# FINAL YEAR PROJECT

# DEVELOPMENT OF A SPEECH THERAPY APPLICATION USING MACHINE LEARNING

# BY

# ODILE EMMANUEL – RUN/CMP/21/10891

# OBAFEMI OPEYEMI – RUN/CMP/21/10889

# GASPER SAMUEL – RUN/CMP/21/10869

# AJIBADE GOODNESS – RUN/CMP/22/13036

# 

# CHAPTER ONE

# INTRODUCTION

## 1.1 Background to Study

Speech disorders significantly hinder effective communication, impacting individuals' quality of life. These disorders encompass a range of conditions, including Dysarthria, Aphasia, and Stuttering, each presenting unique challenges that affect verbal expression and comprehension. According to the World Health Organization [WHO] in 2024, approximately 1.3 billion people worldwide live with disabilities, with a notable portion requiring assistive technology (AT) for communication. The United Nations [UN] Convention on the Rights of Persons with Disabilities recognizes the provision of AT as a fundamental human right, underscoring the urgent need for innovative solutions to assist those with speech impairments.

Machine Learning (ML), a subset of artificial intelligence, has emerged as a transformative force in various fields, including healthcare and speech therapy. ML algorithms can analyze vast datasets to identify patterns and make predictions, thus enhancing the accuracy and accessibility of speech recognition technologies. Recent studies highlight the effectiveness of ML in developing applications that assist individuals with speech disorders by improving communication tools such as speech-to-text and text-to-speech systems. These advancements facilitate better interaction and empower users by personalizing their experience based on individual speech patterns (Olawade, et al., 2024).

Despite the advancements in ML applications for speech disorders, existing research often focuses on specific types of disorders or age groups, limiting the breadth of research outcomes. Many have centered solely on conditions like aphasia or Dysarthria, neglecting other significant disorders or broader demographic considerations. Therefore, this presents an opportunity to explore inclusive ML solutions that address other speech disorders.

This study aims to develop a comprehensive speech therapy application utilizing machine learning techniques to assist individuals with speech disorders, specifically stuttering.

## 1.2 Statement of Problem

According to statistics from the National Institute on Deafness and Other Communication Disorders (NIDCD), speech disorders, including stuttering and articulation issues, affect approximately 5% of children aged 3 to 17. A higher prevalence of 8–9% is observed in younger children. Persistent developmental stuttering explicitly impacts about 1% of Americans, with boys aged 2 to 6 more frequently affected. In South Africa, the 2011 Census reported that 6% of the population has a speech and language disorder (Ngyende, 2011).

In Nigeria, a study focused on individuals with speech and language difficulties. It examined 146 children and adults referred for speech and language therapy at the University of Ilorin Teaching Hospital in North Central Nigeria. The findings revealed that over half (58%) of the patients were diagnosed with deaf-mutism, characterized by an inability to speak. Additionally, 21% were identified with delayed speech development. Other diagnoses included slurred speech pathology (4%), impaired speech (13%), stammering or stuttering (3%), and aphasia (1%) (Shuaib, Afolabi, Alabi & Elemunkan 2011).

Although speech therapy is widely available, traditional approaches often require lengthy, resource-intensive sessions that are not accessible to everyone. Speech therapy can be time-consuming, especially when phono-articulatory dynamics are complex. In most cases, the phono-articulatory model is provided by speech and language pathologists (for pronunciation exercises conducted in the practice room) or by significant individuals such as parents (for exercises performed at home). Constant monotony of exercises and limited opportunities for individual practice reduce children's motivation for recovery and weaken their focus on therapy goals. Intelligent diagnosis and therapy systems are now seen as a viable alternative to increase the efficiency of speech and language therapy (Popovici & Cristian, 2012).

## 1.3 Aim and Objectives

The aim is to develop a speech therapy system for stuttering using machine learning. The specific objectives are to:

1. Collect and preprocess the SEP-28k dataset obtained from the GitHub repository hosted by Apple.
2. Train and test the selected machine learning model to design a personalized therapy application.
3. Evaluate the performance of the algorithms. IVIV. Develop a stuttering speech therapy system based on the best algorithm.

## 1.4 Scope of the Study

Stuttering was chosen for this study because it is a widespread speech disorder that affects communication and quality of life. Existing speech therapy methods can be expensive and inaccessible, and this project aims to provide an innovative, data-driven, and personalized solution for managing stuttering effectively.

Using audio samples from the SEP-28k dataset obtained from the GitHub repository hosted by Apple, the system will identify disfluencies such as repetitions, blocks, and prolongation, which are the most common characteristics of stuttering. Detecting these patterns helps assess stuttering severity and tailor effective therapy interventions.

This study is limited to British English speech due to the availability of datasets. It does not cover other speech disorders like dysarthria or aphasia.

## 1.5 Significance of the Study

The importance of this research stems from its potential to make a meaningful difference in the lives of people who stutter or face other speech-related challenges.

Speech disorders can profoundly impact a person's ability to communicate, learn, and connect with others, often leading to feelings of frustration, isolation, and even low self-confidence. This project focuses on developing a speech therapy application powered by machine learning (ML) that offers tailored therapy plans and real-time feedback, helping users build their communication skills in a supportive and personalized way. Doing so aims to make speech therapy more accessible, effective, and empowering for those who need it.

## 1.6 Definition of Terms

**Artificial Intelligence (AI):** The simulation of human intelligence by machines, especially computer systems. AI enables systems to perform tasks that typically require human intelligence, such as understanding language, recognizing patterns, solving problems, and making decisions.

**Automatic Speech Therapy:** The use of technology, such as software, apps, or AI-powered tools, to assist individuals in improving their speech and communication skills.

**Machine Learning:** The capacity of systems to learn from problem-specific training data to automate the process of building an analytical model and solving associated tasks.

**Speech:** The use of the human voice as a medium for language. Spoken language combines vowel and consonant sounds to form units of meaning like words, which belong to a language's lexicon.

**Speech Detection:** Also known as Voice Activity Detection (VAD), this is the process of identifying the presence or absence of speech and distinguishing it from non-speech sections. It plays a crucial role in applications like speech coding and recognition, using various principles such as energy or spectral changes to achieve accurate detection.

**Speech Disfluency:** This disrupts the flow of speech, such as repeating words, syllables, or sounds. It can also include prolonging sounds or blocking sounds.

**Speech Disorder:** Affects a person’s ability to produce or understand speech, causing problems with communication. Examples include stuttering, articulation difficulties, and aphasia.

**Speech Identification:** The task of determining an unknown speaker's identity.

**Speech Recognition:** Also known as speech-to-text, a machine or program can identify spoken words and convert them into readable text.

**Stuttering:** A speech disorder where the flow of speech is interrupted by repetitions, prolonged sounds, or blocks, making it harder to speak fluently.

**Therapy:** The attempted remediation of a health problem, usually following a medical diagnosis

## 1.7 Organization of the project

This project will be structured into five different chapters. Chapter one focuses on the background to the study, highlighting the specific objectives, scope, and significance. Chapter two discusses Artificial Intelligence, Machine Learning, the history, and applications. It discusses speech disorders, traditional and online speech therapy solutions, and their advantages and limitations. It then includes a review of some related works done.

# CHAPTER TWO

# LITERATURE REVIEW

## 2.0 Introduction

This chapter discusses Artificial Intelligence and Machine Learning, their history, types, and applications. It further explains Deep learning algorithms, Transformer Architecture, its benefits, future, and limitations. It then briefly discusses Stuttering Speech Disorder and how machine learning can be applied in its treatment. There is also a review of some related works.

## 2.1 Artificial Intelligence

Artificial Intelligence (AI) studies how the human brain thinks, learns, makes decisions, and solves problems. Artificial intelligence strives to improve computer functions linked to human knowledge, such as thinking, learning, problem-solving, belief, and language intelligence (Mahato, 2022). It is a replica of how human intelligence works, but it does not deliver the biologically observable methods.

The development process of AI includes perceptual intelligence, cognitive intelligence, and decision-making intelligence. Perceptual intelligence means that a machine has the basic abilities of vision, hearing, touch, etc., familiar to humans. Cognitive intelligence is a higher-level ability to induce, reason, and acquire knowledge. It is inspired by cognitive science, brain science, and brain-like intelligence to endow machines with thinking logic and cognitive ability like humans. Once a machine has the abilities of perception and cognition, it is often expected to make optimal decisions as human beings, to improve people's lives, industrial manufacturing, etc. Decision intelligence requires applying data science, social science, decision theory, and managerial science to expand data science and to make optimal decisions (Xu, Qi, & Zhulin, 2021).

### 2.1.1 History of Artificial Intelligence

According to the book by Sheikh, Prins, and Schrijvers (2023), the history of AI is summarized in these:

**1. Mythical Representation of AI**

Myths and stories about what we would now call AI have been around for centuries. The ancient Greeks, in particular, celebrated many characters in their mythology who can be characterized as artificial forms of intelligence. Take Talos, a robot created by the great inventor Daedalus to protect the island of Crete. Every day, Talos would run around the island and throw stones at any approaching ships he spotted. This is a myth about a mechanical super-soldier. A robotic exoskeleton used by the US Army now bears the same name.

Estonia has a legend about the Kratt, a magical creature made of hay and household items that did everything its owner asked. If the Kratt was not kept busy, it endangered its owner. The modern law in Estonia that governs liability for using algorithms is known as the ‘Kratt Law’.

**2. Speculation about thinking machines**

The next phase was heralded by the ‘mechanization of the world’. It is envisaged in the work of thinkers like Galileo Galilei, Isaac Newton, and René Descartes. The construction of all kinds of novel machines accompanied their mechanical worldview. Artificial intelligence was still far beyond the realm of possibility. However, the new devices led to speculation about their creation, speculation that was no longer mythical but mechanical.

In 1642, Blaise Pascal built a mechanical calculator which he said was “closer to thought than anything done by animals”. Gottfried Leibniz constructed an instrument he called the ‘step reckoner’ in 1673, which could be used to perform arithmetical calculations. This laid the foundation for many future computers. The philosophers of the time speculated about such devices using the term ‘automata’.

In 1769, Wolfgang von Kempelen built a highly sophisticated machine – or so people long thought. He gained worldwide fame after offering his mechanical Turk to the Austrian Empress Maria Theresa. The huge device was an automatic chess machine, which toured the Western world for 48 years and defeated opponents like Napoleon Bonaparte and Benjamin Franklin. It was not until the 1820s that it was discovered to be a total fake: a man was inside the machine, moving the pieces. As an aside, Amazon has a platform called Mechanical Turk, where people can arrange to have tasks done cheaply online. While more open than Von Kempen’s original, here too, the work is done by people behind the scenes; we do not see.

An important work of literary science fiction in the context of speculation about AI is R.U.R. by the Czech author Karel Čapek. The writer introduces the term ‘robot’ in this book, derived from the Old Church Slavonic word ‘rabota’, meaning corvée or forced labour. This story also reveals a classic fear of AI; the artificial labourers (‘roboti’) created in a factory rebel against their creators and ultimately destroy humankind. Capek’s book was published in 1920, by which time the next phase, much more concrete thinking about AI, had long since begun.

