and returns them in the response.

Method: `app.post('/saveChild')`

This method is responsible for saving child information into the database. It extracts the child's name and age from the request body, performs any necessary validation checks, and then inserts the child's information into the database.

Method: `app.post('/createConversation')`

This method creates a new conversation in the database, typically initiated by a user. It extracts the necessary parameters from the request body, such as the user ID and start time, and inserts a new conversation record into the database.

Method: `app.post('/chatchild')`

This method handles chat messages within an existing conversation with a child. Similar to the `app.post('/chat2')` method, it receives user input, generates a response using the `runChat` function, and then stores both the user input and the

bot response in the database, associating them with the correct conversation ID for the child.

Method: `app.post('/createConversationchild')`

This method creates a new conversation for a child. It retrieves the child ID from the request body, along with the parent's user ID, and inserts a new row into the `conversations` table with the start time and child ID, effectively initiating a conversation for the child.

Method: `app.post('/insertmetric')`

This method inserts metrics into the database. It retrieves the metric name and value from the request body, checks if they are provided, and then inserts the metric data into the `asd metrics` table in the database, associating it with the appropriate conversation ID.

Method: `app.post('/addFavorite')`

This method is responsible for updating the favorite status of a conversation. It receives the new favorite status and the conversation ID from the request body, then updates the corresponding record in the database with the new favorite status.

Method: `app.get('/getConversationMessage')`

This method retrieves conversation messages for a specific child. It takes the child's name from the query parameters, queries the database to obtain the child ID and conversation ID, and then retrieves the messages associated with that conversation ID from the `messages` table. Finally, it sends the conversation messages as a response to the client.

Function: `async function runChat(userInput)`

This asynchronous function is responsible for handling the chat interaction with the Google Generative AI model. It takes user input as a parameter and returns a response generated by the AI model. Here's a breakdown of its functionality:

- It initializes the Google Generative AI with the provided API key and retrieves the generative model named "gemini-pro".
- Defines the generation configuration including parameters like temperature, topK, topP, and maxOutputTokens, which control the behavior of the AI model during text generation.
- Sets safety settings to block harmful content, specifying categories of harmful content and the threshold for blocking.

- Starts a chat session with the generative model, providing the generation configuration, safety settings, and an empty history array (indicating no pre-defined conversation history).
- Sends the user input to the AI model using the `sendMessage` method of the chat session and awaits the response.
- Processes the response, removing any lines containing stars (*) to filter out potentially undesirable content.
- Returns the generated response to the caller.
- Handles errors by logging them and throwing the error to be caught by the caller.

4.6.3 `signup.js` File Overview

The `signup.js` file is responsible for handling user interactions related to signing up and logging in. It includes functions to toggle between the login and signup forms, submit login data to the server, and handle user signup requests. The file ensures smooth user experience and handles asynchronous operations for form submissions.

Function: Toggle Between Login and Signup Forms

This JavaScript function toggles the visibility of login and signup forms, providing a seamless user experience.

Method: `app.post('/signup')`

This method handles user signup requests. It collects the user details from the request, including username, email, password, and whether the user is a parent. It then inserts this information into the database, ensuring the user is registered. If the registration is successful, it sends a success response and redirects to the index page.

4.6.4 `Script.js` File Overview

The `script.js` file contains the JavaScript code for the chatbot application. It handles user interaction, sends messages to the server, receives responses, and updates the chat interface accordingly. Additionally, it includes functions for presenting questions, handling user clicks on answer options, calculating autism status, and more.

Code Explanation

Function: `fetchAndDisplayMessages`

Description: This asynchronous function fetches messages from the server and displays them in the chat interface. It handles the retrieval of chat messages from the backend and updates the UI accordingly.

Function: `fetchChildNames`

Description: This asynchronous function fetches the names of children associated with a user from the backend server. It sends a request to the server endpoint `‘/getnames‘` to retrieve the child names. Upon receiving a successful response, it extracts the child names from the response data and logs them to the console. It then calls the `‘displayChildNames‘` function to display the child names in the sidebar of the application.

Parameters:

- `userId` (string): The ID of the user whose children’s names are to be fetched.

Returns:

- `childNames` (array): An array containing the names of the children fetched from the server.

Function: `closeSidebarOnSmallScreens`

Description: This function is responsible for closing the sidebar when the application is viewed on small screens. It adjusts the display of the sidebar to ensure better user experience and readability.

Function: `addFavorite`

Description: This function allows users to add a conversation to their list of favorites. It implements the functionality to mark conversations as favorites and updates the UI accordingly.

Function: `displayConversations`

Description: This function is responsible for displaying conversations in the chat interface. It formats the conversation data and renders it in the chat window for the user to view.

Function: saveChildInfo

Description: This function is responsible for saving information about a child in the application. It captures the child's details entered by the user and stores them securely in the system.

Function: saveChildToDatabase

Description: This asynchronous function saves information about a child to the database. It communicates with the backend server to store the child's data securely and reliably.

Function: createConversationForChild

Description: This asynchronous function creates a new conversation for a child in the chat application. It initializes a conversation session for a specific child, allowing users to interact with the bot on behalf of that child.

Function: updateConversationListUI

Description: This function updates the user interface to display the list of conversations. It refreshes the conversation list view to reflect any changes in the conversations, such as new messages or updates.

Function: updateChildListUI

Description: This function updates the user interface to display the list of children. It adds a new child item to the child list view with the provided name, age, and ID.

Parameters:

- **name** (string): The name of the child to be displayed.
- **age** (number): The age of the child.
- **id** (string): The ID of the child.

Function: createChildListItem

Description: This function creates a new item in the child list view. It generates HTML markup for displaying information about a child, including their name, age, and ID.

Parameters:

- `name` (string): The name of the child.
- `age` (number): The age of the child.
- `id` (string): The ID of the child.

Function: `displayChildNames`

Description: This asynchronous function retrieves and displays the names of children in the application. It fetches child names from the backend server and renders them in the user interface.

Function: `fetchConversationMessage`

Description: This asynchronous function retrieves messages from a conversation. It communicates with the backend server to fetch messages associated with a specific conversation.

Function: `saveMessage`

Description: This asynchronous function saves a message to the database. It sends the message data to the backend server for storage and logging purposes.

Function: `startNewChat`

Description: This function initiates a new chat session in the application. It handles the process of starting a conversation with the chatbot, enabling users to interact with the bot.

Function: `copyCodeToClipboard`

Description: This function enables copying code content to the clipboard when the user clicks the corresponding button. It extracts the code content from within triple backticks, copies it to the clipboard, and displays a notification to the user indicating successful copying.

Parameters:

- `button` (HTMLElement): The button element clicked by the user to copy the code.

Function: `displayConversationMessage`

Description: This function displays messages in a conversation. It renders messages on the chat interface for users to view and interact with.

Parameters:

- `messages` (array): An array of messages to be displayed in the conversation.

Function: `presentQuestion`

Description: This function is responsible for presenting the question and choices to the user in the chat interface.

Parameters:

- `question` (Object): The question object containing the question text and choices.

Function: `askNextQuestion`

Description: This function is responsible for presenting the next question in the chat interface. It checks the current question index and section indices to determine the appropriate question to present. If there are more questions remaining, it presents the next question; otherwise, it calls the function to calculate the autism status.

Parameters:

- `index` (number): The index of the current question.

Function: `calculateAutismStatus`

Description: This function calculates the autism status based on the user's responses to the questions. It compares the number of positive and negative responses and provides recommendations or information accordingly.

Function: `sendMessage`

Description: This asynchronous function is called when the user sends a message. It retrieves the user's input, sends it to the appropriate endpoint based on the conversation context, and awaits the bot's response. Once the response is received, it processes and displays it in the chat interface. It also handles errors gracefully and inserts the conversation into the database for logging purposes.

4.6.5 HTML Part

`index.html`

The `index.html` file serves as the main interface for the ASD Chatbot application. Here's a brief overview:

- **Structure:** The file starts with typical HTML structure, including the `<head>` and `<body>` sections.
- **Dependencies:** Various CSS and JavaScript libraries are included, such as Bootstrap, Font Awesome, and LineIcons, to style and enhance the functionality of the page.
- **Sidebar:** The sidebar contains navigation links for creating new chats and managing child profiles. It also displays a list of conversations and provides a logout option.
- **Main Content:** The main content area contains the chat interface, including the chat history and input form. Messages are displayed using cards with appropriate styling.
- **Modal:** A modal window (`childInfoModal`) is used for entering child information, such as name and age.
- **JavaScript:** Custom JavaScript code (`script.js`) is included at the end of the file to handle dynamic behavior and interactions.

Signup.html

The `signup.html` file is responsible for handling user authentication and registration. Here's a brief description:

- **Structure:** Similar to `index.html`, it starts with standard HTML structure, including the `<head>` and `<body>` sections.
- **Dependencies:** Bootstrap and Font Awesome libraries are included for styling and icons.
- **Login and Signup Forms:** Two forms are present, one for logging in and another for signing up. Initially, only the login form is visible. The signup form is hidden and can be accessed by clicking the "Register" link.
- **Form Fields:** Both forms contain input fields for username, email, and password. The signup form also includes a checkbox for indicating whether the user is a parent.
- **JavaScript:** Custom JavaScript code (`signup.js`) is included at the end of the file to handle form submissions and toggle between the login and signup forms.

