# AI-Enhanced Interview and Assessment System

**Team Members:**

Kritika Acharya(THA077BCT024)
Rashna K.C.(THA077BCT039)
Suyog Acharya(THA077BCT045)
Yogesh Aryal( THA077BCT046)

**Supervised By:**

Er. Sudip Rana

Department of Electronics and Computer Engineering
Institute of Engineering, Thapathali Campus

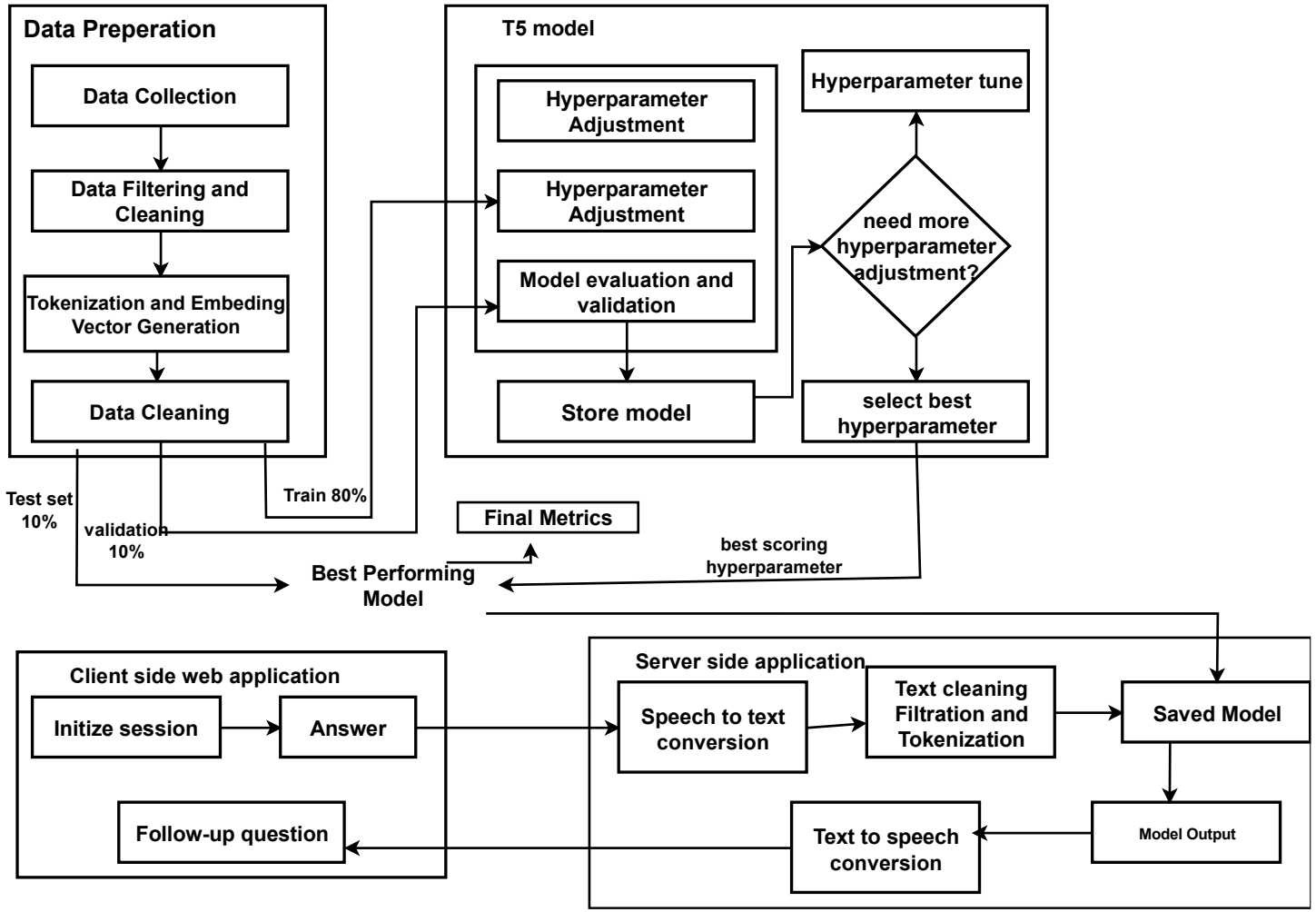
July, 2024

# Objectives

- To implement advanced AI models like T5, Wavenet, Whisper, OpenCV for precise assessment of candidate’s technical qualifications, coding responses, and problem-solving abilities.