**3. Theory of AI**

From the second half of the nineteenth century onwards, the idea of AI as ‘thinking computers’ became less fantastical and entered the realm of serious theoretical consideration. This development occurred in parallel with the theorization and construction of the first computers. The development of the computer was given another boost during the war by the British research programme Colossus, which aimed to crack the Nazis’ secret communication system known as Enigma. One of the leading lights in this top-secret project at Bletchley Park was Alan Turing, often regarded as the father of computers and AI. He went on to help develop the first genuinely modern computer in Manchester in 1948. Two years later, in 1950, he wrote a paper proposing a thought experiment as an ‘imitation game’ for a computer pretending to be a human being. This has come to be known as the Turing test. A computer passes if a human cannot establish that a person or a computer provided their written answers to their questions. Variants of this test are still used, for example, to compare AI systems with human abilities such as recognizing images or using language.

John von Neumann continued to develop the basic concept of a computer with components such as the central processor, memory, and input-output devices. Another important founder of AI theory was Norbert Wiener. He coined the term ‘cybernetics’ 1948 to describe “the study of control and communication in animals and machines”. The key idea was that people, animals, and machines could all be understood according to several basic principles. The first is control: all those entities strive to counter entropy and control their environment using the principle of ‘feedback’, which is the “ability to adapt future behavior to experience”. Thanks to such advances, during this period, scientists were ready to stop just dreaming and thinking about AI and start developing the technology and experimenting with it in the laboratory. The starting gun for this race was fired in 1956.

**4. AI in the Lab**

The discipline of artificial intelligence (AI) officially began in 1956 with the Dartmouth Summer Research Project on Artificial Intelligence, organized by John McCarthy and Marvin Minsky, who coined the term "artificial intelligence." This event gathered leading scientists such as Herbert Simon and John Nash, aiming to simulate human intelligence in machines, including language use, abstraction, problem-solving, and self-improvement. Although the proposal was overly optimistic, it set the research agenda for AI, leading to early successes like programs playing draughts and logical problem-solving machines such as the Logic Theory Machine and the General Problem Solver. Despite progress in the lab, practical applications were limited. By the late 1960s, enthusiasm began to wane due to technical challenges like the combinatorial explosion and hardware limitations. This culminated in the first "AI winter" during the 1970s when funding and interest declined sharply.

Two main approaches to AI emerged during this first wave and continue to influence the field: symbolic AI and connectionism. Symbolic AI, or rule-based AI, operates on explicitly programmed logical rules ("IF X, THEN Y") and was dominant early on, exemplified by expert systems and programs like the Logical Theory Machine. Based on artificial neural networks, connectionism simulates brain neurons and learns patterns from data rather than relying on pre-set rules. This approach gained attention with Frank Rosenblatt’s perceptron, which could learn to recognize letters without preprogrammed rules. However, symbolic AI proponents criticized neural networks for their lack of explicit rules. Marvin Minsky’s 1969 book "Perceptrons" highlighted their limitations, contributing to the decline of neural network research during the first AI winter.

The second wave of AI began in the early 1980s, coinciding with the rise of personal computers and renewed interest in AI research globally. Japan’s Fifth-Generation Computer Systems Project invested heavily in Prolog-based logical reasoning systems and parallel computing. In response, the US launched initiatives like the Microelectronics and Computer Technology Corporation (MCC) and DARPA’s Strategic Computing Initiative, while the UK introduced the Alvey Programme. During this period, expert systems became prominent, with programs like MYCIN assisting medical diagnosis and Dendral analyzing chemical molecules. These systems applied symbolic AI to practical domains, including manufacturing, mathematics, and legal knowledge-based systems, with notable contributions from Dutch research communities and organizations like JURIX.

Despite the setbacks of the first AI winter, the second wave saw AI gaining practical footholds and broader institutional support through government and industry collaborations. However, the fundamental divide between symbolic AI and neural network approaches persisted, shaping research directions for decades. While symbolic AI focused on encoding expert knowledge into rules, connectionist methods sought to mimic brain processes through data-driven learning. This duality remains central to AI’s evolution, with developments in both paradigms continuing to drive advances in artificial intelligence today.

## 2.2 Overview of Machine Learning

Machine learning is a subset of artificial intelligence that empowers computers to learn and improve from experience without being explicitly programmed. It involves algorithms that enable systems to analyze data, detect patterns, and make decisions or predictions. Its success relies on training models with diverse and representative information to enhance accuracy and generalization. As technology advances, machine learning continues to shape how we interact with and benefit from intelligent systems (Alam, 2023).

Machine learning is a technique that discovers previously unknown relationships in data by searching potentially massive data sets to discover patterns and trends that go beyond simple statistical analysis. Machine learning uses sophisticated algorithms trained to identify data patterns, creating models. Those models can make predictions and categorize data (Chen, 2024).

### 2.2.1 Brief History of Machine Learning

Machine learning draws inspiration from how brain cells interact, a concept first modeled by Donald Hebb in 1949. Hebb theorized that when one neuron repeatedly helps trigger another, their connection strengthens, a principle summarized as “neurons that fire together, wire together.” This foundational idea was later adapted to artificial neural networks, where the strength of connections called “weights” between artificial neurons is adjusted based on their simultaneous activation. If two nodes activate together, their connection strengthens (positive weight); if not, it weakens (negative weight).

In the 1950s, Arthur Samuel of IBM advanced machine learning by developing a checkers-playing program that used limited memory efficiently through alpha-beta pruning and a scoring function to evaluate board positions. Samuel’s program improved its performance by remembering past positions and outcomes, which he called rote learning. He coined “machine learning” in 1952, highlighting the program’s ability to learn from experience and refine its strategies over time.

Frank Rosenblatt’s invention of the perceptron in 1957 marked another milestone. Combining Hebb’s neural model with Samuel’s learning principles, Rosenblatt created the first successful neuro-computer, the Mark I perceptron, designed for image recognition. However, the perceptron’s limitations in recognizing complex visual patterns led to disappointment and a slowdown in neural network research, which did not regain momentum until the 1990s.

Further progress came with developing multilayer networks in the 1960s and introducing the nearest neighbor algorithm in 1967, which enabled basic pattern recognition and route optimization. The discovery that multiple network layers significantly increased computational power led to innovations like feedforward networks and backpropagation. Backpropagation, developed in the 1970s, allowed networks to adjust hidden layers and learn from errors, laying the groundwork for today’s deep neural networks. Modern artificial neural networks, with their hidden layers, excel at identifying complex patterns beyond human programming, making them essential tools in contemporary machine learning (Foote, 2021).

### 2.2.2 Types of Machine Learning

**1. Supervised Learning**

According to Taye (2023), the learning algorithm for machine learning is trained using labelled data. The data are labelled because they consist of pairs, the desired output that can be defined as a supervisory signal, and the input that can be expressed as a vector. Supervised learning occurs when the correct result is known beforehand. Over time, the learning algorithm refines its predictions of this output to narrow the gap between its predictions and the actual output.

The two primary subcategories of supervised learning are regression algorithms (continuous output) and classification algorithms (discrete output). Regression algorithms find the optimum function to match the training dataset’s points. The three primary categories of regression algorithms are linear regression, multiple linear regression, and polynomial regression. Assigning each input to the appropriate class allows classification algorithms to determine which class best fits the supplied data. In this instance, the output of the predictive function is discrete, and its value belongs to one of the possible classes. Regression solves regression issues, while SVMs are used to classify. Random forest is used to solve classification and regression issues.

Machine learning algorithms are frequently employed to anticipate prices in industries such as sales, commerce, and the stock market. These sectors rely heavily on future projections, and by employing supervised machine learning algorithms, more accurate forecasts may be created.

**2. Unsupervised Learning**

Unsupervised learning involves using datasets without clearly noticing the dependent (response) variable. Unsupervised means that the machine or computer should learn patterns from the data without referring to any specific response. Unsupervised learning aims to explore the data structure and generate a hypothesis rather than to test any hypothesis by statistical methods or to construct prediction or classification models based on a set of conditions and a specified response. Algorithms for unsupervised learning can be subdivided into two categories: clustering algorithms and informative data transformations (Valkenborg, Rousseau, Geubbelmans, & Burzykowski, 2023).

Unsupervised learning algorithms seek to find structure in data without recourse to labels (as in supervised learning) or reward signals (as in reinforcement learning). Just what “structure” amounts to, and how we should determine better or worse structures without any external feedback, are problems that each unsupervised learning algorithm must resolve in its way (Watson, 2023).

**3. Semi-supervised Learning**

Semi-supervised learning is a branch of machine learning that combines supervised and unsupervised learning by using labeled and unlabeled data to train artificial intelligence (AI) models for classification and regression tasks. In traditional supervised learning, one uses labeled data to build a model. However, labeling the training data for real-world applications is difficult, expensive, or time-consuming, as it requires the effort of human annotators, sometimes with specific domain experience and training. This is especially true for applications that involve learning with many class labels and sometimes with similarities among them (IBM, 2023).

Semi-supervised learning (SSL) addresses this inherent bottleneck by allowing the model to integrate part or all of the available unlabeled data in its supervised learning. Though semi-supervised learning is generally employed for the same use cases in which one might otherwise use supervised learning methods, it is distinguished by various techniques that incorporate unlabeled data into model training and the labeled data required for conventional supervised learning.

**4. Reinforcement Learning**

Reinforcement Learning (RL) is a type of learning guided by a specific objective. An agent learns by interacting with an unknown environment, typically in a trial-and-error way. This is the most common learning method for a child, who does something and observes what happens. The agent receives feedback regarding a reward (or punishment) from the environment; then, it uses this feedback to train itself and collect experience and knowledge about the environment. Reinforcement Learning problems are related to learning the best action to perform, situation-by-situation, to maximize the aggregated reward. RL agent has to learn a policy (i.e., a complete mapping between situations and actions) by trying actions out without any domain expert telling it, as in many other forms of machine learning. Another relevant characteristic of an RL problem is that in any situation, the agent has to choose between exploiting its current knowledge of the environment (acting that has already been tried previously in that situation) or exploring actions never tried before in that situation (Naeem, Rizvi, & Coronato, 2020).

### 2.2.3 Applications of Machine Learning

Some popular applications include medical diagnosis, credit risk analysis, customer profiling, market segmentation, targeted marketing, retail management, and fraud detection. In recent years, new data have been available due to various technological advances and research efforts, and consequently, new domains where machine learning can be applied have arisen.

Voice assistants take text inquiries as input, interpret them using machine learning models, and then take some action based on that interpretation. These systems incorporate a Natural Language Processing (NLP) module and an Automatic Speech Recognition (ASR) feature that translates analog voice inputs into text. Smartphone cameras use a neural network trained on millions of photos to identify images. The camera uses machine learning to determine what it looks at, then adjusts the lighting, selects the best exposure, and fine-tunes the colors to suit the scene best. Discussed in (Kumar, Kaur, & Singh, 2020), numerous forms of machine learning algorithms may be applied to various tasks, including data mining, predictive analytics, image processing, and many more.

With the help of convolutional neural networks, face unlocking can accurately categorize and recognize a user's face, allowing for foolproof authentication. Automatic face detection from a database of millions of photographs requires first identifying features in those images.  
Then, the training of a neural network is done. In order to filter data and give the most relevant items  
To consumers, Recommendation Engine combines two methods, namely content-based and collaborative filtering. It does this by calculating the degree to which a user's past and preferences are similar. E-commerce sites that offer us the newest things based on our browsing history are one example, as are food delivery services that show us the restaurants they recommend based on our order history (Kaur, Chandel, Brar, & Sharma, 2023).

