**NATIONAL UNIVERSITY OF MODERN LANGUAGES**

**ISLAMABAD**



MACHINE LEARNING (LAB)

**PROJECT REPORT**

Nida Shafiq

**Analyze Multiple Binary Detection Techniques**

The project will employ a combination of deep learning architectures, specifically **Convolutional Neural Networks (CNNs)** and **Long Short-Term Memory (LSTM) networks**, to extract meaningful features from the audio signals.

This project implements a hybrid model that combines the strengths of **CNNs** for feature extraction and **LSTMs** for sequential data processing.

**Key components:**

* **Dataset:** The TORGO dataset, containing audio files from dysarthric and non-dysarthric speakers.
* **Feature Extraction:** Mel-Frequency Cepstral Coefficients (MFCCs) are extracted from the audio files.
* **Model Training:** CNNs, LSTMs, and a hybrid CNN+LSTM model are trained using the extracted MFCCs as input.
* **Model Evaluation:** The performance of the trained models is evaluated using various metrics such as accuracy, precision, recall, and F1-score.

Here’s the proper explanation of each and every technique I used:

**Traditional Techniques:**

* **Convolutional Neural Network (CNN):**

**Implementation:**

The provided code implements a Convolutional Neural Network (CNN) architecture using Keras and TensorFlow. The CNN model consists of convolutional layers with ReLU activation, max pooling, and fully connected layers. It's compiled with the 'Adam' optimizer and 'sparse\_categorical\_crossentropy' loss function.

The model is trained on a set of MFCC features extracted from audio files, with labels representing the presence or absence of dysarthria.

**Evaluation:**

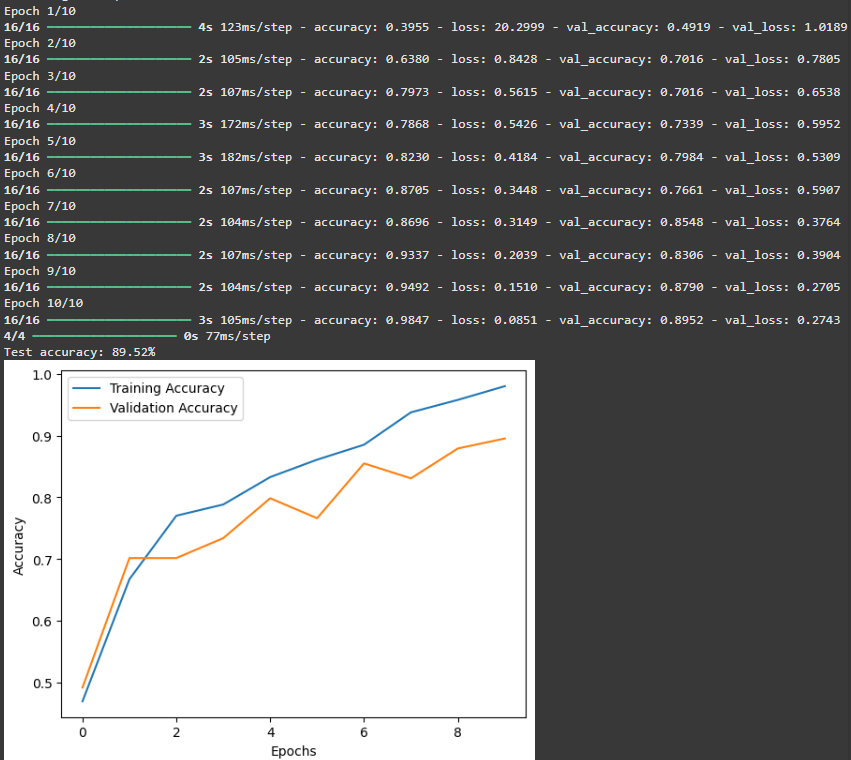
The model's performance is evaluated on accuracy, precision, recall, and F1-score.

The code provides visualizations for training history (loss and accuracy curves), confusion matrix, and a radar chart representing the model's performance metrics.