- To automate the technical interview process, reducing manual effort and ensuring unbiased evaluations.

# Methodology - [1] (System Block Diagram)



# Methodology - [2]

- We have meticulously prepared a dataset of 2084 questions for MERN stack interview.
- Each entry in the dataset includes fields such as 'question,' 'answer,' 'followup_question,' 'difficulty,' 'topic,' 'category,' and 'keywords' to ensure comprehensive coverage and facilitate easy retrieval and context switching.
- To enhance the system's capability, we are fine-tuning a pre-trained T5 (Text-to-Text Transfer Transformer) model on our dataset.
- This fine-tuning process adapts the model to generate and ask questions that are specifically tailored to MERN stack development.

# Methodology - [3]

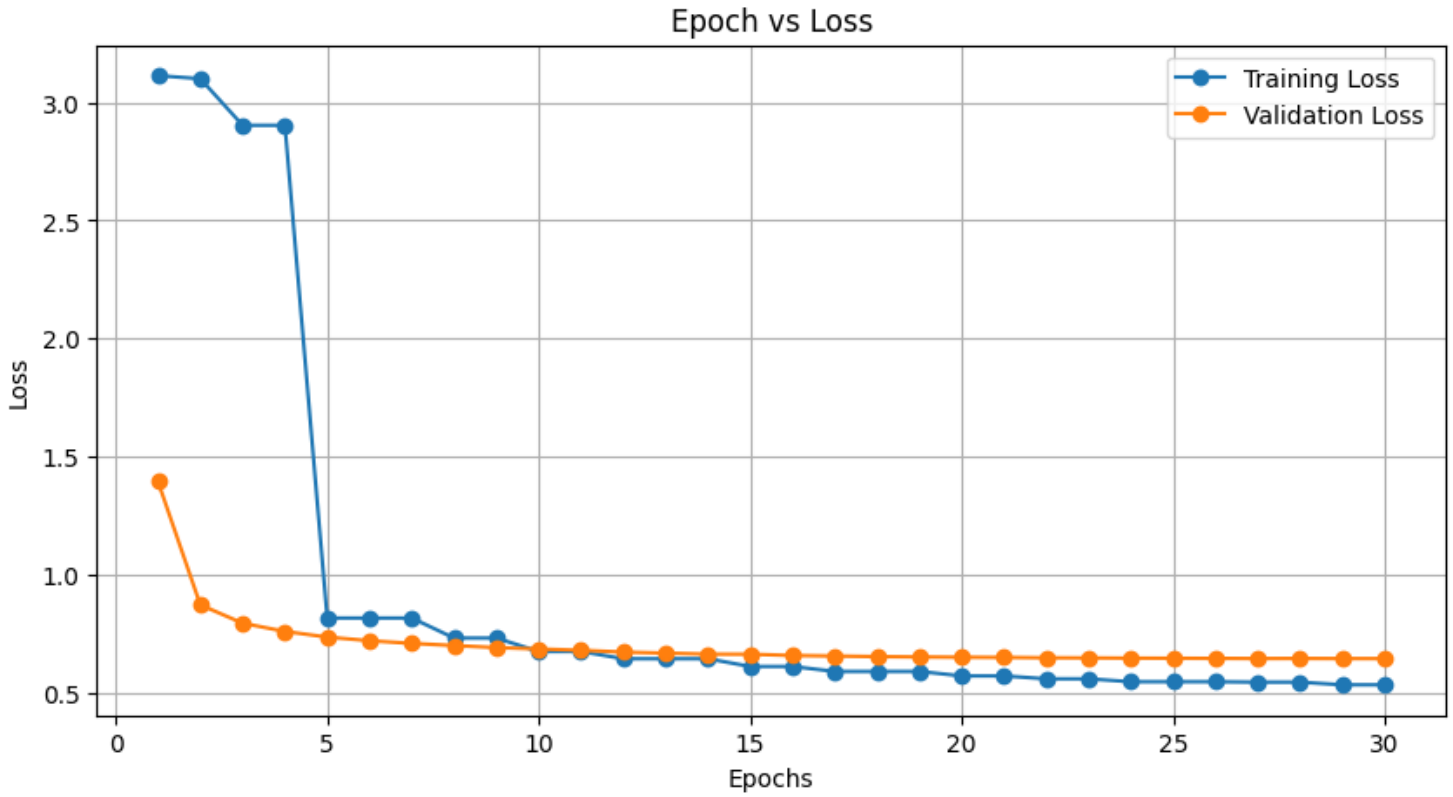
- During the interview, the model will dynamically generate questions based on the candidate's responses, ensuring that the difficulty and relevance of questions are appropriately adjusted.
- We will still be collecting more dataset and continue fine tuning our model until we get the best hyperparameters.
- We are still working on handling context switching seamlessly based on the job description provided (frontend, backend, or full MERN stack) context switching
- After properly implementing scoring mechanism based on keywords matching, this model will be ready for further implementations.