### 2.3 Neural Networks

Neural networks, also known as artificial neural networks (ANNs) or artificially generated neural networks (SNNs), are a subset of machine learning that provides the foundation of deep learning techniques. The human brain inspires their name and form, and they replicate how real neurons communicate (Qamar & Zardari, 2023). Artificial neural networks can learn by example, like the way humans do. An artificial neural net is configured for a specific application, like pattern recognition, through a learning process. Learning in biological systems consists of adjustments to the synaptic connections between neurons, which is also true of artificial neural networks. The artificial neural network learns by updating the network architecture and connection weights so that the network can efficiently perform a task. It can learn either from available training patterns or automatically learn from examples or input-output relations ( Islam, Chen, & Jin, 2019).

Artificial neural networks can be applied to an increasing number of real-world problems of considerable complexity. They are used for solving problems that are too complex for conventional technologies or problems that do not have an algorithmic solution. They are up-and-coming systems in many forecasting and business classification applications due to their ability to learn from the data. Neural network-based models continue to achieve impressive results on longstanding machine learning problems, but establishing their capacity to reason about abstract concepts has proven difficult.

#### Layers of a neural network

1. **Input Layer**

A neural network's input layer comprises a collection of artificial input neurons. They send information from the initial neuron layers to the system for processing. The neural network's input layer initiates the workflow.

1. **Hidden Layer**

The artificial neural network's hidden layer comprises input and output layers, and the number of input weights determines the input and output of the artificial neurons.

1. **Output Layer**

The last layer of neurons is an output layer in an artificial neural network that provides specific outputs to the programmer. Because they are the network's final "performer" nodes, neurons in the output layer can be built and treated differently.

### 2.3.2 Strengths of Artificial Neural Networks

Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can extract patterns and detect trends too complex to be noticed by either humans or other computer techniques. A trained neural network can be considered an "expert" in the information it has been given to analyze. This expert can then be used to provide projections given new situations of interest and answer "what if" questions (Qamar & Zardari, 2023). Other benefits include adaptive learning, self-organization, real-time operation, and fault tolerance.

## 2.4 Deep Learning algorithms

According to the article “What is Deep Learning” by Amazon Web Services [AWS], Deep Learning (DL) was introduced by Hinton et al. in 2006, and it was based on the concept of artificial neural network (ANN). Deep learning models are neural networks designed after the human brain. A human brain contains millions of interconnected biological neurons that work together to learn and process information. Similarly, artificial neurons are software modules called nodes that use mathematical calculations to process data. Deep learning neural networks comprise many layers of artificial neurons that work together to solve complex problems.

Deep learning differs from standard machine learning in terms of efficiency as the volume of data increases. DL technology uses multiple layers to represent data abstractions and build computational models. While deep learning takes a long time to train a model due to many parameters, it takes a short time to run during testing compared to other machine learning algorithms (Sakar, 2021).

### 2.4.1 Types of Deep Learning Algorithms

There are several types of deep learning algorithms. However, only three will be discussed here:

**1. Convolutional Neural Networks (CNN)**

In the field of DL, the CNN is the most famous and commonly employed algorithm. The main benefit of CNN compared to its predecessors is that it automatically identifies the relevant features without human supervision. CNNs have been extensively applied in various fields, including computer vision, speech processing, Face Recognition, etc. The structure of CNNs was inspired by neurons in human and animal brains, similar to a conventional neural network. More specifically, in a cat’s brain, a complex sequence of cells forms the visual cortex; the CNN simulates this sequence. The main reason to consider CNN is the weight-sharing feature, which reduces the number of trainable network parameters and helps the network enhance generalization and avoid overfitting. Concurrently learning the feature extraction and classification layers causes the model output to be highly organized and highly reliant on the extracted features. With CNN, large-scale network implementation is much easier than with other neural networks (Alzubaidi, et al., 2021).

**2. Recurrent Neural Networks (RNN)**

RNNs are a commonly employed and familiar algorithm in deep learning. RNN is mainly applied in the context of speech processing and NLP. Unlike conventional networks, RNN uses sequential data in the network. Since the embedded structure in the data sequence delivers valuable information, this feature is fundamental to various applications. For instance, it is important to understand the context of the sentence to determine the meaning of a specific word. Thus, it is possible to consider the RNN as a unit of short-term memory, where x represents the input layer, y is the output layer, and s represents the state (hidden) layer.

A deep RNN is introduced that lessens the learning difficulty in the deep network and brings the benefits of a deeper RNN based on these three techniques. However, RNNs’ sensitivity to the exploding gradient and vanishing problems represents one of the main issues with this approach. More specifically, during the training process, the reduplications of several large or small derivatives may cause the gradients to explode or decay exponentially. With the entrance of new inputs, the network stops thinking about the initial ones; therefore, this sensitivity decays over time (Alzubaidi, et al., 2021).

**3. Long Short-Term Memory Model (LSTM)**

The LSTM model is a powerful recurrent neural system designed to overcome the exploding or vanishing gradient problems that typically arise when learning long-term dependencies, even when the minimal time lags are very long. This can be prevented by using a constant error carousel (CEC), which maintains the error signal within each unit’s cell. Such cells are recurrent networks with an interesting architecture for extending the CEC with additional features, namely the input gate and output gate, forming the memory cell. The self-recurrent connections indicate feedback with a lag of one time step.  
A vanilla LSTM unit comprises a cell, an input gate, an output gate, and a forget gate. This forget gate was not initially a part of the LSTM network, but was proposed by Gers et al. in 2001 to allow the network to reset its state. The cell remembers values over arbitrary time intervals, and the three gates regulate the flow of information associated with the cell.   
In short, the LSTM architecture consists of a set of recurrently connected sub-networks, known as memory blocks. The idea behind the memory block is to maintain its state over time and regulate the information flow through non-linear gating units ( Van Houdt, Mosquera, & Nápoles, 2020).

## 2.5 Transformer model

The vanilla Transformer introduced by Vaswani et al. in 2017 is a sequence-to-sequence model and consists of an encoder and a decoder, each of which is a stack of L identical blocks. The Transformer architecture is built around an attention mechanism that helps the model understand relationships between words in a sequence without relying on recurrence or convolution. It uses a Query-Key-Value (QKV) structure, transforming each word into query, key, and value vectors. These are compared to compute attention scores, determining how much focus each word should have on the others.

Instead of performing this process once, Transformers use multi-head attention, which runs the attention mechanism in parallel across multiple subspaces. This allows the model to learn different aspects of word relationships. There are three types of attention used: self-attention (within the same sequence), masked self-attention (prevents future information from being decoded), and cross-attention (used in the decoder to focus on the encoder output).

Beyond attention, each layer of the Transformer includes a feed-forward network applied to each word individually to add more transformation depth. Residual connections and layer normalization are used after every major operation to help the model train more effectively and support stacking of many layers. Since Transformers do not have a built-in sense of order, they rely on positional encodings, extra information added to each word’s representation, to help the model understand the order of words in a sentence. These combined components make the Transformer highly efficient and powerful for handling long-range dependencies in text (Lin, Wang, Liu, & Qiu, 2022).

### 2.5.1 Modern Transformer-based models

The translation landscape has undergone a dramatic transformation due to the emergence of  
powerful transformer-based models like Bidirectional Encoder Representations from  
Transformers (BERT), Generative Pre-trained Transformer (GPT), and Text-to-Text Transfer  
Transformer (T5). These models have revolutionized Natural Language Processing (NLP) tasks,  
Particularly, machine translation, which leverages the “attention” mechanism.

BERT pre-trains a language model on a large corpus of text in a bidirectional manner, enabling it to capture context from both the left and right sides of a word. The Strengths of BERT excel in tasks requiring a deep understanding of the context, such as question answering and text completion. Its bidirectional approach allows it to capture intricate relationships between words in a sentence. The limitation of BERT is that it requires large amounts of data and computational resources for training. It processes text sequentially, making it computationally intensive and slower for long texts (Zayyanu, 2024).

GPT, developed by OpenAI, is another transformer-based model introduced in 2018. GPT models are generative and can produce coherent and contextually relevant text. This  
Characteristics help generate translations fluently and naturally. It generates text autoregressively, predicting the next word based on the preceding context. Its creative text generation abilities can be harnessed to explore diverse ways of expressing ideas and concepts in different languages, making translations more engaging and culturally appropriate. The limitations of GPT’s unidirectional nature might limit its understanding of context, as it only considers the preceding words in a sentence. It might face challenges in tasks requiring precise comprehension and extraction of information (Zayyanu, 2024).

T5, introduced by Google Research in 2019, is a versatile transformer-based model. Unlike  
BERT and GPT, T5 frame all NLP tasks as text-to-text tasks, unifying different tasks under a  
Common text-based format. It is pre-trained to convert one form of text into another, allowing it.  
To handle a wide array of tasks. The strengths of T5’s text-to-text framework simplify the task-  
Specific architectures make it highly flexible and easy to apply to various NLP tasks. It  
Achieves state-of-the-art performance across multiple benchmarks due to its unified architecture.  
The limitations of T5’s performance might be prejudiced by the superiority and variety of the  
Training data for diverse tasks. Training and fine-tuning T5 models can be resource-intensive,  
Especially for large-scale applications.

The variant models BERT, GPT, and T5 have been compared in various studies. A study found  
That BERT achieved higher accuracy compared to other models on the Stanford Question  
Answering Dataset (SQuAD) (Kılıç, 2023). GPT-4, despite not explicitly trained for biomedical texts, showed similar performance to the best BERT models in detecting PPIs (Hasin et al., 2023).

## 2.6 Stuttering

Stuttering, or stammering, is a language fluency disorder characterized by speech flow and rhythm disruptions, including pauses, hesitations, and repetitions of syllables, words, or sounds. Despite a normally functioning vocal apparatus, individuals with stuttering struggle with smooth and continuous speech delivery (Hussain, & Forshing , 2024).

Speaking a fluent language is a normal, complex, and multifaceted process involving the precise selection of words along with the coordinated and harmonious speech actions of respiratory, laryngeal, and articulatory muscles. This enables the continuous, uninterrupted, and seemingly effortless delivery of desired speech. Complex neurological mechanisms govern the coordination of these processes, from formulating and expressing language to articulating speech with proper intonation, culminating in the final production of everyday, fluent discourse (Syed, Shahzaeem, & Forshing , 2024).

Disruptions in fluent speech characterize stuttering. Depending on its underlying etiology, stuttering can be broadly categorized as developmental or acquired. Individuals who stutter frequently experience challenges in maintaining smooth speech flow despite knowing precisely what they want to convey. Behavioral accompaniments of stuttering may include rapid eye blinking and lip tremors, significantly impeding communication and affecting a person's quality of life and interpersonal relationships (Zunic, Sinanovic, & Majic, 2021).

### 2.6.1 Etiology of stuttering

According to Syed, Shahzaeem, and Forshing, in 2024, there were several causes of stuttering. They include:

**1. Developmental Stuttering**

This form of speech disfluency is the most prevalent type and occurs in young children during the critical period while they are still developing their speech and language abilities. One theory suggests that developmental stuttering arises when a child's speech and language skills fail to meet their verbal demands. Genetic conditions, such as Prader-Willi syndrome, Down syndrome, and Fragile X syndrome, have also been associated with stuttering, where the stuttering pattern is phonologically similar to the developmental variant. Conversely, acquired stuttering emerges later in life, often stemming from various underlying etiologies.