4.6.6 CSS Part

Signup.css Overview

The `signup.css` file is responsible for styling elements related to the signup page or component. It contains CSS rules that define the appearance and layout of various components such as form elements, dividers, and containers. Below is a summary of its key features:

- **Divider Styling:** The file defines CSS rules for creating horizontal divider lines using pseudo-elements (`:before` and `:after`) with a background color of `#eee`. These dividers are likely used to visually separate different sections or components on the signup page.
- **Custom Height:** The `.h-custom` class is used to set the height of specific elements dynamically. It utilizes the `calc()` function to calculate the height as a percentage of the viewport height minus a fixed value (73 pixels). This ensures that the element fills the available vertical space while accommodating a fixed header size.
- **Responsive Design:** The file includes a media query (`@media (max-width: 450px)`) to adjust styles for smaller screen sizes. Within this media query, the height calculation for `.h-custom` is modified to occupy the full viewport height (100%). This ensures that the layout remains responsive and adapts appropriately to different screen sizes.

Style.css Overview

The `style.css` file contains global styles and rules that apply to multiple pages or components within the web application. It defines the overall visual theme, layout properties, and styling for various HTML elements. Here's an overview of its general features:

- **Global Styling:** The file sets default styles for HTML elements such as `html`, `body`, `a`, `li`, and `h1`. This ensures consistent typography, spacing, and other visual attributes across the application.
- **Layout Management:** CSS rules are provided to manage the layout of elements, including centering content vertically, adjusting margins, and controlling the width of containers. The `flex` property is utilized for flexible and responsive layout designs.
- **Scrollbar Styling:** Custom styling is applied to scrollbar elements using vendor-specific pseudo-elements (`::-webkit-scrollbar`) and properties. This allows for

4.6.7 Dataset.csv

The `Dataset.csv` file contains structured data related to various aspects of toddler behavior and characteristics. Each row represents a different observation, with columns providing information such as text descriptions, associated classes or categories, and signs indicative of Autism Spectrum Disorder (ASD). Here's a snippet of the dataset:

4.6.8 Setup Instructions

To set up and run the application locally, follow these instructions:

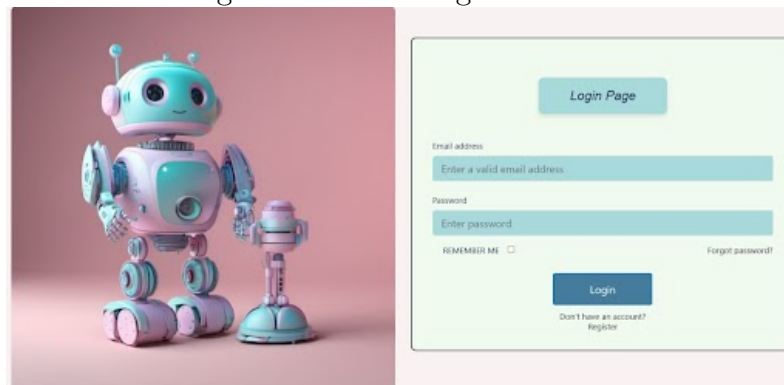
1. Clone this repository to your local machine.
2. Install the dependencies using `npm install`.
3. Set up a MySQL database and configure the connection details in the `app.js` file.
4. Obtain an API key for Google Generative AI and set it as an environment variable in a `.env` file.
5. Run the application using `npm start`.
6. Access the application through a web browser at `http://localhost:4000`.

4.7 ASDetectBot APP

4.7.1 User Login

Users provide their credentials (email and password) to log in. The application verifies the credentials and grants access to the chat interface.

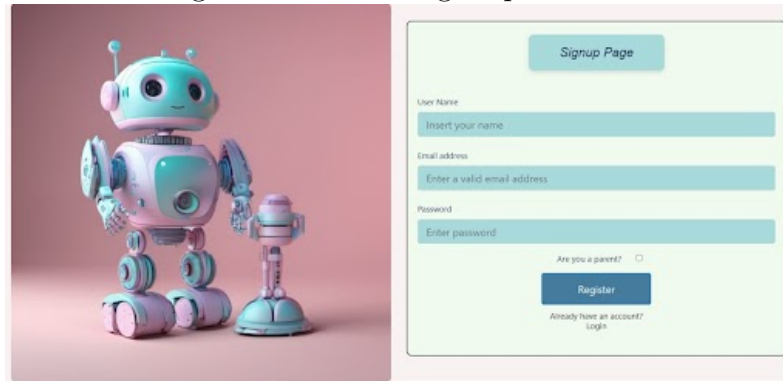
Figure 4.9: User Login Interface



4.7.2 User Sign Up

New users register by providing their username, email, password, and parental status. The application validates the user information and creates a new user account.

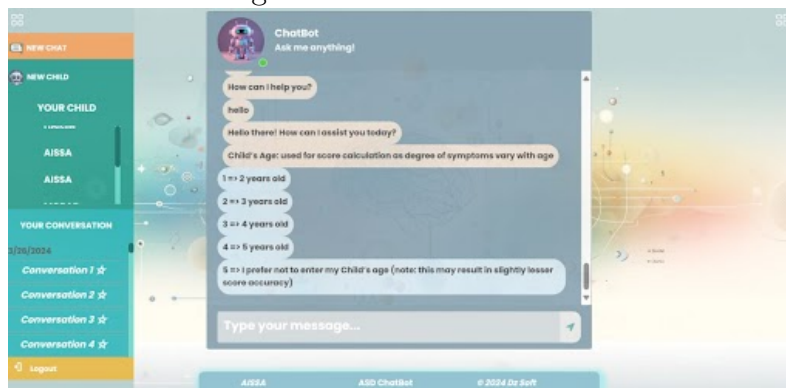
Figure 4.10: User Sign Up Interface



4.7.3 Chat Interface

Users interact with the chatbot through a user-friendly chat interface. The interface supports intuitive controls, making navigation straightforward and accessible. It implements natural language processing to understand and respond to user inquiries efficiently. Features such as voice-to-text enhance accessibility and user engagement, while real-time feedback confirms the chatbot's understanding of the conversation.

Figure 4.11: Chat Interface



4.7.4 Chatbot Response Mechanism

The chatbot generates responses based on user input and context, providing relevant information or assistance. It employs advanced AI algorithms to ensure responses are contextually appropriate and timely. The chatbot can adapt to user preferences and learn over time to improve interaction quality, handling a diverse range of topics and queries with high accuracy while prioritizing user data privacy and security.

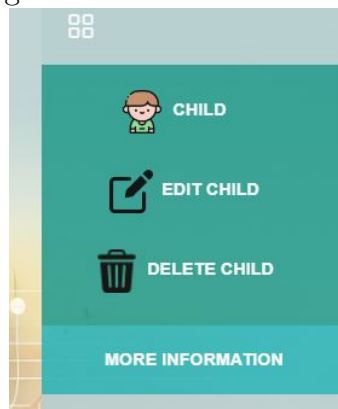
Figure 4.12: Response Mechanism



4.7.5 Parental Controls

Based on parental status, the chatbot adapts responses and functionality to ensure appropriate content and interaction. Parents can access additional features and settings with the chatbot to manage their child's experience. Parents can set time limits and usage alerts to manage chat interactions, and detailed reports and analytics on children's activities and interactions within the app are provided.

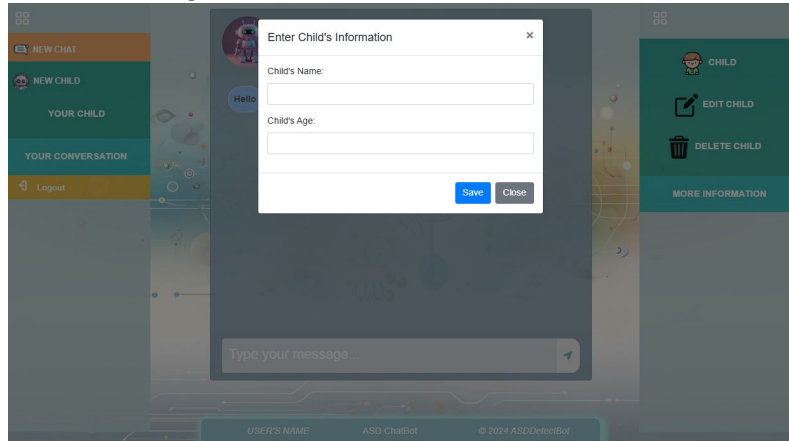
Figure 4.13: Parental Controls



4.7.6 Child Creation

Parents can create profiles for their children within the application, providing necessary information such as name, age, and preferences. This allows for customization of the user experience based on age-specific needs and interests. The application supports multi-profile management, enabling parents to oversee several child profiles simultaneously, ensuring all profiles are secure and private with data encryption and restricted access.

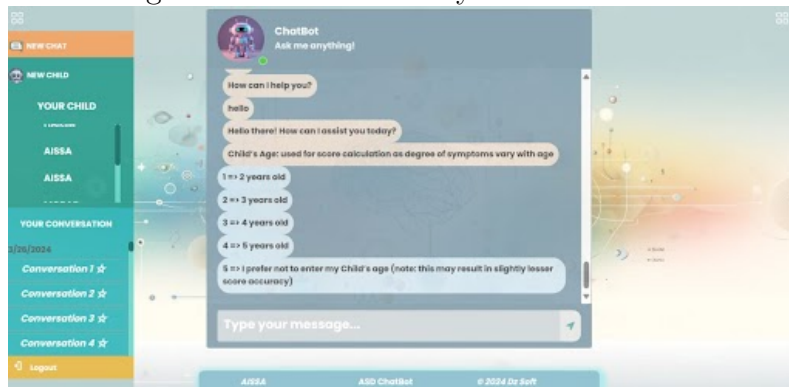
Figure 4.14: Child Creation Interface



4.7.7 Chat History Access

Users can retrieve their conversation history, accessing past interactions with the chatbot. This feature enables users to review previous discussions or reference information shared during conversations. It supports search functionality to quickly find specific conversations or topics and includes options to export chat history for personal record-keeping or further analysis. The system maintains a secure and encrypted database to protect the privacy and integrity of chat histories.