# Results

After training T5 for Conditional Generation Model for 30 epoch, we have achieved the following results:

- BLEU Score: 19.190004885263253
- ROUGE1: 0.434205237202659 9
- ROUGE2: 0.24555359560370596
- ROUGEL: 0.4008472264115854
- Perplexity: 1.14749276638031
- Cosine similarity: 0.6690911688991612

# Results-[1]
# Loss Plot



# Results-[2]
# Question Generation

```
Time left for MongoDB: 1:27
Total time left: 36:27

Question (Medium) : Describe the role of db.collection.find() method in MongoDB.
Your answer:  Describe the role of db.collection.find() method in MongoDB.

Time left for MongoDB: 0:12
Total time left: 35:12

Question (Medium) : What is a MongoDB Status?
Your answer: In Node.js, "status" typically refers to the current state or health of an application, including its runtir

--- Starting Node.js section (Time limit: 15 minutes) ---

Time left for Node.js: 14:59
Total time left: 34:25

Question (Easy) : Give an example of creating a File system using'status' in Node.js.
Your answer: In Node.js, use `fs.stat` to check a file's status and existence, e.g., `fs.stat('example.txt', (err, stats)

Time left for Node.js: 14:14
Total time left: 33:40

Question (Easy) : How do you check a file's status and existence?
Your answer:
```

# Results-[3]
# Question Generation

```
Question (Easy) : What is the purpose of the delete command in MongoDB?
Your answer: The purpose of the `delete` command in MongoDB is to remove documents from a collection based on specified crite

Time left for MongoDB: 9:37
Total time left: 44:37

Question (Easy) : What is the purpose of MongoDB's $db.collection.update command?
Your answer: The purpose of MongoDB's `$db.collection.update` command is to modify existing documents in a collection based o

Time left for MongoDB: 8:54
Total time left: 43:54

Question (Easy) : How does MongoDB handle data migration?
Your answer: MongoDB handles data migration by using tools and processes such as `mongodump` and `mongorestore` for backing u

Progressing to Medium difficulty questions.

Time left for MongoDB: 8:30
Total time left: 43:30

Question (Medium) : How do you manage MongoDB's $mongodump aggregation stage for data migration?
Your answer: MongoDB’s `$mongodump` is not an aggregation stage; it's a utility for creating backups of your database, which

Time left for MongoDB: 8:03
Total time left: 43:03

Question (Medium) : Describe the $mongodump method used for aggregation in MongoDB.
Your answer: MongoDB’s `$mongodump` is not an aggregation stage; it's a utility for creating backups of your database, which

Time left for MongoDB: 7:23
Total time left: 42:23
```

# Results-[4]
# Dataset


```
    {
        "question": "Explain how to achieve high availability with file descriptor management in Node.js.",
        "answer": "Achieving high availability with file descriptor management in Node.js involves using load balancing, clustering, or distributed systems to distribute file operations across multiple instances. It ensures redundancy and fault tolerance.",
        "followup_question": "How do you troubleshoot performance issues related to file descriptor usage in Node.js?",
        "difficulty": "hard",
        "topic": "Node.js",
        "keywords": [
            "high availability",
            "file descriptor management",
            "Node.js",
            "file handling"
        ],
        "category": "backend"
    },
```

# Results Analysis

- On training the model in 2083 questions dataet, we achieved BLEU Sore of 19.19 which indicates a moderate level of similarity between the generated and reference texts.
- A ROUGE- 1 :A score of  0.434 indicates a substantial overlap, suggesting that the generated text captures many of the important words from the reference text.
- ROUGE-2: A score of 0.246 indicates that some bigram sequences are correctly captured
- ROUGE-L: A score of 0.401 shows that significant portions of the reference text are being generated correctly.