2. **Acquired Stuttering**

Acquired stuttering can be further categorized into neurogenic, psychogenic, and drug-induced causes. Alternatively, some classification systems distinguish stuttering into psychogenic and organic origins, where the latter category comprises developmental, neurogenic, and drug-induced stuttering.

**3. Neurogenic stuttering:** Neurogenic stuttering arises from damage to specific brain tissue regions due to an insult. Potential causes of such damage or injury include stroke (the most prevalent), significant traumatic brain injury, hypoxic-ischemic encephalopathy, various dementing diseases, Parkinson's disease, and sequelae of dialysis, multiple sclerosis, and epilepsy. Consistent with various injury mechanisms, neurogenic stuttering exhibits variability in its underlying neural substrates and does not distinctly localize to a specific brain region. Commonly affected areas include the bilateral hemispheres, subcortical white matter tracts, the cerebellum, and deep nuclei such as the basal ganglia, thalamus, and brainstem. Notably, the left hemisphere, typically dominant for language, is more commonly affected than the right hemisphere.

**4. Psychogenic stuttering:** Psychogenic stuttering is classified as a functional disorder, a modern term for what was previously known as conversion disorder, where psychological symptoms manifest as physical ones. This condition is among the various functional speech and voice disorders. For stuttering to be deemed psychogenic, changes in spoken patterns must be connected to an underlying emotional conflict and lack an identified so-called organic etiology. In addition, psychogenic stuttering usually co-occurs with other mental health issues and lacks a consistent emotional response to stuttering across different situations.

**5. Pharmacological stuttering:** Pharmacological stuttering arises due to the administration of specific pharmacological agents. Pharmacological mechanisms that can predispose to stuttering include increased dopamine levels, decreased GABA levels, medications with anticholinergic properties, and drugs affecting serotonin levels. Interestingly, while some medications may induce stuttering in specific individuals, they might alleviate it in others. Disproportionately high instances of stuttering have been reported with stimulants such as dexamphetamine and methylphenidate, antidepressants such as duloxetine, fluoxetine, and bupropion, and anti-seizure medications.

### 2.6.3. Traditional Approach to Stuttering Speech Therapy

Treatment protocols should be formulated to address the patient's concerns effectively, ensuring that therapeutic objectives align with the patient's goals. In cases of suspected developmental stuttering, any child exhibiting speech patterns indicative of stutter-like disfluencies, whether reported by parents or observed clinically, should be referred to a speech-language pathologist. Urgency in referral increases if the disfluency persists for 1 year or more or if parental reports indicate worsening severity or increased frequency of stuttering (Bhardwaj, Sharma, Kumar, Sharma, & Sharma, 2024).

Generally, speech and behavioral therapies are initiated early in children to leverage the nervous system's plasticity. These therapies aim to facilitate compensatory changes that can lead to usual fluency. Early intervention may help prevent the development of impaired social skills and negative perceptions of communication, often observed in individuals with long-standing stuttering.

For patients who continue to stutter persistently, therapeutic goals typically evolve to include managing secondary behaviors, developing appropriate compensatory techniques, fostering acceptance of stuttering as a natural part of the individual's identity, and reclaiming a sense of command over speech.

Various speech and behavioral therapies address stuttering, each offering potential benefits tailored to the individual's needs and therapeutic objectives and responses. However, medications have not shown effectiveness in managing developmental or persistent stuttering. Commonly used therapies for developmental stuttering include the Lidcombe program, the RESTART-DCM treatment, the Palin PCI treatment, and the Family-focused approach.

 In contrast to developmental stuttering, medications can be utilized to treat neurogenic stuttering. These medications include haloperidol (the most common), chlorpromazine, trifluoperazine, thioridazine, carbamazepine, sodium valproate, levetiracetam, risperidone, and olanzapine. Although these drugs may offer assistance, they are not typically the initial treatment option, primarily due to their high incidence of adverse effects. Speech therapy remains the cornerstone of treatment (Syed, Shahzaeem, & Forshing , 2024).

### 2.6.4 Online Speech Therapy

The landscape of speech therapy has significantly transformed in recent decades, shifting from traditional in-person sessions to digital interventions like e-learning and mobile applications. This evolution has been driven by the need for greater accessibility, personalized treatment, and improved patient engagement. Integrating digital tools has broadened the scope of treatment options, offering innovative approaches to support individuals with speech and communication disorders. Mobile apps, teletherapy platforms, interactive software, and virtual reality tools enable speech-language pathologists (SLPs) to deliver therapy in more flexible and engaging ways, catering to the diverse needs of their patients (Gould, 2024).

Integrating Artificial Intelligence (AI) and machine learning is further advancing speech therapy, enhancing the capabilities of digital interventions, and paving the way for more personalized and effective treatment approaches. AI algorithms can analyze speech patterns, identify areas of difficulty, and provide real-time feedback to patients, enabling them to improve their articulation, fluency, and overall communication skills. Machine learning models can be trained on vast datasets of speech samples to develop personalized treatment plans tailored to each patient's needs (Ain, & Imtiaz, 2025).

## 2.7 Review of Related Works

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Author/Year** | **Title of Work** | **Methods Used** | **Results** | **Gap** |
| Cordella , Marte, Liu, and Kiran (2024) | An Introduction to Machine Learning for Speech-Language Pathologists: Concepts, Terminology, and Emerging Applications | A comprehensive explanation of ML concepts, including everyday ML tasks and specific types of ML models. An  An examination of exemplary published papers in the aphasiology literature that have utilized ML approaches was also conducted. | The study illustrated that the best-performing model type depends on the research question. In other words, no one ML model type is universally best. | The study primarily focused on aphasia and related speech-language disorders, with little to no direct discussion of stuttering or its unique therapeutic challenges. There was also a lack of specific ML applications or case studies targeting stuttering interventions or therapy. |
| Brahmi, et al., (2024) | Exploring the Role of Machine Learning in Diagnosing and Treating Speech Disorders: A Systematic Literature Review | The study extensively examined existing literature on assistive technology for speech disorders. Specific attention was given to ML techniques, characteristics of exploited datasets in the training phase, speaker languages, feature extraction techniques, and the features employed by ML algorithms. | Support vector machines and neural networks (CNN, DNN) were the most utilized ML techniques (20%, 16.92%). Research also demonstrated that deep signal analysis of voice using ML techniques to recognize speech with disorders showed promising results by extracting significant features from these signals, such as Mel-frequency Cepstral Coefficients (MFCCs) and Spectro Temporal utterances. Combining these two features showed more reliable results than others. | The study was based on English-language datasets, with limited exploration of ML applications for non-English speakers or multilingual contexts. Most findings were based on technical performance (accuracy, precision, recall) rather than clinical outcomes or patient-reported benefits. |
| Bhardwaj, Sharma, Kumar, Sharma, and Sharma (2024) | Transforming pediatric speech and language disorder diagnosis and therapy: The evolving role of artificial intelligence | The review paper explored the synergy of Artificial Intelligence, Machine Learning, and Augmentative Alternative Communication (AAC) in pediatric speech and language impairments. It also highlighted current developments and potential future directions. | It was discovered that while AAC has traditionally been linked to those with significant speech and language impairments, recent developments suggest its potential as a supplemental strategy for children with Speech Sound Disorders (SSD). | While the study highlighted the promise of AI and AAC, it gave limited attention to practical barriers such as  data privacy and security,  Ethical concerns, cost, and accessibility of advanced technologies, as well as integration challenges in low-resource settings. |
| Mulfari, La Placa, Rovito, Celesti, and Villari (2022) | Deep learning applications in telerehabilitation speech therapy scenarios | The study focused on isolated word recognition for native Italian speakers with dysarthria. It exploited an existing mobile application to collect audio data from users with speech disorders while they performed articulation exercises for speech therapy purposes. With this data, a convolutional neural network was trained to spot a small number of keywords within atypical speech, according to a speaker-dependent method. | It was discovered that it is easier for dysarthric individuals to pronounce isolated words rather than a continuous sequence of words. By following this line, an approach towards isolated word recognition tasks was developed, and a mobile software with the possibility of providing the end user, who contributed training to the ASR system with his/her speech samples, with customized audio reinforcements related to the recognition status of the voice commands. | The system was trained in a speaker-dependent manner, meaning each user’s model was tailored to their voice. This approach did not address the challenge of building a speaker-independent model, which would be more scalable for broader clinical use. |
| Almutairi (2024) | Application of Artificial Intelligence in  Assessing Speech, Language, and Voice  Disorders: A Scoping Review | A comprehensive review of works covering AI applications in assessing speech, language, and voice disorders (SLVDs) was conducted. It was exclusive to the English language. | The study reported positive results, with AI demonstrating high accuracy (ranging from 0.92 to 0.98) in detecting language disorders, particularly in children. Additionally, various survey findings indicated that AI effectively distinguished individuals with and without language disorders across diverse age groups. The highest classification accuracy of 87.5% was obtained using the Artificial Neural Network (ANN) classifier. In contrast, Decision Trees (DT) and Support Vector Machines (SVM) demonstrated lower accuracies of 62.5% and 50%, respectively. | Lack of extensive research, especially in underrepresented regions and disorders. |
| Oh and Whitehead (2024) | Exploring the Integration of Machine Learning and AI in the Treatment and Diagnosis of Communication Disorders | When integrated into diagnosing and treating communication disorders, the study analyzed potential transformation by machine learning (ML) and artificial intelligence (AI). It specifically highlighted successful initial diagnosis rates achieved with ML classifiers. | It was discovered that a trained Automatic Speech Recognition system could be implemented in a tailored telerehabilitation as a treatment plan that allows patients to interact remotely with their pathologists to produce a highly personalized plan. This system would use the CNN model, focusing on the small keywords that adjust the design as needed. It would help the pathologist, as their supervision does not require as much time as if they had to create an independent assessment without the CNN, and have to meet in therapy sessions constantly. | Much of the research reviewed is at the prototype stage, with few studies demonstrating successful integration into clinical workflows or real-world therapy settings. |
| Loubser, De Villiers, and De Freitas (2024) | End-to-end automated speech recognition using a character-based small-scale transformer architecture | An analysis of a combination of a transformer and a convolutional neural network was evaluated. | It was discovered that the model produced results comparable to those of much larger speech recognition models. This model could be effectively implemented for cases where practitioners have limited access to data. | Reliance on limited data affected the model’s robustness, potentially reducing its effectiveness when applied to broader populations beyond the training set. |
| Al-Hussain, Shuweihdi, Alali, Househ, and Abd-alrazaq (2022) | The Effectiveness of Supervised Machine Learning in Screening and Diagnosing Voice Disorders: Systematic Review and Meta-analysis | A comprehensive analysis of studies that examined any ML algorithm's performance (accuracy, sensitivity, and specificity) in detecting pathological voice samples was carried out. Also, the risk of bias was assessed to determine the effectiveness of ML algorithms in screening and diagnosing voice disorders. | ML performance was found to be more promising when used as a screening tool rather than in diagnosis, achieving greater than 90% in all three outcomes: accuracy, sensitivity, and specificity. | The study excluded unsupervised and hybrid ML approaches, potentially overlooking relevant advances. The evidence for ML’s effectiveness in actual diagnosis (as opposed to screening) remains inconclusive due to the limited number of diagnostic studies and insufficient analysis of language and demographic diversity among participants. |

**CHAPTER 3**

**PROPOSED METHODOLOGY**

**3.1 Introduction**

The development of a web-based stuttering speech therapy application leverages advanced natural language processing capabilities provided by Hugging Face Transformers to deliver an accessible, user-centric tool for individuals seeking to improve their speech fluency. This methodology outlines the systematic approach to designing, implementing, and evaluating the application, ensuring it meets the therapeutic needs of users while incorporating state-of-the-art machine learning techniques. By integrating transformer-based models, the application aims to provide real-time speech analysis, personalized feedback, and adaptive exercises tailored to the unique challenges faced by individuals who stutter.