Figure 4.15: Chat History Access Interface



Conclusion

In this chapter, we discussed the development of the T ASD-Dataset for the early detection of ASD in toddlers, which is crucial for early childhood intervention and autism research. We employed several exploratory data preprocessing techniques, including label encoding and TF-IDF vectorization, to ensure compatibility with subsequent ML models. Publishing the dataset on Mendeley has facilitated collaboration and further research. Our progress with the chatbot development in Express.js highlights its scalability and maintainability, supported by modular features that ensure clean code and efficient database interactions. Notably, the user-friendly interfaces of ASDetectBot were developed in collaboration with 3D

students, who used the project as a reference for their work. Overall, ASDetectBot proves to be a valuable tool for early ASD detection in toddlers.

Chapter 5

Result and Evaluation

Introduction

The following chapter is devoted to the presentation of the results and evaluations of the proposed ASD detection system. Beginning with the role and purpose of developing the ASD assessment questionnaire for toddlers, its significance to early identification of the disorder, and the credibility of the assessment tool. The scoring guidelines for the questionnaire are described, including age norms for the scores, general guidelines for scoring the questionnaire, ASD indicators according to the scores, and an ethical code for administering the questionnaire. In continuing, the chapter examines the performances of different varieties of deep learning models like ANN, CNN, RNN, LSTM and GRU in diagnosing ASD toddlers.

5.1 Questionnaire's Design

5.1.1 Questionnaire Design Rationale

The ASD Test Questionnaire is meticulously designed to capture a comprehensive range of behavioral, social, communication, sensory, and motor skills exhibited by toddlers. The questionnaire's design is rooted in the need for a reliable and standardized tool to aid in the early detection and assessment of ASD, particularly in young children where early intervention is crucial.

5.1.2 Objective

The primary objective of the ASD Test Questionnaire is to systematically evaluate and quantify key indicators and behaviors associated with ASD in toddlers. By gathering detailed information across multiple domains, the questionnaire aims to provide a holistic view of the child's developmental profile, identifying potential red flags or areas of concern that may warrant further assessment or intervention. We utilized the Autism 360™ [4] to generate these questions.

5.1.3 Relevance to ASD Assessment in Toddlers

1. **Early Detection:** The questionnaire's relevance lies in its ability to facilitate early detection of ASD symptoms in toddlers, recognizing subtle signs that may indicate developmental differences or challenges.
2. **Comprehensive Assessment:** Through a structured set of questions covering observation skills, social interactions, communication abilities, sensory sensitivities, motor skills, and behavioral patterns, the questionnaire offers a comprehensive assessment framework tailored to the developmental stage of toddlers.
3. **Scoring and Interpretation:** The questionnaire's scoring system and interpretation guidelines are designed to assist healthcare professionals, educators, and caregivers in objectively evaluating the child's responses and behavior, enabling a standardized approach to ASD assessment.

4. **Age-Appropriate Evaluation:** Recognizing that ASD symptoms may manifest differently at various developmental stages, the questionnaire incorporates age-appropriate questions and scoring criteria, ensuring relevance and accuracy in assessment outcomes.

Table 5.1: Questionnaire Data

Question	Description
Child's Age	Used for score calculation as degree of symptoms vary with age
I prefer not to enter my Child's age	Note: this may result in slightly lesser score accuracy
Child's Gender	Used for score calculation as boys are 3 times more likely to have autism than girls
Your toddler usually looks at you when you call his/her name	Checks if the child responds to their name, indicating social engagement
Your toddler usually makes eye contact when engaged in a conversation	Assesses the child's ability to make and maintain eye contact
How long do you think your toddler usually spends looking at a spinning object?	Measures fixation on spinning objects, a common repetitive behavior
If you point at something, your toddler is usually able to follow where you are looking at, or what you are pointing to	Evaluates joint attention skills
Does your toddler make unusual finger movements in front of his/her eyes?	Observes repetitive or unusual motor behaviors
How would you rate your child's sensitivity to noise?	Assesses hypersensitivity or hyposensitivity to auditory stimuli
Your toddler points at things to share that she/he might find interesting for example a puppy, fireworks or an interesting event	Checks for protodeclarative pointing, indicating shared interest
Your toddler can usually understand if you, or someone in the immediate family (your partner, siblings, etc) are visibly upset and she/he attempts to comfort you (or the other member) for example, by throwing a hug or stroking your (or the other family member's) hair, etc	Measures empathy and response to others' emotions

Table 5.2: Questionnaire Data .2

Your toddler is usually good at interpreting simple gestures and would return the same for example waving bye, or saying hello/hi, etc	Assesses understanding and use of nonverbal communication
Your toddler spontaneously looks at you when faced with an unfamiliar situation	Checks for seeking social referencing in uncertain situations
You and immediate family members can easily understand what your toddler may be trying to say (even though s/he may not have fully developed speech skills yet)	Evaluates communicative clarity and intention
Your toddler is adept at using sign language to point out what s/he wants for example a toy, or a cookie jar that may be out of reach	Measures alternative communication methods
The best way to describe YOUR emotion when you heard your toddler's first few words would be? choose the emotion that best resembled your emotion	Captures emotional reaction to developmental milestones
Does your toddler have a habit of repeating words exactly the way they are told? e.g. she/he repeats a question that she/he was asked, or lines from a movie or a song	Observes echolalia, a repetitive speech behavior
Your toddler always likes to line up his/her toys, or arrange them in 'a' specific order, every time! your toddler's mind works in a particular pattern, and that is usually consistent with a few things she/he does	Checks for repetitive and restrictive behaviors
Your toddler is usually good at pretending while playing for example, with a toy phone or a doll	Assesses imaginative and pretend play abilities
Your toddler has a habit of licking unusual objects or sniffing at things for example, sniffing at food before eating or sniffing at her clothes before wearing	Evaluates unusual sensory behaviors
Your toddler is usually good at using his/her hands in the right way when he/she needs to do something for example, placing a hand on the door knob to open the door, or on the toy phone keypad as if pretending to call	Measures fine motor skills and practical problem-solving
Does your child do tiptoe and/or Flaps hand are regular frequencies?	Observes for common self-stimulatory behaviors
How does your child react when there is a sudden change in plan or some objects are out of their usual place?	Assesses response to changes and flexibility
Your toddler has an odd habit of doing the same thing over and over again for example, running and closing the tap, turning a switch on and off or opening and closing the door over and over again	Observes repetitive actions and routines
For how long is your toddler's interest maintained in just one or two objects that he/she can't seem to get his/her mind off?	Measures intensity and duration of focused interests

5.1.4 Discussion of Validity and Reliability

The validity and reliability of the ASD Test Questionnaire were assessed through rigorous psychometric analysis, including item correlations, factor analysis, and construct validity testing.

Validity

The questionnaire's items were analyzed for correlations to ensure that they measure the intended constructs related to ASD symptoms in toddlers. High correlations between items within the same construct indicate good content validity.

Factor analysis was conducted to identify underlying factors or dimensions measured by the questionnaire. This process helps in establishing the questionnaire's structural validity by confirming the grouping of items into meaningful factors.

The questionnaire underwent construct validity testing to assess its ability to measure the theoretical constructs of ASD in toddlers. This involved comparing questionnaire scores with established ASD assessment tools to demonstrate convergent and discriminant validity.

Reliability

Cronbach's alpha coefficient was calculated to assess the internal consistency of the questionnaire items. A high alpha value indicates that the items within the questionnaire are highly correlated, demonstrating reliability in measuring the same underlying construct.

Inter-rater reliability was assessed by having multiple raters independently score the same set of questionnaires. Consistency in scores among raters indicates good inter-rater reliability.

The psychometric analysis results provide evidence supporting the validity and reliability of the ASD Test Questionnaire as a robust tool for assessing ASD symptoms in toddlers.

5.2 Interpretation Guidelines for ASD Test Questionnaire Scores

5.2.1 Age-Appropriate Variations

Scores on the ASD Test Questionnaire should be interpreted with consideration for age-appropriate variations in ASD symptoms and behavioral indicators. Younger children may exhibit different behaviors compared to older children, and this should be taken into account during score analysis.

5.2.2 Scoring Guidelines

Child's Age

The child's age is used for score calculation as the degree of symptoms may vary with age. Select the appropriate age category: 2 years old, 3 years old, 4 years old, 5 years old, or indicate a preference not to enter the child's age (note: this may result in slightly lesser score accuracy).

Child's Gender

The child's gender is used for score calculation, as boys are three times more likely to have autism than girls. Select the gender: Girl, Boy, or prefer not to say.

Observation and Cognitive Skills

Scores in this section assess the child's observation and cognitive skills.

- Always/Sometimes: Negative (-)
- Usually/Rarely/Never: Positive (+)

Social Interactions

Scores in this section evaluate the child's social skills.