# Results Analysis-[1]

- Perplexity : A perplexity score of 1.147 is quite low, indicating that the model is very confident in its predictions.
- Cosine Similarity: . A cosine similarity of 0.669 suggests that the generated text is relatively close to the reference text in the vector space

# Results Analysis-[2]

In the Loss Plot,

- The training loss starts at around 3.0 and drops sharply to about 0.8 by epoch 5.
- The validation loss starts at around 1.5 and decreases rapidly to around 0.8 by epoch 5.
- This sharp decrease in both training and validation loss indicates that the model is quickly learning the underlying patterns in the data during the initial training phase.
- Both training and validation losses stabilize, with training loss around 0.5 and validation loss around 0.6-0.7.
- The close values of training and validation loss suggest that the model is not overfitting and generalizes well to unseen data.

# Results Analysis-[3]
# T5 for Conditional Generation-[1]

For question generation using Pretrained-T5 for Conditional Generation model,

- The T5 tokenizer from the pretrained T5-base model is used to convert text into token IDs.
- The tokenization is applied to the dataset. The result is a DataFrame where each row contains the tokenized input and target text.
- We have implemented LoRA (Low-Rank Adaptation) method to fine-tune pretrained model in a parameter-efficient way.
- LoRA adapts the pretrained model by learning low-rank matrices that approximate the necessary updates.

# Results Analysis-[4]
# T5 for Conditional Generation-[2]

- **Model Loading**: Loads T5 model and tokenizer from a specified path for question generation.
- **Configuration**: Defines interview categories with time limits and sets difficulty levels and corresponding scores.
- **Question Management**: Tracks asked questions to avoid duplicates and defines subcategories for each main category.
- **Generate Unique Questions**: Uses T5 model to generate questions based on category, difficulty, subcategory, and previous answers.
- **Evaluate Answers**: Placeholder function to score user answers.
- **Format Time**: Converts time into minutes and seconds for display.

# Results Analysis-[5]
# T5 for Conditional Generation-[3]

- **Interview Process**: Manages the interview flow, tracking time and scores, adjusts difficulty based on user performance and provides updates on remaining time and total score.
- **Execution**: Runs the interview simulation by initiating the main function.

# Results Analysis-[5]
# Dataset Overview

Concise explanation of each key in the dataset:

-**question**: The main query or prompt that is being asked. It provides the context or topic that needs to be addressed.

-**answer**: The response or explanation provided for the question. It includes detailed information relevant to the query.

- **followup_question**: A related question that seeks to delve deeper into the topic or address related issues.

-**difficulty**: The level of challenge or complexity associated with the question, often categorized as easy, medium, or hard.

- **topic:** The specific area or subject matter of the question.

# Results Analysis-[6]
# Dataset Overview-[1]

-**keywords**: A list of important terms or phrases associated with the question and answer. These help in understanding the core concepts and themes.

-**category**: The broader classification or group under which the question falls. It groups questions into categories based on their nature or application area.