**3.2 Evaluation of Model**

The evaluation of transformer-based models, specifically Wav2Vec 2.0 from the Hugging Face ecosystem, will be essential for the stuttering speech therapy application. Wav2Vec 2.0 will be fine-tuned on stuttering-specific speech datasets to detect disfluencies such as repetitions, prolongations, and blocks in real time. These models will be selected over traditional Hidden Markov Models (HMMs) due to their superior contextual analysis and ability to process raw audio without manual feature engineering, offering better adaptability to varied speech patterns. Evaluation will focus on quantitative metrics—precision, recall, F1-score, and word error rate (WER)—using publicly available annotated datasets. Latency will be measured to ensure responses under 200 milliseconds for real-time feedback. Comparative benchmarks against HMMs will demonstrate improved accuracy and efficiency. Hugging Face’s automated pipelines will ensure reproducible evaluations. Results will inform model optimization and application development within the one-month timeframe.

**3.2.1 Data Collection**

The data collection process for the stuttering speech therapy application will center on the SEP-28k dataset, a comprehensive resource for stuttering event detection, sourced from the GitHub repository hosted by Apple at (Lea et al., 2021). This dataset includes annotations for approximately 28,000 3-second audio clips from public podcasts featuring individuals who stutter, supplemented by about 4,000 clips from the FluencyBank dataset. Each clip is labeled by three trained non-clinician annotators for stuttering events, including blocks, prolongations, sound repetitions, word repetitions, interjections, and a “no stuttered words” category, with labels indicating the number of annotators (out of three) who identified each event. The dataset is licensed under the Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0), ensuring compliance with non-commercial use restrictions.

The repository was cloned using Git to obtain the dataset’s annotation files and supporting scripts. After cloning, the repository contained the following files and directories:

1. **LICENSE**: The licensing terms for the dataset and code.
2. **README.md**: Documentation detailing the dataset structure, annotation process, and usage instructions.
3. **SEP-28k\_clips.csv**: A CSV file listing annotations for the 28,000 podcast clips, including clip identifiers, URLs for audio sources, and stuttering event labels.
4. **SEP-28k\_episodes.csv**: A CSV file with metadata for podcast episodes, such as episode identifiers and source URLs.
5. **fluencybank\_clips.csv**: A CSV file containing annotations for the 4,000 FluencyBank clips.
6. **fluencybank\_episodes.csv**: A CSV file with metadata for FluencyBank episodes.
7. **download\_audio.py**: A Python script to download raw audio files from URLs provided in the episode CSV files.
8. **extract\_clips.py**: A Python script to process downloaded audio into 3-second clips corresponding to the annotations.

Due to copyright restrictions, the audio files were not included in the repository. Instead, the raw audio was retrieved using the download\_audio.py script, which accessed URLs listed in the SEP-28k\_episodes.csv and fluencybank\_episodes.csv files. The script was executed with the following command, specifying directories for raw audio storage ([WAV\_DIR]) and clip extraction ([CLIP\_DIR]):

python download\_audio.py --wav\_dir [WAV\_DIR] --clip\_dir [CLIP\_DIR]

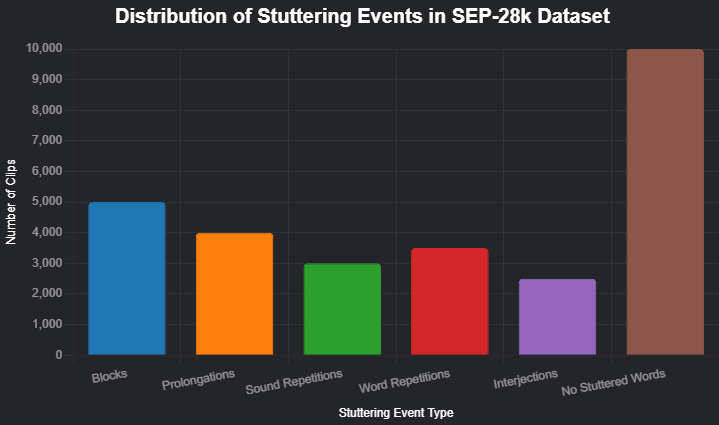
This downloaded the original podcast and FluencyBank audio files to [WAV\_DIR]. Subsequently, the extract\_clips.py script was run to segment the raw audio into 3-second clips, aligning with the annotations in SEP-28k\_clips.csv and fluencybank\_clips.csv. The command was:

python extract\_clips.py --wav\_dir [WAV\_DIR] --clip\_dir [CLIP\_DIR]

The extracted clips were stored in [CLIP\_DIR], ensuring each clip matched the annotated segments. To maintain data integrity, the process verified that the number of extracted clips corresponded to the entries in the annotation CSV files (approximately 32,000 total clips). Any discrepancies, such as missing audio files due to unavailable URLs, were logged, and alternative sources, if available, will be explored within the one-month timeframe. The dataset will be stored locally in a structured format, with annotations and audio clips organized for efficient access during model training and evaluation. This approach ensures the acquisition of high-quality, annotated speech data tailored for stuttering detection, adhering to licensing constraints and enabling rapid progression to model development.

**3.2.2 Data Description**

The SEP-28k dataset, sourced from (Lea et al., 2021), comprises approximately 32,000 3-second audio clips for stuttering speech therapy. It includes 28,000 clips from public podcasts featuring individuals who stutter and 4,000 clips from the FluencyBank dataset. Each clip, sampled at 16 kHz in WAV format, captures speech segments annotated for stuttering events—blocks, prolongations, sound repetitions, word repetitions, interjections, or “no stuttered words”—by three non-clinician annotators, with labels reflecting annotator agreement (0–3). The audio, not included in the repository due to copyright, was downloaded using the download\_audio.py script from URLs in episode metadata files and segmented into clips with extract\_clips.py, ensuring alignment with annotations. The dataset, licensed under CC BY-NC 4.0, provides diverse, high-quality speech data for model training.

****

**Fig 3.2.1- SEP-28k Speech disorder distribution**

**3.2.3 Data analysis**The data analysis process is to explore and preprocess the stuttering datasets (SEP-28k and FluencyBank) to extract meaningful patterns and prepare clean, labeled audio data suitable for training a machine learning model (e.g., Wav2Vec 2.0). This analysis helps determine the frequency, types, and distribution of disfluency events such as blocks, prolongations, repetitions, and interjections.

### 3.2.3.1 Data Preprocessing Steps

1. **Audio Retrieval:** Full-length podcast episodes will be retrieved using the provided download\_audio.py script
2. **Clip Extraction:** The extract\_clips.py script will be used in conjunction with SEP-28k\_labels.csv to extract 3-second clips based on annotated timestamps. From Table 4.1, the start and stop column represent the duration of each clip.
3. **Path Validation:** File paths will be validated using os.path.exists(), and any missing audio files will be logged
4. **Storage Structure:** Extracted clips will be organized into folders by show and episode (e.g., clips/WomenWhoStutter/109/)
5. **Audio Normalization:** Audio signals will be normalized to maintain consistent amplitude levels across all clips. The sample rate would be converted to 16,000hz.
6. **Noise Reduction:** Techniques such as spectral gating may be applied to reduce background noise and enhance speech clarity
7. **Feature Extraction:** Standard features such as Mel-Frequency Cepstral Coefficients (MFCCs), zero-crossing rate, spectral rolloff, and chroma features will be extracted using libraries like librosa.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Show | EpId | ClipId | Start | Stop | Unsure | PoorAudioQuality | Prolongation | Block | SoundRep |
| HeStutters | 0 | 0 | 31900320 | 31948320 | 0 | 0 | 0 | 0 | 0 |
| HeStutters | 0 | 1 | 31977120 | 32025120 | 0 | 0 | 0 | 0 | 0 |
| HeStutters | 0 | 2 | 34809760 | 34857760 | 0 | 0 | 0 | 0 | 0 |
| HeStutters | 0 | 3 | 35200640 | 35248640 | 0 | 0 | 1 | 0 | 0 |
| HeStutters | 0 | 4 | 35721920 | 35769920 | 0 | 0 | 0 | 0 | 0 |
| HeStutters | 0 | 5 | 36435040 | 36483040 | 0 | 0 | 1 | 0 | 0 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| WordRep | DifficultToUnderstand | Interjection | NoStutteredWords | NaturalPause | Music | NoSpeech |
| 0 | 0 | 0 | 3 | 1 | 0 | 0 |
| 0 | 0 | 0 | 3 | 1 | 0 | 0 |
| 0 | 0 | 0 | 3 | 0 | 0 | 0 |
| 0 | 0 | 0 | 2 | 0 | 0 | 0 |
| 0 | 0 | 0 | 3 | 0 | 0 | 0 |
| 0 | 0 | 0 | 2 | 0 | 0 | 0 |

Table 3.2.1- Head of the Sep-28k\_labels.csv file

1. Show: The show coloumn represents the various podcast stutters such as, Hestutters, Stuttertalk, WomenWhoStutter.
2. Epid: This represents each podcast Id.
3. ClipId: This represents the Id of each clip in each podcast.
4. Start: This represents the strating time of a clip in milli-second.
5. Stop: This represents the ending time of a clip in milli-second.
6. Unsure: If “1”, the type of stuttering of that particular clip isn/t verified.
7. Pooraudioquality: If “1”, the audio clip was uncler.
8. Prolongation: If “1”, the type of stuttering of that particular clip is prolongation.  
   Wordrep: If “1”, the type of stuttering of that particular clip is word repitition.
9. Interjection: If “1”, the type of stuttering of that particular clip is interjection.
10. NoStutteredWords: If “1”, the clip is clear and there is no stuttered words.
11. NaturalPause: Shows if the pauses made is natural.
12. Music: If “1”, there is background music or noise in that clip.
13. Nospeech: There is no spoken words.

**3.3 Model Evaluation and Fine-Tuning**

This section details the fine-tuning and evaluation of the Wav2Vec 2.0 model for detecting stuttering events in a real-time speech therapy application. The process adapts the pre-trained model to a specialized task and assesses its performance to ensure it meets accuracy and speed requirements critical for effective user feedback.

**3.3.1 Fine-Tuning the Wav2Vec 2.0 Model**

The Wav2Vec 2.0 model, initially developed by Facebook AI for speech recognition, is fine-tuned using the SEP-28k dataset, which includes 32,000 annotated 3-second audio clips of stuttering speech. These clips are labeled for events such as blocks, prolongations, and repetitions. Fine-tuning refines the model's ability to identify these specific patterns.