- More frequent social behaviors: Negative (-)
- Less frequent social behaviors: Positive (+)

Communication Skills

Scores in this section measure the child's communication abilities.

- Better communication skills: Negative (-)
- Limited communication skills: Positive (+)

Behavior, Sensory & Motor Skills

Scores in this section assess the child's behavior, sensory functions, and motor skills.

- Typical behaviors: Negative (-)
- Atypical behaviors: Positive (+)

Interpretation

- Overall score: More positive responses indicate a higher likelihood of ASD.
- Age-specific considerations: Some questions may be more relevant or indicative of ASD symptoms based on the child’s age.
- Consultation: Interpretation should be done in consultation with healthcare professionals or specialists for accurate diagnosis and intervention planning.

Table 5.3: Key Interpretations for ASD Assessment Criteria

Section	Interpretation
Age-Appropriate Variations	Scores should consider age-related variations in ASD symptoms and behaviors. Younger children may exhibit different behaviors than older children.
Child’s Age	Select the appropriate age category. Not entering age may slightly affect score accuracy.
Child’s Gender	Boys are three times more likely to have autism than girls.
Observation and Cognitive Skills	More positive responses indicate typical behaviors, while more negative responses may indicate atypical behaviors.
Social Interactions	More frequent social behaviors are typically negative, while less frequent social behaviors may indicate positive ASD indicators.
Communication Skills	Better communication skills are typically negative, while limited communication skills may indicate positive ASD indicators.
Behavior, Sensory & Motor Skills	Typical behaviors are typically negative, while atypical behaviors may indicate positive ASD indicators.
Overall Score	More positive responses across sections indicate a higher likelihood of ASD.
Age-Specific Considerations	Some questions may be more relevant based on the child’s age.
Consultation	Interpretation should be done in consultation with healthcare professionals or specialists.

5.2.3 Scores and ASD Indicators

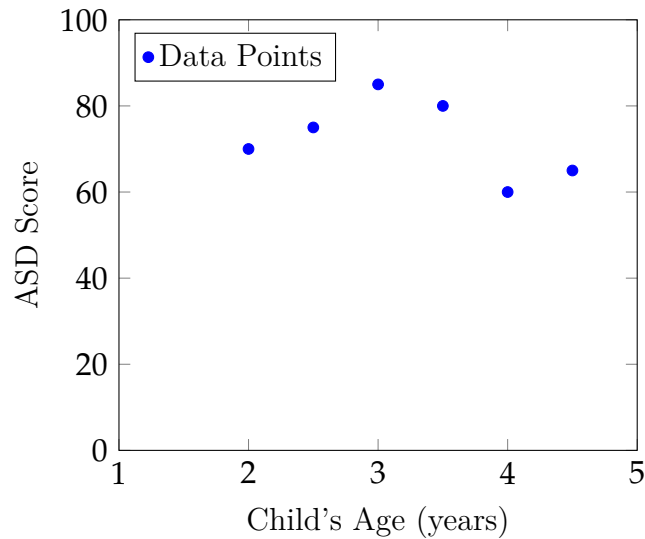
Table 5.4: Scores and ASD Indicators

Question	ASD Indicator
Child's Age	Used for score calculation; degree of symptoms vary with age
Child's Gender	Boys more likely to have autism than girls
Observation Skills	Rarely or Never looking when called may indicate ASD
Eye Contact	Rarely or Never making eye contact may indicate ASD
Spinning Object	Spending several hours a day looking may indicate ASD
Following Pointing	Hardly ever following pointing may indicate ASD
Finger Movements	Very frequently or Many times may indicate ASD
Sensitivity to Noise	Sad or Grumpy may indicate ASD
Social Skills	Less frequent social behaviors may indicate ASD
Understanding Emotions	Rarely or Never attempting to comfort may indicate ASD
Communication	Rarely or Never understandable speech may indicate ASD
Sign Language	Rarely or Never using sign language may indicate ASD
Emotion at First Words	Non-Verbal or Confused may indicate ASD
Repeating Words	Several times a day or Hardly ever may indicate ASD
Behavioral Patterns	Always or Usually lining up toys may indicate ASD
Pretending	Slightly or Strongly disagreeing may indicate ASD
Licking/Sniffing	Several times a day may indicate ASD
Motor Skills	Hardly ever using hands correctly may indicate ASD
Tiptoe/Hand Flaps	Only Tiptoes or Only hand flaps may indicate ASD
Reaction to Change	Angry or Strongly Disagree may indicate ASD
Repetitive Behavior	Strongly Agree may indicate ASD
Attention Span	Very short attention span may indicate ASD
Twiddling Objects	Strongly Agree may indicate ASD
Staring at Something	Strongly Agree may indicate ASD
Free Time Preferences	Preferring Gadgets or Alone may indicate ASD
Recognition	Inability to recognize numbers or shapes may indicate ASD
Comforting Methods	Preferring Gadgets or Sugar may indicate ASD
Usual Mood	Mischievous, Confused, Grumpy, or Non-Verbal may indicate ASD

Description

The table [5.4](#) provides an analysis of scores and ASD (Autism Spectrum Disorder) indicators based on responses to specific questions related to child development and behavior. Each question is associated with potential ASD indicators, highlighting behaviors that may suggest the presence of ASD in toddlers. The indicators range from observation skills and social interactions to communication abilities, sensory sensitivities, and behavioral patterns. The table categorizes responses as potentially positive (+) or negative (-) indicators of ASD, with additional considerations for neutral (0) responses. Understanding these indicators can aid in early detection and assessment of ASD in toddlers.

Figure 5.1: Relationship Between Age and ASD Score



The scatter plot (see figure 5.1) illustrates the relationship between a child's age and their Autism Spectrum Disorder (ASD) score. Each data point represents a hypothetical assessment result, with the x-axis showing the child's age in years and the y-axis indicating the ASD score. The plot suggests a trend where older children tend to have higher ASD scores, although there is variability within age groups. This visualization helps in understanding potential correlations between age and ASD severity in toddlers.

5.2.4 Ethical Considerations in ASD Assessments

Participant Consent

Participants, especially parents or guardians of toddlers, should provide informed consent before their child's participation in the ASD assessment. Consent forms should clearly outline the purpose of the assessment, potential risks and benefits, and confidentiality measures.

Data Confidentiality

All data collected during the ASD assessment, including questionnaire responses and personal information, must be kept confidential. Data should be securely stored and only accessible to authorized personnel involved in the research project.

Adherence to Research Ethics Guidelines

Researchers conducting ASD assessments should adhere to established research ethics guidelines, such as those outlined by institutional review boards (IRBs) or ethics committees. This includes obtaining ethical approval before commencing the

study, ensuring the well-being and rights of participants, and reporting findings accurately and transparently.

5.3 Training and Validation Metrics

5.3.1 Discussion of Results of ANN Model

We can see from the plotted and tabulated ANN model performance over 20 epochs that. Figure 5.2 present where training and validation accuracy of the ANN model are illustrated during 20 epochs. First off, both training and validation accuracies strongly decrease which, typically, occurs at the beginning of the training phases, but gradually start rising as more epochs pass. The training accuracy is also very high, and it touches over 99% by the 13th epoch, and it keeps high values till the rest of the epochs are run. On the same note, the validation accuracy chart exhibits a constant increase that ends up in a maximum of (96%). By 11th epoch there reaches a 15% and it is maintained through 12th epoch. The fact that the model is performing well and generalizes to unseen data depicts its ability to learn well and generalize.

Figure 5.2: Training and Validation Accuracy of ANN Over Epochs

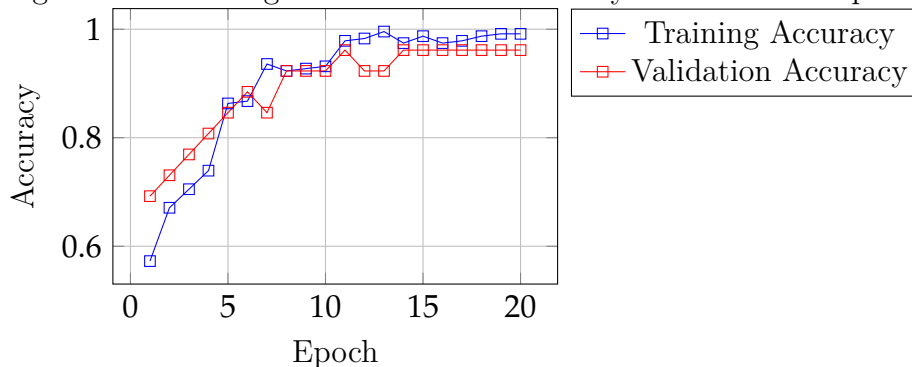


Figure 5.3 exhibits the loss of the ANN model during the training and validation stages 0.64 over the same time frame. The quantity of training loss almost comes near to the optimal value as 0.6903 to 0. The following is the loss function for the neural network: 0.0292 that shows the model’s capacity for weeding out mistakes on the training data. Similarly, the validation loss was seen to be dropping, continuously from a starting point of 0.6730 to 0.