# Results Analysis-[7]
# Dataset Overview-[2]

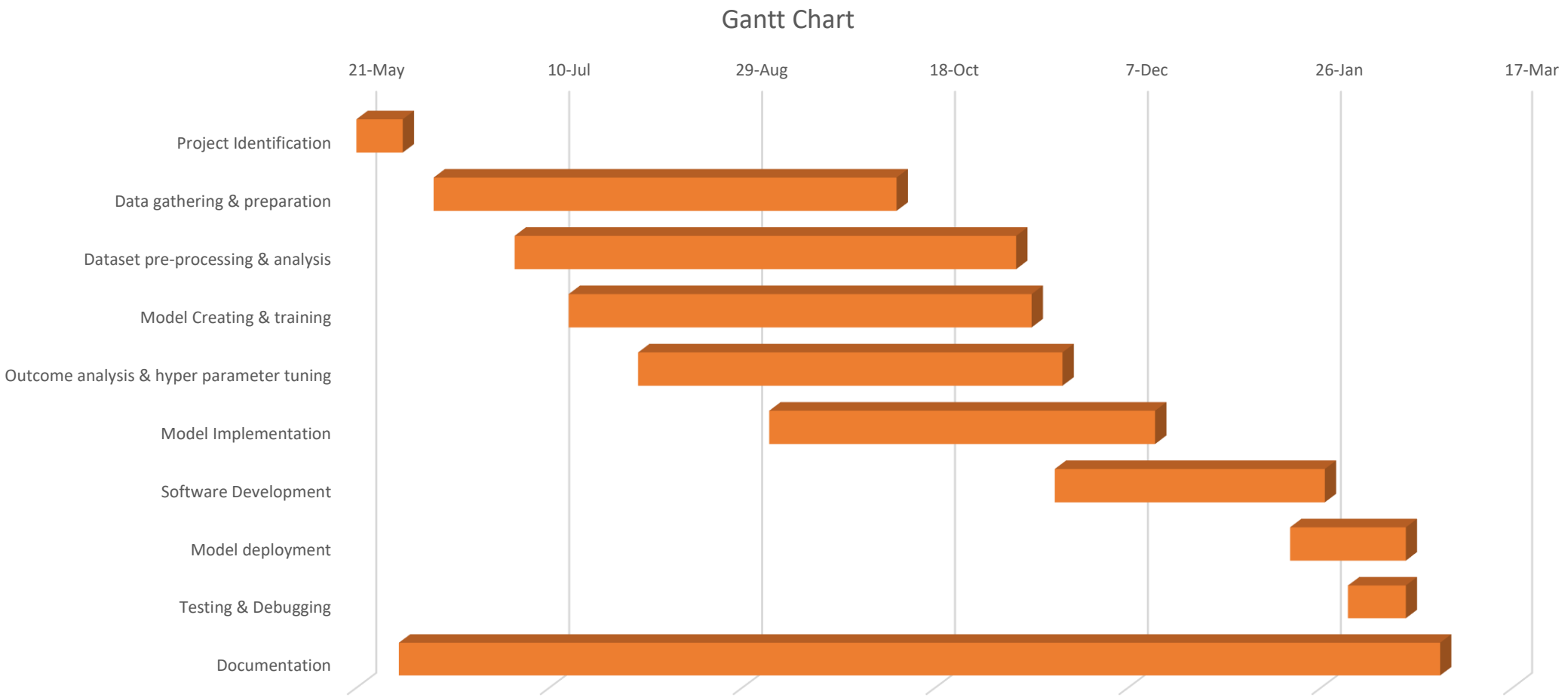
**Dataset Collection Sources:**

- GeeksForGeeks
- Flexiple
- Turing
- MDN Web Docs
- Official docs of each language

# Remaining Tasks

- Increasing the question dataset
- Using different fine tuning techniques and hyperparameter adjustment to increase the performance of T5 for Conditional Generation Model
- Integrating the model of T5 that generates introductory questions with T5 for Conditional Generation Model
- Adding the scoring mechanism in answers
- Integrating TTS and SST with final model
- Integrating the proctoring system
- Developing user friendly interface

# Tentative Timeline



# Estimated Project Expenses

| **Cost Domain** | **Expected Cost** |
|---|---|
| Cloud Storage | Rs. 3000 |
| Miscellaneous | Rs. 3000 |
| **Total** | **Rs. 6000** |

# References – [1]


- [1] F. Chopra and I. Haalad, "Conducting Qualitative Interviews with AI(2023)," Cesifo Working Papers, p. 10666, 2023.
- [2] B. C. Lee and B. Y. Kim, "Development of an AI-based Interview System for Remote Hiring," International Journal of Advanced Research in Engineering and Technology (IJARET), vol. 12, no. 3, pp. 654-663, 2021.
- [3] A. Nigam, R. Pasricha, T. Singh and P. Churi, "A Systematic Review on AI-based Proctoring Systems: Past, Present and Future," Educ Inf Technol, vol. 26, pp. 6421-6445, 2021.
- [4] P. R. S B, M. Agnihotri and D. B. Jayagopi, "Improving Asynchronous Interview Interaction with Follow-up Question Generation," International Institute of Information Technology, vol. 6, pp. 79-89, 2021.

# References – [2]


- [5] T. R.K, B. Shinde, N. Rasal and S. Ghorpade, "Smart Interview System using AI Technology," International Research Journal of Modernization in Engineering Technology and Science, vol. 6, no. 2, 2024.
- [6] R. Mengi, H. Ghorpade and A. Kakade, "Fine-tuning T5 and RoBERTa Models for Enhanced Text Summarization and Sentimental Analysis," 2023.
- [7] N. Mulla and P. Gharpure, "Automatic question generation: a review of methodologies, datasets, evaluation metrics, and applications, Progress in Artificial Intelligence, vol. 12, pp. 1-32, 2023.

# References – [3]


- [8] M. Wng, P. XIe, Y. Du and X. Hu, "T5-Based Model for Abstractive Summarization: A Semi-Supervised Learning Approach with Consistency Loss Functions," Applied Sciences, vol. 13, no. 12, 2023.
- [9] V. D. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," arXiv preprint arXiv:1609.03499, vol. 12, 2016.