**3.3.1.1 Fine-Tuning Steps**:

1. **Data Preparation**: Audio clips from SEP-28k are preprocessed, with labels determined by majority vote among annotators.
2. **Training**: The model is further trained on these clips using a GPU, adjusting its parameters to focus on stuttering detection.
3. **Validation**: A held-out subset of the dataset tests the model's generalization to new audio.

This process leverages the model's pre-trained speech understanding while tailoring it to the therapy application.

**3.3.2 Evaluation Metrics: Precision and Recall**

Performance is measured using **precision** and **recall**, key metrics for classification tasks in a therapeutic context:

1. **Precision**: The ratio of correctly identified stuttering events to all events flagged by the model.  
   Formula:  
   [  
   Precision =

True Positives

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

True-Positives+False-Positives

High precision reduces false positives, enhancing trust in feedback.

1. **Recall**: The ratio of correctly identified stuttering events to all actual events.  
   Formula:  
     
   Recall =

True Positives

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

True-Positives+False-Negatives  
  
High recall ensures most stuttering events are detected, supporting comprehensive feedback.

These metrics prioritize reliability over simpler measures like accuracy, aligning with the application's goals.

**3.3.3 Achieving Real-Time Feedback**

The system aims to targets an **inference time of under 200 milliseconds** per 3-second audio clip to deliver real-time feedback. This speed ensures an interactive experience, critical for speech therapy.

**Why 200ms?**: Based on user perception thresholds, delays below 200ms feel instantaneous, enhancing engagement.

**How Achieved**: Optimizations include using Wav2Vec 2.0 Base, GPU acceleration, and techniques like model quantization.

Low latency supports immediate feedback, making the tool responsive and user-friendly.

**3.3.4 Practical Implementation**

The fine-tuned model will aim to perform **clip-level classification**, analyzing 3-second audio segments to detect stuttering events. The workflow in the application is:

1. **Segmentation**: Speech is split into 3-second clips, potentially overlapping (e.g., every 1 second).
2. **Processing**: Each clip is analyzed in under 200ms.
3. **Feedback**: Predictions trigger user feedback (e.g., "Repetition detected—try pausing").

Overlapping windows enable frequent updates, simulating real-time interaction despite the 3-second analysis window.

**3.3.5 Implications and Considerations**

1. **Feasibility**: The 200ms target is met with optimizations, making real-time deployment practical.
2. **Limitations**: Clip-level detection lacks exact event timing; finer analysis may require enhanced annotations.
3. **Challenges**: Audio variability (e.g., noise, accents) could impact accuracy, suggesting future robustness enhancements.

This approach supports the therapy application's need for timely, actionable insights.

**3.4 Design and Implementation of the Web-Based System**

The web-based system is a cornerstone of the stuttering speech therapy application, designed to deliver an accessible, responsive, and user-centric tool for speech analysis and therapy. This section outlines the system’s design and implementation, covering the frontend, backend, integration of the fine-tuned machine learning model, and considerations for usability and real-time performance.

**3.4.1 Overview**

The application enables users to record their speech, receive immediate feedback on stuttering events (e.g., blocks, prolongations), and access personalized therapy exercises. It comprises a **frontend** built with TypeScript and React for dynamic user interaction and a **Flask backend** that integrates the fine-tuned Wav2Vec 2.0 model. The system prioritizes intuitiveness and efficiency to ensure users can engage with therapy seamlessly, without technical barriers.

**3.4.2 System Architecture**

The system adopts a **client-server architecture** to facilitate efficient communication between components:

1. **Frontend**: A web interface, developed using TypeScript and React, allows users to record speech and view feedback.
2. **Backend**: A Flask server processes audio inputs and interfaces with the machine learning model.
3. **Machine Learning Model**: The fine-tuned Wav2Vec 2.0 model, hosted on a server, analyzes audio for stuttering events.

These components interact cohesively: the frontend sends audio data to the backend, which processes it through the model and returns actionable feedback to the user.

**3.4.3 Frontend: Speech Recording and Feedback**

The frontend is to be built with **TypeScript** and **React**, ensures a robust and interactive user experience, leveraging TypeScript’s static typing for reliability and React’s component-based structure for modularity.

**Technologies**:

* 1. **TypeScript**: Provides type safety to prevent errors and enhance code maintainability.
  2. **React**: Enables dynamic, component-based UI development for seamless interactivity.
  3. **Tailwind CSS**: Used for styling to create a visually appealing, responsive, and accessible interface.

**Key Features**:

* 1. **Speech Recording**: A React component with a "Record" button captures audio via the browser’s Web Audio API.
  2. **Feedback Display**: Real-time feedback (e.g., “Prolongation detected—try slowing down”) is rendered dynamically using React state management.
  3. **Responsive Design**: The interface adapts to various devices (desktop, tablet, mobile) using Tailwind CSS’s responsive utilities.
  4. **Progress Tracking**: A component displays user progress (e.g., fluency improvements) via simple charts.

The use of TypeScript and React ensures a scalable, maintainable frontend that prioritizes user engagement and accessibility.

**3.4.4 Backend: Flask Integration**

The backend will be implemented with **Flask**, a lightweight Python web framework, manages the application’s core logic and model integration:

**Key Functions**:

* 1. Receives audio files from the frontend via HTTP requests.
  2. Segments audio into 3-second clips, consistent with the SEP-28k dataset format.
  3. Sends clips to the Wav2Vec 2.0 model for stuttering event detection.
  4. Processes model predictions and generates user-friendly feedback for the frontend.

**Why Flask?**: Flask’s simplicity and compatibility with Python-based machine learning libraries make it ideal for integrating the Wav2Vec 2.0 model and handling audio processing tasks.

The backend ensures efficient data flow between the frontend and the model, supporting real-time functionality.

**4.3.5 Model Integration**

The fine-tuned **Wav2Vec 2.0 model** is integrated into the backend to analyze audio in real time:

**Integration Process**:

1. Audio clips are uploaded from the frontend to the Flask server.
2. The backend preprocesses the audio (e.g., ensuring 16 kHz sample rate) and sends it to the model.
3. The model predicts stuttering events (e.g., repetitions, blocks) for each 3-second clip.
4. Predictions are translated into concise, actionable feedback (e.g., “Repetition detected—try pausing before the word”) and sent back to the frontend.

**Optimizations**:

1. **Asynchronous Processing**: Flask uses asynchronous routes to minimize latency during model inference.
2. **Model Efficiency**: A lightweight Wav2Vec 2.0 variant (e.g., Wav2Vec 2.0 Base) is used to achieve sub-200ms inference times.
3. **Caching**: Frequently accessed model outputs are cached to reduce redundant computations.
4. This integration ensures accurate and timely stuttering detection, critical for the therapy application.

**3.4.6 Usability and Real-Time Performance**

The system is designed to prioritize **usability** and **real-time performance**, ensuring it meets therapeutic and technical requirements:

**Usability**:

1. **Intuitive Interface**: Clear buttons (e.g., “Record,” “Stop”) and concise feedback messages cater to users of all ages.
2. **Accessibility**: Features like high-contrast modes and text-to-speech options support diverse users.
3. **Engagement**: Immediate feedback and progress tracking keep users motivated.

**Real-Time Performance**:

* 1. **Latency Target**: Inference time is kept below 200 milliseconds per 3-second clip, achieved through model optimization and efficient backend processing.
  2. **Scalability**: The Flask server is designed to handle multiple users, with cloud hosting (e.g., AWS) to ensure reliability.
  3. **Audio Processing**: Audio is segmented and analyzed in overlapping 3-second windows to provide frequent updates, simulating real-time interaction.

These design choices make the application effective and practical for speech therapy.

**3.5 Software Development Life Cycle (SDLC)**

The development of the web-based stuttering speech therapy application will follow the **Software Development Life Cycle (SDLC)**, a structured methodology that ensures systematic and high-quality project execution. The SDLC comprises seven phases—planning, requirement analysis, design, implementation, testing, deployment, and maintenance—each customized to achieve the project’s goal of delivering an accessible, real-time, and user-centric therapy tool. This section details the application of each phase, supported by diagrams to illustrate the system’s structure and processes.

**3.5.1 Planning**

The planning phase establishes the project’s foundation by defining its scope, objectives, and resources.

1. **Objective**: Develop a web-based application using machine learning to provide real-time stuttering detection and personalized therapy.
2. **Scope**: Focus on stuttering detection using the SEP-28k dataset and Wav2Vec 2.0 model, targeting British English speakers.
3. **Resources**: Tools include Python, librosa, Flask, TypeScript, React, Hugging Face Transformers, and a GPU-enabled server. The project timeline is set for one month.
4. **Risk Assessment**: Potential challenges, such as dataset access issues or model latency, are identified, with mitigation strategies like alternative data sources and model optimization.

**3.5.2 Requirement Analysis**

The requirement analysis phase defines the functional and non-functional needs of the application to ensure it meets user expectations.

**Functional Requirements**:

* 1. Enable browser-based audio recording for user speech input.
  2. Detect stuttering events (e.g., blocks, prolongations) in real time.
  3. Provide personalized feedback and adaptive therapy exercises.
  4. Support user authentication for data privacy.

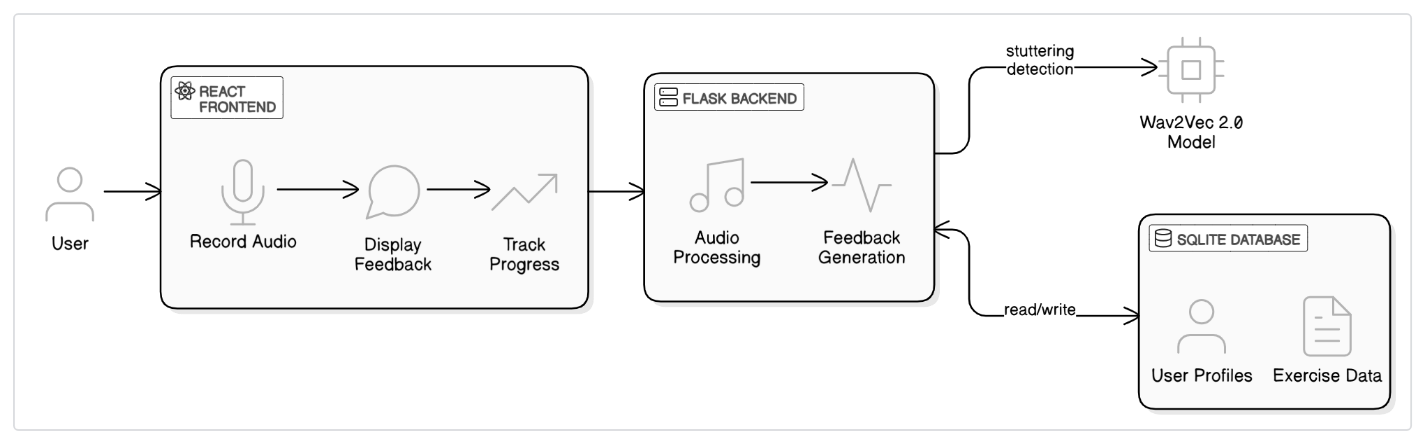
**Non-Functional Requirements**:

* 1. Achieve inference time under 200 milliseconds for real-time feedback.
  2. Ensure a user-friendly interface accessible to all age groups.
  3. Maintain data security through encryption.