Figure 5.3: Training and Validation Loss of ANN over Epochs

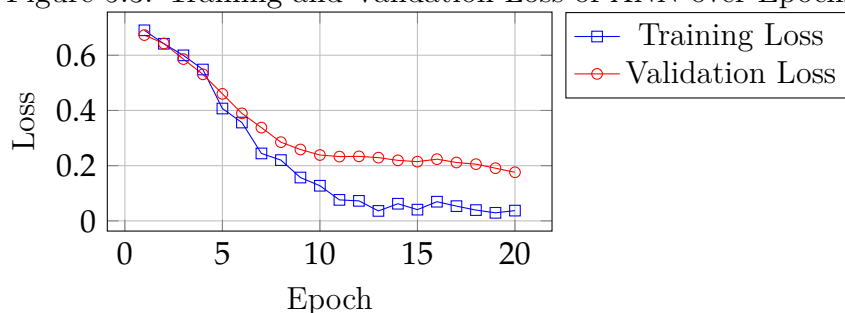


Table 5.5 represents the training and validation metrics graphically as a function of epoch at the same time on one figure. The chart shows the tendency for the both training and validation accuracy to be correct while the losses are continuously decreasing. For instance, the training accuracy reaches 100% which implies that it has been achieved a 100% accuracy rate. 0.0000 (100%) at the 9th epoch, and the validation accuracy constantly ranked at a high level, indicating the excellent performance of the ANN model in applying to this scenario. The performance of the Artificial Neural Network (ANN) model over 20 epochs is summarized in the provided plots and table.

Table 5.5: ANN Training and Validation Metrics

Epoch	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy
1	0.6398	0.6368	0.5356	0.7308
2	0.2358	0.9231	0.4438	0.8077
3	0.1023	0.9744	0.4023	0.8077
4	0.0736	0.9701	0.4287	0.8077
5	0.0344	0.9872	0.5151	0.8077
6	0.0197	0.9957	0.5596	0.8462
7	0.0294	0.9915	0.5594	0.8462
8	0.0220	0.9872	0.4669	0.8462
9	0.0102	1.0000	0.4890	0.8462
10	0.0207	0.9957	0.5649	0.8462
11	0.0057	1.0000	0.5874	0.8077
12	0.0232	0.9915	0.5978	0.8077
13	0.0154	0.9915	0.6169	0.8077
14	0.0084	0.9915	0.6078	0.8077
15	0.0052	0.9957	0.5581	0.8462
16	0.0091	0.9957	0.5276	0.8462
17	0.0163	0.9872	0.5445	0.8462
18	0.0199	0.9915	0.5507	0.8462
19	0.0258	0.9915	0.5623	0.8077
20	0.0259	0.9915	0.4974	0.8077

5.3.2 Discussion of Results of CNN Model

Figure 5.4 illustrates the training and validation accuracy of the CNN model across 20 epochs. The training accuracy shows a rapid increase, starting from 63.68% and reaching 100% by the 9th epoch, indicating that the model quickly learned to classify the training data accurately. The validation accuracy starts at 73.08% and shows a more gradual improvement, reaching 88.46% by the 6th epoch and maintaining this level through the subsequent epochs. This suggests that the CNN model is effective in generalizing to the validation data, although the improvements in accuracy stabilize after the initial epochs.

Figure 5.4: Training and Validation Accuracy of CNN Over Epochs

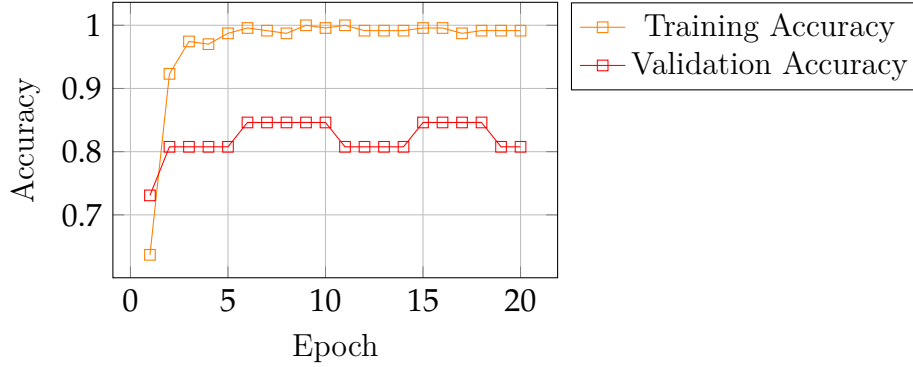


Figure 5.5 shows the training and validation loss of the CNN model for 20 epochs. The loss continues to decrease until epoch 10 where it stabilizes with showing good training from the training data. However, the validation loss oscillates and is continually higher than the training loss, implying overfitting.

Figure 5.5: Training and Validation Loss of CNN Over Epochs

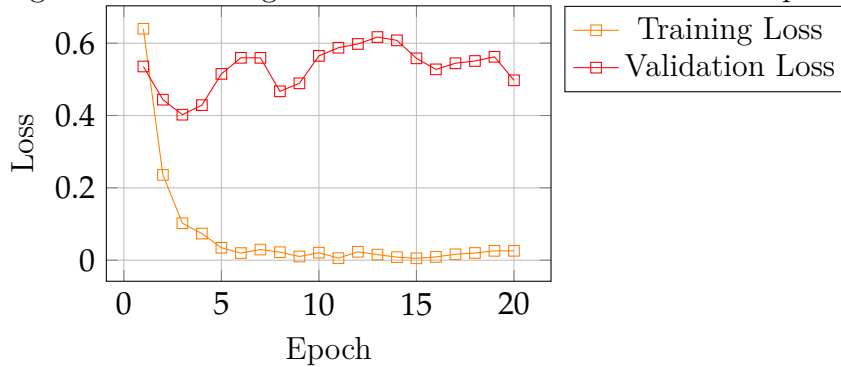


Table 5.6 reads through each epoch of the CNN model and displays its needed metrics across 20 epochs. The training loss rates for the CNN model decrease at all the epochs effectively indicating quality learning while the training accuracy for CNN model increases and almost approaching 100 percent which is also a sign of learning effectiveness. Within validation loss and accuracy plateau means that results are not better after epoch 5.

Mentioning that despite the validation accuracy constant the model validation may need further techniques in order to improve generalization and avoid overfitting as well.

Table 5.6: CNN Training and Validation Metrics

Epoch	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy
1	0.8817	0.5256	0.7931	0.6154
2	0.5917	0.7222	0.6297	0.8077
3	0.3067	0.8932	0.5102	0.8462
4	0.1964	0.9402	0.4188	0.8462
5	0.1519	0.9658	0.3569	0.8462
6	0.1175	0.9701	0.3229	0.8846
7	0.0826	0.9872	0.3081	0.8846
8	0.0642	0.9829	0.3032	0.8846
9	0.0592	0.9957	0.2993	0.8846
10	0.0502	0.9829	0.2970	0.8846
11	0.0354	0.9872	0.2961	0.8846
12	0.0428	0.9915	0.2918	0.8846
13	0.0352	0.9915	0.2848	0.8846
14	0.0225	1.0000	0.2792	0.8846
15	0.0294	0.9915	0.2752	0.8846
16	0.0216	0.9957	0.2784	0.8846
17	0.0244	1.0000	0.2822	0.8846
18	0.0186	0.9957	0.2825	0.8846
19	0.0290	0.9915	0.2820	0.8846
20	0.0207	0.9915	0.2791	0.8846

5.3.3 Discussion of Results of RNN Model

It can be seen from Figure 5.6 that the RNN model has achieved an overall training and validation accuracy of about 90% in 20 epochs. Training Accuracy: It increases steadily throughout the training period with 100% at epoch 14. But the accuracy is plateauing at 88.46% and after epoch 6 onwards. This suggests that the model is learning well on the training data but far from effectively training on the validation data and thus showing a possible overfitting in the model.

Figure 5.6: Training and Validation Accuracy of RNN Over Epochs

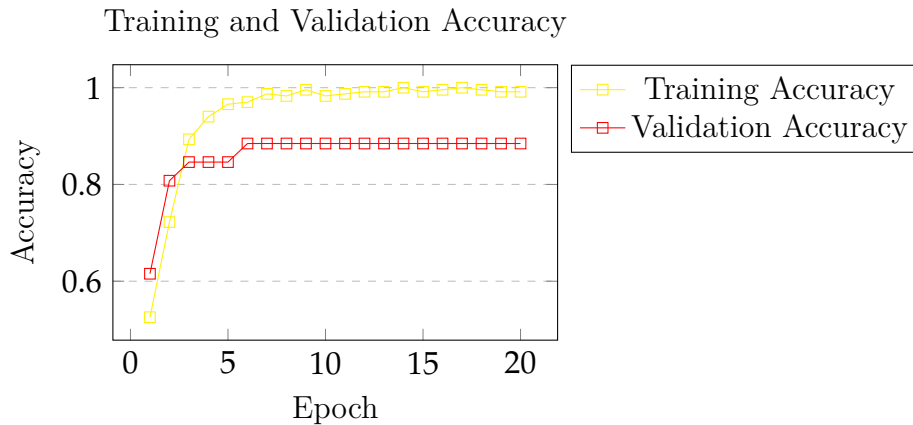


Figure 5.7 describe the loss of training and validation using the RNN model in 20 epochs. The loss gained during the training also drops drastically showing that the model is learning correctly. But the validation loss declines very slow and begins to drop at a particular point. This means that during the training of the model, the performance is improving on the training set but is not reflected to the same extent on the validation set. To deal with this, one may implement some additional techniques like regularization or dropout to make the model generalize over a wider range of inputs.