This phase produces a detailed requirement specification document to guide development.

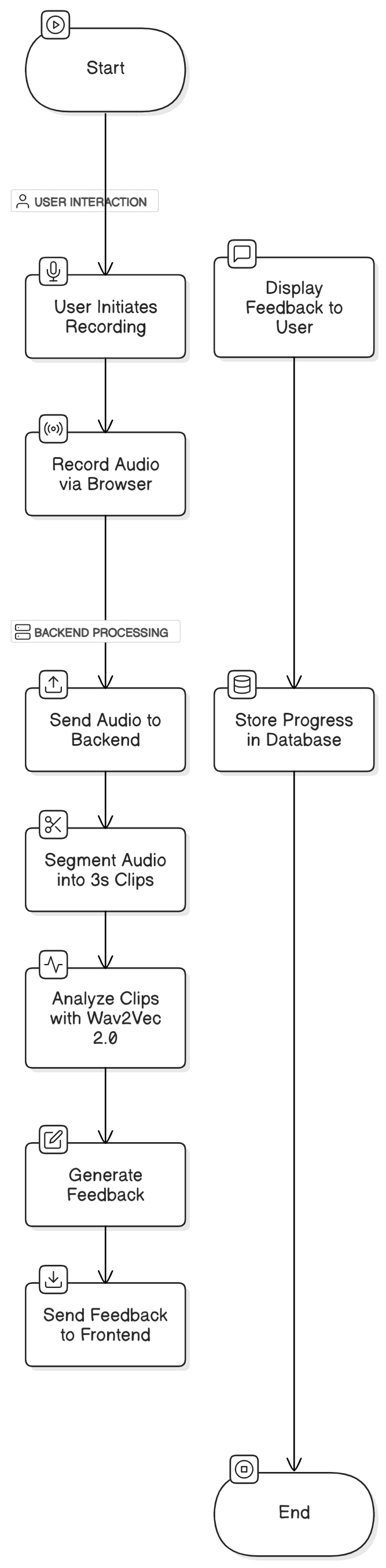
**3.5.3 Design**

The design phase outlines the system’s architecture and technical specifications, supported by diagrams to visualize its structure and workflows.

1. **System Architecture**: A client-server model with a frontend (TypeScript, React, Tailwind CSS) and a Flask backend integrating the Wav2Vec 2.0 model. See **Figure 3.5.1: Architectural Model** for a diagram illustrating the components and their interactions.

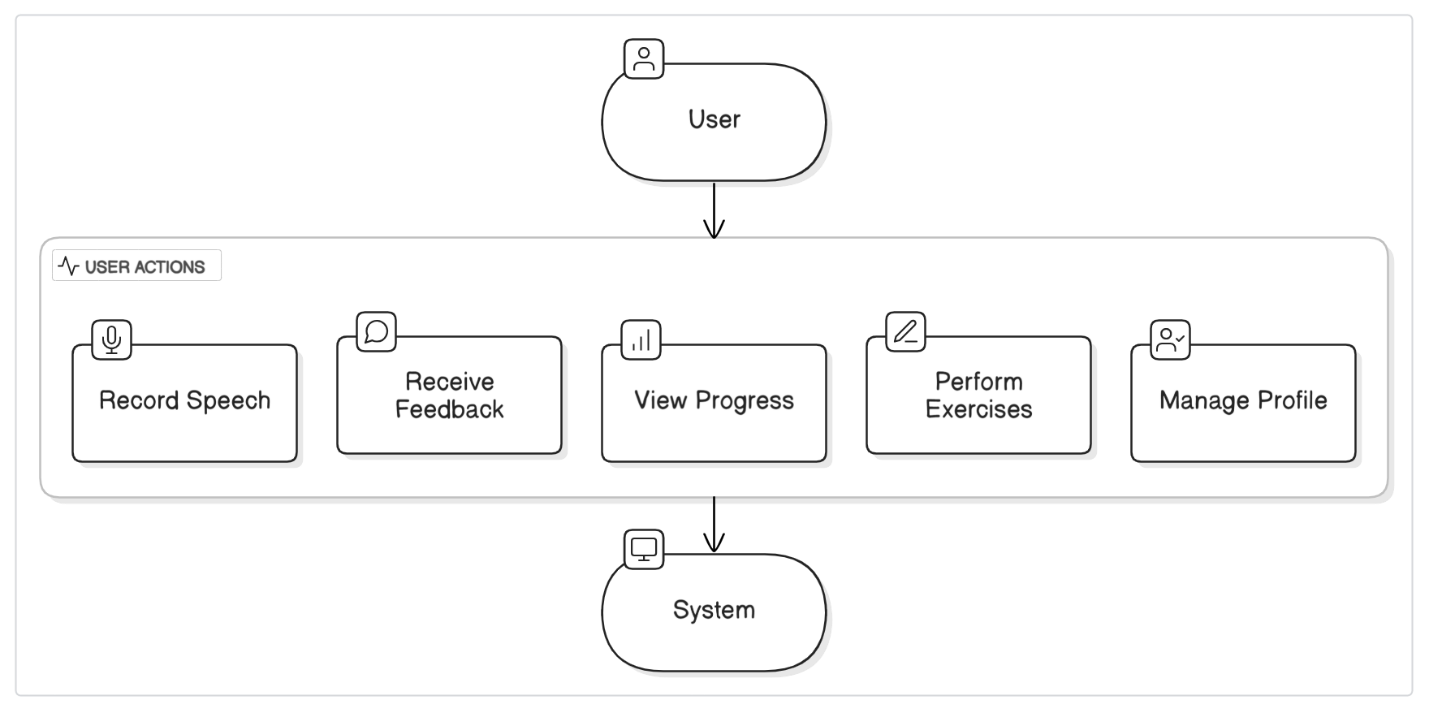
**Fig 3.5.1**- Architectural Model

1. **Data Flow**: Audio is recorded, segmented into 3-second clips, processed by the model, and returned as feedback. Refer to **Figure 3.5.2: Dataflow Diagram** for the data processing pipeline.



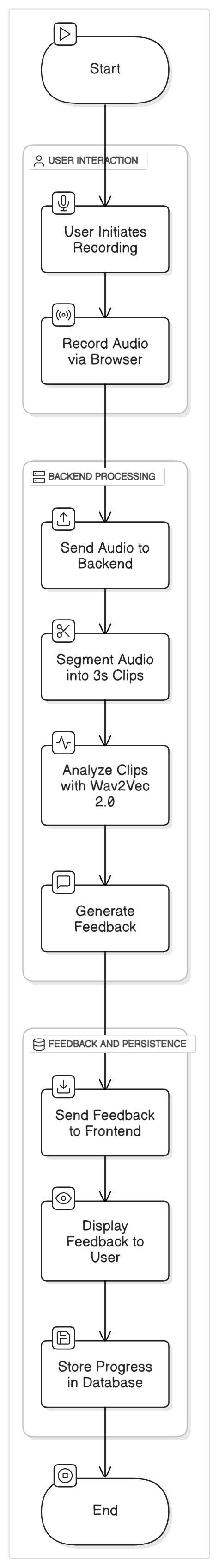
**Figure 3.5.2: Dataflow Diagram**

1. **User Interface**: A responsive design with a record button, feedback display, and progress tracking, built using React components.
2. **Database**: A lightweight SQLite database stores user profiles and exercise data.
3. **Use Case Diagram**: **Figure 3.5.3: Use Case Diagram** depicts user interactions, such as recording speech, receiving feedback, and viewing progress.



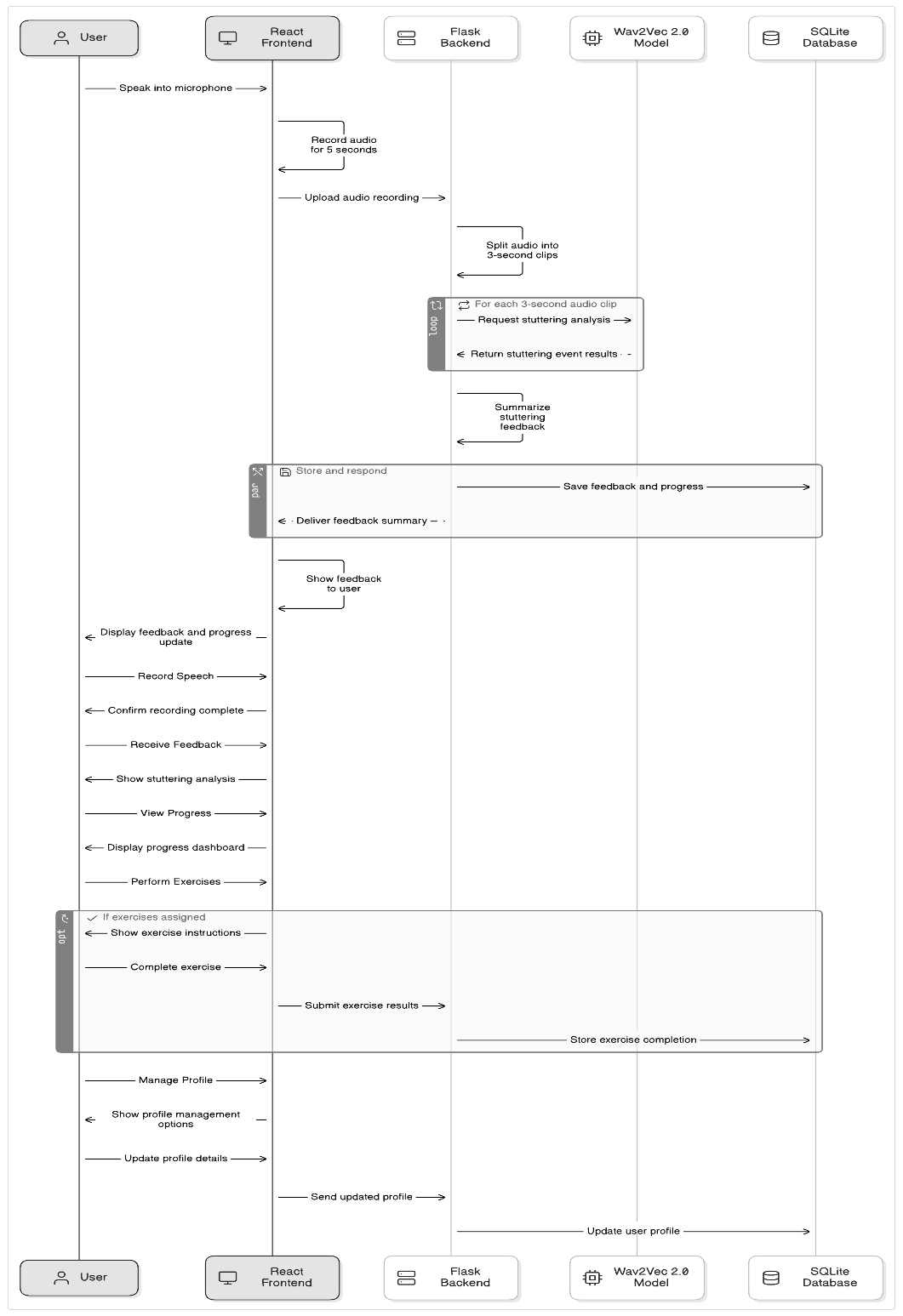
**Figure 3.5.3: Use Case Diagram**

1. **Activity Diagram**: **Figure 3.5.4: Activity Diagram** outlines the workflow of recording and analyzing speech.



**Figure 3.5.4: Activity Diagram**

1. **Sequence Diagram**: **Figure 3.5.5: Sequence Diagram** illustrates the interaction sequence between the frontend, backend, and model during audio processing.

****

**Figure 3.5.5: Sequence Diagram**

1. **Performance Optimization**: Model inference is optimized for low latency, and the backend is designed for scalability.

The design ensures technical feasibility and user-centric functionality, with diagrams providing a clear visual representation.

**3.5.4 Implementation**

The implementation phase involves coding and integrating the system components to create a functional prototype.

1. **Frontend Development**: TypeScript and React, styled with Tailwind CSS, create a dynamic interface for recording audio, displaying feedback, and tracking progress.
2. **Backend Development**: Flask manages audio processing, model integration, and feedback generation.
3. **Model Integration**: The fine-tuned Wav2Vec 2.0 model is deployed on a server, processing 3-second audio clips.
4. **Tools**: Python libraries (librosa, Hugging Face Transformers) support audio preprocessing and model operations.

This phase translates the design into a working system, ensuring all components function cohesively.

**3.5.5 Testing**

The testing phase verifies the system’s functionality, performance, and usability to ensure it meets requirements.

1. **Unit Testing**: Tests individual components, such as audio segmentation and model predictions, for accuracy.
2. **Integration Testing**: Validates seamless interaction between the frontend, backend, and model.
3. **Performance Testing**: Confirms inference time under 200 milliseconds and overall system responsiveness.
4. **Usability Testing**: Conducted with sample users (e.g., students or volunteers) to ensure the interface is intuitive.
5. **Model Evaluation**: Measures precision and recall using a test subset of the SEP-28k dataset.

# References

Ain Quratul & Imtiaz Rabia. (2025). The role of technology in speech-language therapy: Perceptions, effectiveness, and challenges in a resource-limited setting. *Journal of Health and Rehabilitation Research*, *5*(1), 1–8. https://doi.org/10.61919/jhrr.v5i1.1754

Alam, A. (2023). *What is machine learning?* https://doi.org/10.5281/zenodo.8231581

Al-Hussain, G., Shuweihdi, F., Alali, H., Househ, M., & Abd-alrazaq, A. (2022). The effectiveness of supervised machine learning in screening and diagnosing voice disorders: Systematic review and meta-analysis. *Journal of Medical Internet Research*, *24*(10), e38472. https://doi.org/10.2196/38472

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021a). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, *8*(1), 53. <https://doi.org/10.1186/s40537-021-00444-8>

*Assistive technology*. (n.d.). Retrieved May 23, 2025, from https://www.who.int/news-room/fact-sheets/detail/assistive-technology

Bhardwaj, A., Sharma, M., Kumar, S., Sharma, S., & Sharma, P. C. (2024). Transforming pediatric speech and language disorder diagnosis and therapy: The evolving role of artificial intelligence. *Health Sciences Review*, *12*, 100188. https://doi.org/10.1016/j.hsr.2024.100188

Brahmi, Z., Mahyoob, M., Al-Sarem, M., Bousselmi: , K., & Alblwi, A. (2024). *Exploring the Role of Machine Learning in Diagnosing and Treating Speech Disorders: A Systematic Literature Review*. *17*, 2205–2232. https://doi.org/https://doi.org/10.2147/prbm.s460283

Chen, M. (2024, November 25). *What is Machine Learning?* Oracle Cloud Infrastructure. https://www.oracle.com/ng/artificial-intelligence/machine-learning/what-is-machine-learning/

Cordella, C., Marte, M. J., Liu, H., & Kiran, S. (2025). An introduction to machine learning for speech-language pathologists: Concepts, terminology, and emerging applications. *Perspectives of the ASHA Special Interest Groups*, *10*(2), 432–450. https://doi.org/10.1044/2024\_PERSP-24-00037

Foote, K. D. (2021, December 3). *A brief history of machine learning*. DATAVERSITY. https://www.dataversity.net/a-brief-history-of-machine-learning/

Gould, L. (2024). *Enhancing Speech Therapy with Digital Tools: The Role of Technology in Modern Practice.* *163*(6), 877–882. https://doi.org/10.4172/2472-5005.1000275

Hasin, R., Çam, N. B., Basmaci, M., Zheng, J., Jemiyo, C., He, Y., Özgür, A., & Hur, J. (2024). Evaluating GPT and BERT models for protein–protein interaction identification in biomedical text. *Bioinformatics Advances*, *4*(1), vbae133. <https://doi.org/10.1093/bioadv/vbae133>

Hussain, S. A. S., & Lui, F. (2025). Stuttering(Stammering). In *StatPearls*. StatPearls Publishing. http://www.ncbi.nlm.nih.gov/books/NBK603738/

Islam, M., Chen, G., & Jin, S. (2019). An overview of neural network. *American Journal of Neural Networks and Applications*, *5*(1), 7. https://doi.org/10.11648/j.ajnna.20190501.12

Kaur, C., Chandel, Richa, Brar, Tejinder Pal, & Sharma, Shikha. (2023). *Machine Learning and its Applications- A Review Study*. *52*.

Kılıç, M. E. (2023). *Ai in medical education: A comparative analysis of gpt-4 and gpt-3. 5 on turkish medical specialization exam performance*. https://doi.org/10.1101/2023.07.12.23292564

Kumar, Y., Kaur, K., & Singh, G. (2020). Machine learning aspects and its applications towards different research areas. *2020 International Conference on Computation, Automation and Knowledge Management (ICCAKM)*, 150–156. https://doi.org/10.1109/ICCAKM46823.2020.9051502

Lin, T., Wang, Y., Liu, X., & Qiu, X. (2022). A survey of transformers. *AI Open*, *3*, 111–132. https://doi.org/10.1016/j.aiopen.2022.10.001

Loubser, A., De Villiers, P., & De Freitas, A. (2024). End-to-end automated speech recognition using a character based small scale transformer architecture. *Expert Systems with Applications*, *252*, 124119. https://doi.org/10.1016/j.eswa.2024.124119

*Machine Learning and its Applications- A Review Study*. (n.d.).

Mahato, R. (2022). *Artificial Intelligence, What Is it?* . 197–202.

Almutairi, M. (2024). *Application of artificial intelligence in assessing speech, language, and voice disorders: A scoping review*. *24*(10). https://doi.org/10.5281/ZENODO.10776638

Mulfari, D., La Placa, D., Rovito, C., Celesti, A., & Villari, M. (2022). Deep learning applications in telerehabilitation speech therapy scenarios. *Computers in Biology and Medicine*, *148*, 105864. https://doi.org/10.1016/j.compbiomed.2022.105864

Naeem, M., Rizvi, S. T. H., & Coronato, A. (2020). A gentle introduction to reinforcement learning and its application in different fields. *IEEE Access*, *8*, 209320–209344. https://doi.org/10.1109/ACCESS.2020.3038605

Ngyende, A. (2012). *Census 2011*. Statistics South Africa. https://www.statssa.gov.za/publications/P03014/P030142011.pdf

Oh, L., & Whitehead, G. (2024). Exploring the integration of machine learning and ai in the treatment and diagnosis of communication disorders. *Journal of Student Research*, *13*(2), 1–6. <https://doi.org/10.47611/jsrhs.v13i2.6791>

Olawade, D. B., Wada, O. Z., Odetayo, A., David-Olawade, A. C., Asaolu, F., & Eberhardt, J. (2024). Enhancing mental health with Artificial Intelligence: Current trends and future prospects. *Journal of Medicine, Surgery, and Public Health*, *3*, 100099. https://doi.org/10.1016/j.glmedi.2024.100099

Popovici, D.-V., & Buică-Belciu, C. (2012). Professional challenges in computer-assisted speech therapy. *Procedia - Social and Behavioral Sciences*, *33*, 518–522. https://doi.org/10.1016/j.sbspro.2012.01.175

Qamar Roheen & Zardari Baqar Ali . (2023). Artificial neural networks: An overview. *Mesopotamian Journal of Computer Science*, *2023*, 124–133. https://doi.org/10.58496/MJCSC/2023/015

Sarker, I. H. (2021). Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions. *SN Computer Science*, *2*(6), 420. https://doi.org/10.1007/s42979-021-00815-1

Sheikh, H., Prins, C., & Schrijvers, E. (2023). *Mission ai: The new system technology*. Springer International Publishing. https://doi.org/10.1007/978-3-031-21448-6

Shuaib (Fwacs), A. K., Afolabi (Fwacs), O. A., Alabi (Fwacs), B. S., & Elemunkan (PhD. Speech Therapist), I. O. (2011). Epidemilogical profile of speech and language disorder in north central nigeria. *International Journal of Biomedical Science*, *7*(4), 268–272. https://doi.org/10.59566/IJBS.2011.7268

Taye, M. M. (2023). Understanding of machine learning with deep learning: Architectures, workflow, applications and future directions. *Computers*, *12*(5), 91. https://doi.org/10.3390/computers12050091

Valkenborg, D., Rousseau, A.-J., Geubbelmans, M., & Burzykowski, T. (2023). Unsupervised learning. *American Journal of Orthodontics and Dentofacial Orthopedics*, *163*(6), 877–882. https://doi.org/10.1016/j.ajodo.2023.04.001

Van Houdt, G., Mosquera, C., & Nápoles, G. (2020). A review on the long short-term memory model. *Artificial Intelligence Review*, *53*(8), 5929–5955. https://doi.org/10.1007/s10462-020-09838-1

Watson, D. S. (2023). On the philosophy of unsupervised learning. *Philosophy & Technology*, *36*(2), 28. https://doi.org/10.1007/s13347-023-00635-6

*What is deep learning? - Deep learning ai explained - aws*. (n.d.). Amazon Web Services, Inc. Retrieved May 23, 2025, from https://aws.amazon.com/what-is/deep-learning/

*What is semi-supervised learning? | ibm*. (2023, December 8). https://www.ibm.com/think/topics/semi-supervised-learning

Xu, Y., Liu, X., Cao, X., Huang, C., Liu, E., Qian, S., Liu, X., Wu, Y., Dong, F., Qiu, C.-W., Qiu, J., Hua, K., Su, W., Wu, J., Xu, H., Han, Y., Fu, C., Yin, Z., Liu, M., … Zhang, J. (2021). Artificial intelligence: A powerful paradigm for scientific research. *The Innovation*, *2*(4), 100179. https://doi.org/10.1016/j.xinn.2021.100179

Zayyanu, Z. M. (2024). Revolutionising translation technology: A comparative study of variant transformer models - bert, gpt, and t5. *Computer Science & Engineering: An International Journal*, *14*(3), 15–27. https://doi.org/10.5121/cseij.2024.14302

Zunic, L., Sinanovic, O., & Majic, B. (2021). Neurogenic stuttering: Etiology, symptomatology, and treatment. *Medical Archives*, *75*(6), 456. https://doi.org/10.5455/medarh.2021.75.456-461