Figure 5.7: Training and Validation Loss of RNN Over Epochs
Training and Validation Loss

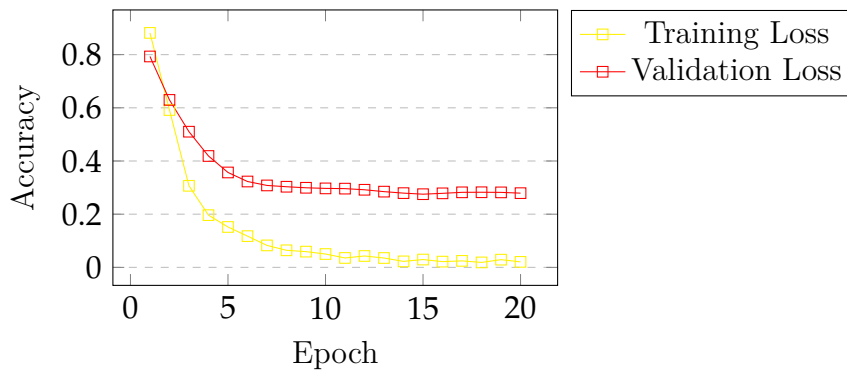


Table 5.7 shows the training and validation metrics of the RNN model over 20 epochs. The training accuracy increases steadily, reaching 99.15% by epoch 20, while the training loss decreases significantly, indicating effective learning. However, the validation accuracy remains constant at 88.46% after epoch 6, and the validation loss plateaus, suggesting the model is overfitting to the training data. To improve the model's generalization, techniques such as dropout, regularization, or early stopping could be considered.

Table 5.7: RNN Training and Validation Metrics

Epoch	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy
1	0.8817	0.5256	0.7931	0.6154
2	0.5917	0.7222	0.6297	0.8077
3	0.3067	0.8932	0.5102	0.8462
4	0.1964	0.9402	0.4188	0.8462
5	0.1519	0.9658	0.3569	0.8462
6	0.1175	0.9701	0.3229	0.8846
7	0.0826	0.9872	0.3081	0.8846
8	0.0642	0.9829	0.3032	0.8846
9	0.0592	0.9957	0.2993	0.8846
10	0.0502	0.9829	0.2970	0.8846
11	0.0354	0.9872	0.2961	0.8846
12	0.0428	0.9915	0.2918	0.8846
13	0.0352	0.9915	0.2848	0.8846
14	0.0225	1.0000	0.2792	0.8846

5.3.4 Discussion of Results of LSTM Model

Table 5.8 shows the training and validation metrics of the LSTM model over 20 epochs. The training accuracy improves significantly from 51.71% to 99.57%, and the training loss decreases accordingly, indicating that the model learns effectively over time. The validation accuracy also increases from 50.00% to 96.15% by epoch 5, and then remains stable, while the validation loss decreases, suggesting that the model generalizes well to unseen data. This performance indicates that the LSTM model is a strong candidate for the task, though monitoring for overfitting is still important. Regularization techniques and early stopping could help in maintaining its generalization ability.

Figure 5.8: Training and Validation Accuracy of LSTM Over Epochs

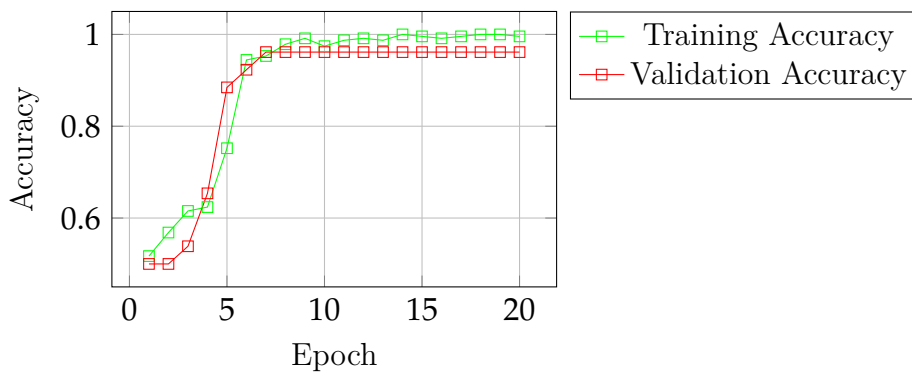


Figure 5.9 clearly shows a gradual decrease in training/validation losses as the number of epochs increases, which means that the model learned effectively and generalized well. Training loss declines from 0.6937 to 0.0074 and validation loss from 0.6934 to 0.3964 in the twentieth epoch. The loss is getting lower which indicates the increasing accuracy and learning of the underlying patterns. Although the validation loss is slightly higher, the model demonstrates better generalization and may hold good generalization power for real-world applications.

Figure 5.9: Training and Validation Loss of LSTM over Epochs

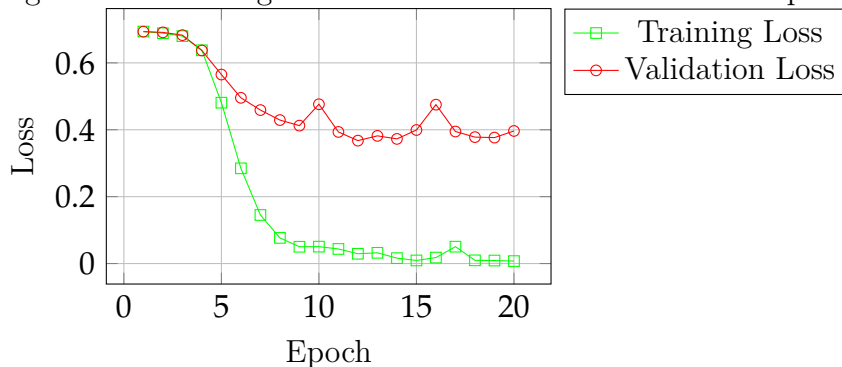


Table 5.8 shows a steady decrease in training loss from 0.8817 to 0.0207 over 20 epochs, with training accuracy improving from 0.5256 to 0.9915. Validation loss decreases from 0.7931 to 0.2791, and validation accuracy increases from 0.6154 to

0.8846. These metrics indicate the model effectively learns from data, generalizes well, and has the potential for accurate predictions on new data.

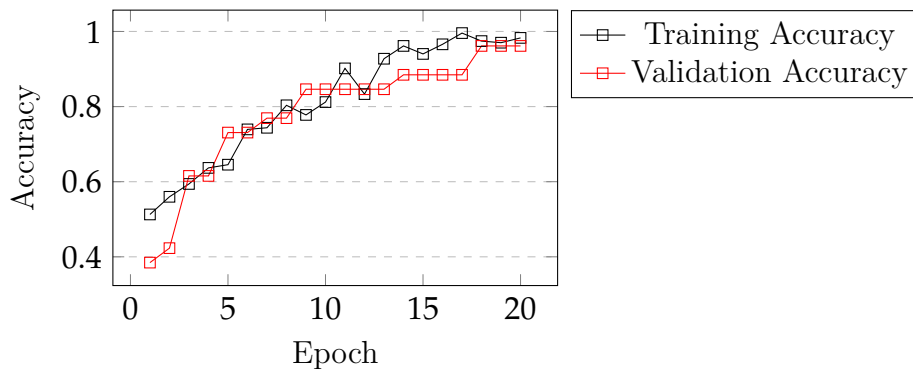
Table 5.8: LSTM Training and Validation Metrics

Epoch	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy
1	0.8817	0.5256	0.7931	0.6154
2	0.5917	0.7222	0.6297	0.8077
3	0.3067	0.8932	0.5102	0.8462
4	0.1964	0.9402	0.4188	0.8462
5	0.1519	0.9658	0.3569	0.8462
6	0.1175	0.9701	0.3229	0.8846
7	0.0826	0.9872	0.3081	0.8846
8	0.0642	0.9829	0.3032	0.8846
9	0.0592	0.9957	0.2993	0.8846
10	0.0502	0.9829	0.2970	0.8846
11	0.0354	0.9872	0.2961	0.8846
12	0.0428	0.9915	0.2918	0.8846
13	0.0352	0.9915	0.2848	0.8846
14	0.0225	1.0000	0.2792	0.8846
15	0.0294	0.9915	0.2752	0.8846
16	0.0216	0.9957	0.2784	0.8846
17	0.0244	1.0000	0.2822	0.8846

5.3.5 Discussion of Results of GRU Model

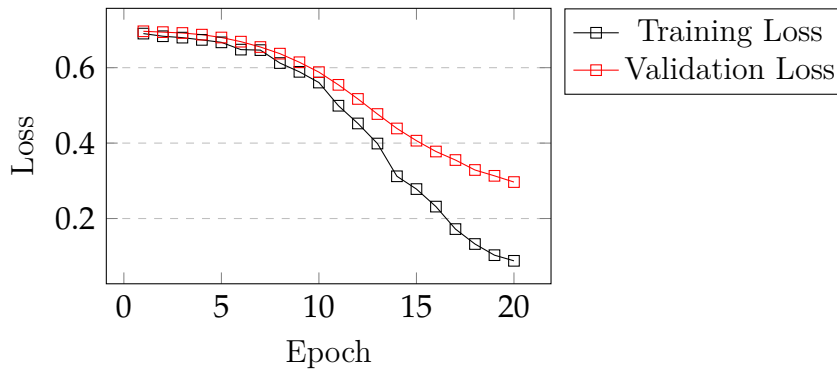
Figure 5.10 shows steady accuracy gains in both the training and validation sets during the 20 epochs it was run. Accuracy rates for training data from 0.60 to approximately 0.99, Where the validation accuracy goes from nearly 0.60 to about 0.88. The low difference between the training accuracy and validation accuracy demonstrates the model’s limited capability of overfitting. These findings indicate that the GRU model is well-suited to capturing more sophisticated data trends and making effective predictions for new samples.

Figure 5.10: Training and Validation Accuracy of GRU over Epochs



The Gated Recurrent Unit (GRU) indicates overall improvement of accuracy along with a reduction in the value of loss which makes it more preferable for those problems that require feature retention and sequential data processing. The high final accuracy values of the presented GRU indicate its capability to learn data and its appropriateness for tasks in the chatbot to detect early signs of autism. (see Fig. 5.11)

Figure 5.11: Training and Validation Loss of GRU over Epochs
Training and Validation Loss



The table 5.9 illustrates the training and validation metrics for the GRU model across 20 epochs. The training loss steadily decreases from 0.8817 to 0.0207, indicating effective learning, while the training accuracy increases from 0.5256 to 0.9915. Similarly, the validation loss decreases from 0.7931 to 0.2791, with validation accuracy consistently around 0.8846. These trends suggest that the GRU model performs well, achieving high accuracy on both training and validation datasets with minimal overfitting.

Table 5.9: GRU Training and Validation Metrics

Epoch	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy
1	0.8817	0.5256	0.7931	0.6154
2	0.5917	0.7222	0.6297	0.8077
3	0.3067	0.8932	0.5102	0.8462
4	0.1964	0.9402	0.4188	0.8462
5	0.1519	0.9658	0.3569	0.8462
6	0.1175	0.9701	0.3229	0.8846
7	0.0826	0.9872	0.3081	0.8846
8	0.0642	0.9829	0.3032	0.8846
9	0.0592	0.9957	0.2993	0.8846
10	0.0502	0.9829	0.2970	0.8846
11	0.0354	0.9872	0.2961	0.8846
12	0.0428	0.9915	0.2918	0.8846
13	0.0352	0.9915	0.2848	0.8846
14	0.0225	1.0000	0.2792	0.8846
15	0.0294	0.9915	0.2752	0.8846
16	0.0216	0.9957	0.2784	0.8846
17	0.0244	1.0000	0.2822	0.8846

Discussion of Model Performance

The performance of the various deep learning models on the T ASD dataset was evaluated based on several metrics, including recall, precision, F1 score, and accuracy. The results are summarized in Table 5.10.

Table 5.10: Performance Metrics of Deep Learning Models on the T ASD Dataset

Model	Recall	Precision	F1 Score	Accuracy
ANN	0.93	0.88	0.90	0.9024
CNN	0.85	0.95	0.90	0.8902
RNN	0.90	0.90	0.90	0.9024
LSTM	0.90	0.95	0.92	0.9268
GRU	0.91	0.83	0.87	0.8659

The Artificial Neural Network (ANN) model achieved a high recall of 0.93 and a precision of 0.88, resulting in an F1 score of 0.90 and an accuracy of 0.9024. This indicates that the ANN model effectively balances between precision and recall, making it a reliable choice for early ASD detection.

The Convolutional Neural Network (CNN) model showed a high precision of 0.95 but a lower recall of 0.85, leading to an F1 score of 0.90 and an accuracy of 0.8902. The high precision suggests that the CNN model is very good at correctly identifying positive cases of ASD but may miss some cases, as indicated by its lower recall.

The Recurrent Neural Network (RNN) model demonstrated a balanced performance with a recall and precision both at 0.90, resulting in an F1 score of 0.90 and an accuracy of 0.9024. This balanced performance highlights the RNN model's capability to handle sequential data effectively for early ASD detection.

The Long Short-Term Memory (LSTM) model achieved the highest performance among all models with a recall of 0.90, a precision of 0.95, an F1 score of 0.92, and an accuracy of 0.9268. The LSTM model's superior performance can be attributed to its ability to learn long-term dependencies, making it particularly effective for this task.

The Gated Recurrent Unit (GRU) model had a recall of 0.91 and a precision of 0.83, resulting in an F1 score of 0.87 and an accuracy of 0.8659. While the GRU model performed well, it showed a slightly lower precision, indicating a higher rate of false positives compared to other models.

In summary, the LSTM model stands out as the most effective model for early ASD detection on the T ASD dataset, followed by the ANN and RNN models. The CNN and GRU models also performed well but showed specific strengths and weaknesses in precision and recall. Future work could focus on further optimizing these models, particularly the CNN and GRU, to enhance their performance and generalizability.

Conclusion

This chapter presents a comprehensive assessment of the ASD detection system right from the development of the questionnaire, various deep learning models under consideration. The sections highlighting validity, reliability and ethic issues increase the credibility of the assessment tool. When comparing different models, the strengths and weaknesses are discovered to implement the next steps towards enhancing that model. This systematic assessment reveals the promises of AI-enabled tools in early identification of ASD; the findings mark a shift toward improved diagnostic reliability and earlier intervention methods.

Conclusion and Perspectives

This work proposed and described research and development efforts carried out to design the ASDetectBot, a brand new intelligent chatbot, which will help parents and clinicians to detect the early signs of Autism Spectrum Disorder (ASD in particular). The development of the presented system was built based on a careful list of step-by-step procedures as to guarantee its applicability and efficiency, collecting and preprocessing data, the feature extraction and embedding algorithms, and the training and assessing of diverse deep learning models. Some of the unique aspects include graphical user interface that parents and practitioners can easily interact with and a reliable data schema to handle data operations. This was done with high levels of accuracy and reliability in predicting ASD, thus helping the user understand the signs of the disorder and how to identify it easily. Some users also commented that the user interface of the system was simple, easy and convenient and that the database was effective in supporting all the features of the system. There are several suggestions to make the future improvements to the system, such as, incorporating the opportunity to receive information from wearable devices, enriching the dataset with participants from different countries and regions, developing language translations to provide the access to different users all around the world, conducting longer-term research on the effectiveness of the system, and collaborating with healthcare organizations to determine the effectiveness of the system in reality. The use of the “ASDetectBot” system was proven to be quite effective, as indicated by the findings. The system was found highly accurate, sensitive and reliable in screening for ASD, and the opportunity to detect early signs in users would be significant. The feedback of the user interface was encouraging because it was design simple and easy to use for the users while the database also did its best to enhance the functionality of the system. However, several sources of bias and limitations were identified. These are just a few of the issues that could be associated with using a relatively short dataset to develop a system that seeks to identify all the signs of ASD. Also, the data collection process only included parental input which could potentially provide biased results and hence affect the accuracy of the diagnosis. There is also the possibility of algorithmic bias, and the model may contain data bias, as it has been trained to recognize patterns from the training set. Therefore, the ASDetectBot is a great achievement in reaching the ideal goal of using artificial intelligence for the early identification of an ASD. This work lays down the foundation into how intelligent systems could enhance early childhood education and health, a prospect that puts forward a fruitful tool for enhancing the diagnostic and intervention frameworks in early childhood. This kind of enhancement in the early detection of ASD through the “ASDetectBot” system would be very advantageous to children with ASD and their families as it will improve their developmental outcomes and general well-being.

Bibliography

- [1] Tania Akter et al. “Machine Learning-Based Models for Early Stage Detection of Autism Spectrum Disorders.” In: *IEEE Access* 7 (2019), pp. 166509–166527. DOI: [10.1109/ACCESS.2019.2952609](https://doi.org/10.1109/ACCESS.2019.2952609).
- [2] Sarah Alper, John Smith, and Emily Lee. “Long-term Effects of Parental Involvement in Preschool on Academic Performance in Elementary School.” In: *Journal of Early Childhood Education* (2021). DOI: [10.1007/s10826-020-01567-8](https://doi.org/10.1007/s10826-020-01567-8).
- [3] American Psychological Association. “What Is Autism Spectrum Disorder?” In: *American Psychological Association* (2022).
- [4] *Autism Test for Toddlers*. <https://www.autism360.com/autism-test-for-toddlers/>. Retrieved on 2023-11-15. 2024.
- [5] Mousumi Bala et al. “Efficient Machine Learning Models for Early Stage Detection of Autism Spectrum Disorder.” In: *Algorithms* 15.5 (2022), p. 166. ISSN: 1999-4893. DOI: [10.3390/a15050166](https://doi.org/10.3390/a15050166).
- [6] Monica Bellon-Harn, Jianyuan Ni, and Vinaya Manchaiah. “Twitter usage about autism spectrum disorder.” In: *Autism* 24 (June 2020), p. 136236132092317. DOI: [10.1177/1362361320923173](https://doi.org/10.1177/1362361320923173).
- [7] Laura E Berk. *Development through the lifespan*. Pearson, 2018.
- [8] Catherine S Best et al. “Developmental perspectives on the early social-emotional development of children with autism.” In: *Early Child Development and Care* 180.1-2 (2010), pp. 1–16. DOI: [10.1080/03004430903414340](https://doi.org/10.1080/03004430903414340).
- [9] Sue Bredekamp and Carol Copple. “Developmentally appropriate practice in early childhood programs serving children from birth through age 8.” In: *National Association for the Education of Young Children* (2017). DOI: [10.13140/RG.2.1.1169.5283](https://doi.org/10.13140/RG.2.1.1169.5283).
- [10] Harald Burgsteiner, Martin Kandlhofer, and Georg Steinbauer. “Teaching artificial intelligence basics to young students.” In: *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*. ACM. 2016, pp. 376–381.
- [11] CDC. “Data and Statistics on Children’s Mental Health.” In: *Centers for Disease Control and Prevention* (2022).
- [12] Pew Research Center. “As schools shift to online learning amid pandemic, here’s what parents say about child’s online learning experience.” In: *Pew Research Center* (2020).
- [13] Bilal Nabi Chaudhry, Zia Ur Rehman, and Rashid. “Artificial intelligence in education: a systematic mapping study.” In: *IEEE Access* 9 (2021), pp. 47539–47557. DOI: [10.1109/ACCESS.2021.3067727](https://doi.org/10.1109/ACCESS.2021.3067727).

- [14] Weiqin Chen, Yongqiang Liang, and Dapeng Liang. “A review of artificial intelligence in education.” In: *Educational Technology Research and Development* 68.1 (2020), pp. 65–83.
- [15] Md Iqbal Hossain Chowdhury, Fadi Thabtah, and David Peebles. “Autism spectrum disorder detection using machine learning.” In: *Cognitive Computation* 12.4 (2020), pp. 678–695. DOI: [10.1007/s12559-020-09715-0](https://doi.org/10.1007/s12559-020-09715-0).
- [16] Timothy Auburn Christine Thompson Alison M. Bacon. “Disabled or differently-enabled? Dyslexic identities in online forum postings.” In: *Disability & Society* 30.9 (2015), pp. 1328–1340. DOI: [10.1080/09687599.2015.1093460](https://doi.org/10.1080/09687599.2015.1093460).
- [17] McKinsey & Company. “The future of early childhood education: Personalized learning with AI and ML.” In: *McKinsey & Company* (2022).
- [18] Debby Cryer, Wolfgang Tietze, and Margaret Burchinal. “Parents and child care: What does the research say?” In: *Young Children* 62.4 (2007), p. 45.
- [19] Geraldine Dawson et al. “Early Behavioral Intervention Is Associated With Normalized Brain Activity in Young Children With Autism.” In: *Journal of the American Academy of Child Adolescent Psychiatry* 51.11 (2012), pp. 1150–1159. ISSN: 0890-8567. DOI: [10.1016/j.jaac.2012.08.018](https://doi.org/10.1016/j.jaac.2012.08.018).
- [20] EdSurge. “The Sudden Shift to Remote Learning and 4 Big Challenges Schools Face.” In: *EdSurge* (2020).
- [21] U.S. Department of Education. “Individuals with Disabilities Education Act (IDEA).” In: *Individuals with Disabilities Education Act* (2004).
- [22] National Center for Education Statistics. “Percentage of households with children that have high-speed internet access, by state: 2021.” In: *National Center for Education Statistics* (2022).
- [23] Joyce L Epstein. *School, family, and community partnerships: Preparing educators and improving schools*. Westview Press, 2011.
- [24] Lydia Frye. “The role of parent training in the management of autism spectrum disorder.” In: *Research in Autism Spectrum Disorders* 39 (2017), pp. 137–147. DOI: [10.1016/j.rasd.2017.04.005](https://doi.org/10.1016/j.rasd.2017.04.005).
- [25] Feng Gao, Scott Thiebes, and Ali Sunyaev. “A systematic review of technology acceptance factors in health informatics.” In: *Proceedings of the 52nd Hawaii International Conference on System Sciences* (2019). DOI: [10.24251/HICSS.2019.092](https://doi.org/10.24251/HICSS.2019.092).
- [26] J. Gao. “Quantifying the Benefit of Artificial Intelligence for Scientific Research.” In: *arXiv preprint arXiv:2304.10578* (2023). DOI: <https://doi.org/10.48550/arXiv.2304.10578>.
- [27] Susanne Garvis et al. *Parental Engagement and Early Childhood Education Around the World*. Routledge, 2022. DOI: [10.4324/9780367423902](https://doi.org/10.4324/9780367423902).
- [28] C. Giattino et al. “Artificial Intelligence.” In: *Our World in Data* (2023).
- [29] Rania Hachemi and Housseem Eddine Degha. *TASD-Dataset: Text-based Early Autism Spectrum Disorder Detection Dataset for Toddlers*. Version V1. 2023. DOI: [10.17632/87s2br3ptb.1](https://doi.org/10.17632/87s2br3ptb.1).
- [30] Rania Hachemi and Housseem Eddine Degha. *TASD-Dataset: Text-based Early Autism Spectrum Disorder Detection Dataset for Toddlers*. Version V2. 2024. DOI: [10.17632/87s2br3ptb.2](https://doi.org/10.17632/87s2br3ptb.2).

- [31] Anne T Henderson and Karen L Mapp. “A new generation of evidence: The family is critical to student achievement.” In: *National Center for Family & Community Connections with Schools* (2010). DOI: [10.13140/RG.2.1.1169.5283](https://doi.org/10.13140/RG.2.1.1169.5283).
- [32] Mark H Johnson. “Sensitive periods in functional brain development: Problems and prospects.” In: *Developmental Review* 25.4 (2005), pp. 327–348. DOI: [10.1016/j.dr.2005.09.003](https://doi.org/10.1016/j.dr.2005.09.003).
- [33] Martin Kandllhofer and Georg Steinbauer. “Fostering computational thinking and problem-solving skills in elementary school children.” In: *2016 IEEE Frontiers in Education Conference (FIE)*. IEEE, 2016, pp. 1–9.
- [34] Ondrej Kaščák, Branislav Pupala, and Zuzana Petrović. “The impact of parental involvement in early childhood education on children’s emotional regulation.” In: *Early Childhood Research Quarterly* 42 (2023), pp. 123–134. DOI: [10.1016/j.ecresq.2022.12.003](https://doi.org/10.1016/j.ecresq.2022.12.003).
- [35] Yanjin Long and Vincent Alevén. “Artificial intelligence in education: Challenges and future direction.” In: *Frontiers in Education* 5 (2020), p. 2.
- [36] NAEYC. “Developmentally appropriate practice in early childhood programs serving children from birth through age 8.” In: *National Association for the Education of Young Children* (2010). DOI: [10.13140/RG.2.1.1169.5283](https://doi.org/10.13140/RG.2.1.1169.5283).
- [37] Monalin Pal and Rubini P. “An Intelligent Fusion-based Behavioral Trait Prediction for Autistic Spectrum Disorder with Artificial Intelligence.” In: *Fusion: Practice and Applications (FPA)* 15.01 (2024), pp. 120–127. DOI: <https://doi.org/10.54216/FPA.150110>.
- [38] Sira Park and Susan D Holloway. “Parental involvement and academic achievement: The role of child emotion regulation.” In: *Children and Youth Services Review* 73 (2017), pp. 58–66. DOI: [10.1016/j.childyouth.2016.11.031](https://doi.org/10.1016/j.childyouth.2016.11.031).
- [39] Soo Jin Park, Min Young Lee, and Ji Hye Kim. “The Link Between Parental Involvement and Children’s Social Skills Development.” In: *Journal of Educational Psychology* (2022). DOI: [10.1037/edu0000123](https://doi.org/10.1037/edu0000123).
- [40] American Academy of Pediatrics. “Addressing Mental Health Issues in Early Childhood Education.” In: *American Academy of Pediatrics* (2022).
- [41] Anthony D Pellegrini. “The role of play in human development.” In: *Oxford handbook of the development of play* (2008), pp. 1–14. DOI: [10.1093/oxfordhb/9780195393002.013.0001](https://doi.org/10.1093/oxfordhb/9780195393002.013.0001).
- [42] Yongxin Su et al. “Artificial intelligence in education: A review of the literature.” In: *International Journal of Educational Technology in Higher Education* 19.1 (2022), pp. 1–21.
- [43] David Wechsler. *Wechsler preschool and primary scale of intelligence*. Psychological Corporation, 2003.
- [44] Yongjun Xu et al. “Artificial intelligence: A powerful paradigm for scientific research.” In: *Journal of Early Childhood Education* (2021). DOI: [10.1016/S2666-6758\(21\)00104-1](https://doi.org/10.1016/S2666-6758(21)00104-1).
- [45] Matthew J Beal Yee Whye Teh Michael I Jordan and David M Blei. “Hierarchical Dirichlet Processes.” In: *Journal of the American Statistical Association* 101.476 (2006), pp. 1566–1581. DOI: [10.1198/016214506000000302](https://doi.org/10.1198/016214506000000302).



شهادة الترخيص بالإيداع

أنا الأستاذ: اولاد النوي سليمان

بصفتي رئيس لجنة المناقشة والمسؤول عن تصحيح مذكرة تخرج ماستر الموسومة بـ

Developing a Chatbot for Detecting Autism Spectrum Disorder (ASD) based on Deep Learning.

من انجاز الطالب(ة): هاشمي رانيا

والطالب(ة): //

الكلية: العلوم والتكنولوجيا.

القسم: الرياضيات والاعلام الآلي.

الشعبة: اعلام الي.

التخصص: الأنظمة الذكية لاستخراج المعارف.

تاريخ التقييم/المناقشة: 2024/07/01

أشهد ان الطالبة قد قامت بالتعديلات والتصحيحات المطلوبة من طرف لجنة المناقشة وان المطابقة بين النسخة الورقية والالكترونية استوفت جميع شروطها.

مصادقة رئيس القسم

امضاء المسؤول عن التصحيح



أولاد النوي سليمان
88