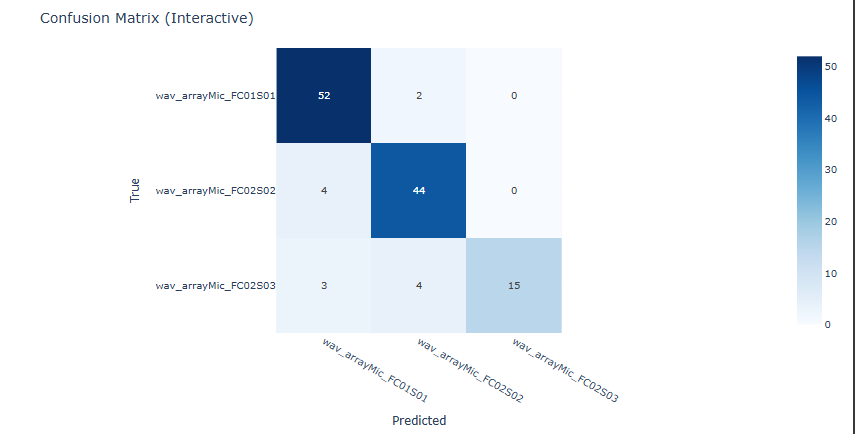
**Results:**

The CNN model achieves an accuracy of around 89.52%, with a precision of 88%, recall of 90%, and F1-score of 89%.

It is trained on 10 epochs and the training time was around 23.61 seconds.



**Visual Representation:**



* **Recurrent Neural Network (RNN):**

**Implementation:**

The code implements a Recurrent Neural Network (RNN) architecture using Keras and TensorFlow. Specifically, it uses a Long Short-Term Memory (LSTM) layer to handle temporal dependencies in the audio data. The model is trained on a set of MFCC features extracted from audio files, with labels representing the presence or absence of dysarthria.

**Evaluation:**

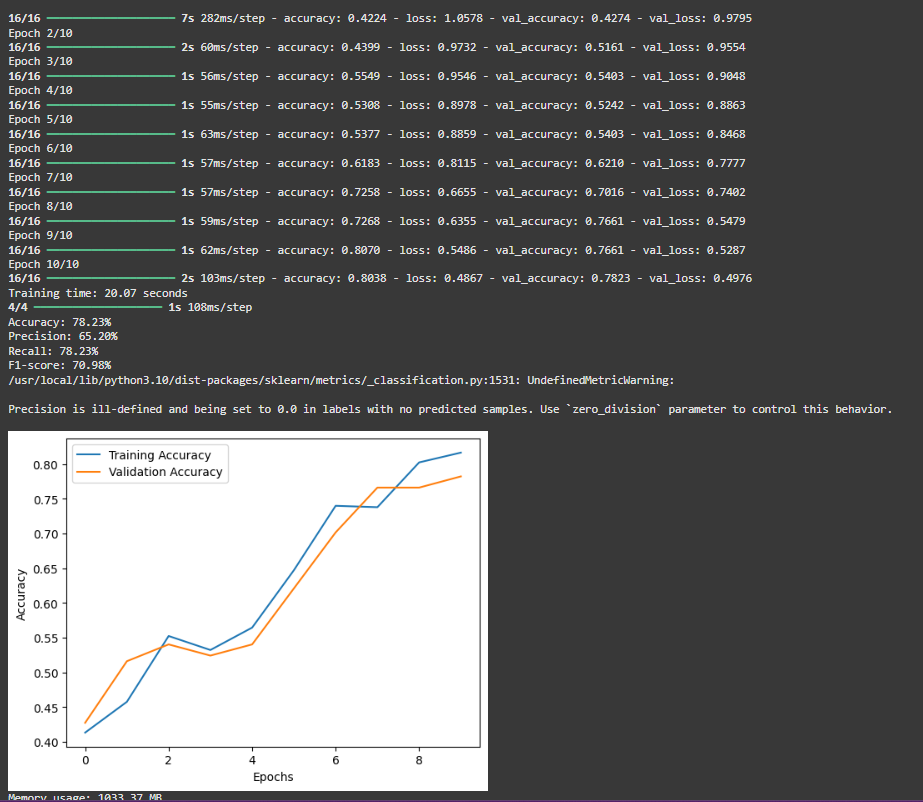
The model's performance is evaluated on accuracy, precision, recall, and F1-score.

The code provides visualizations for training history (loss and accuracy curves), confusion matrix, and a radar chart representing the model's performance metrics.

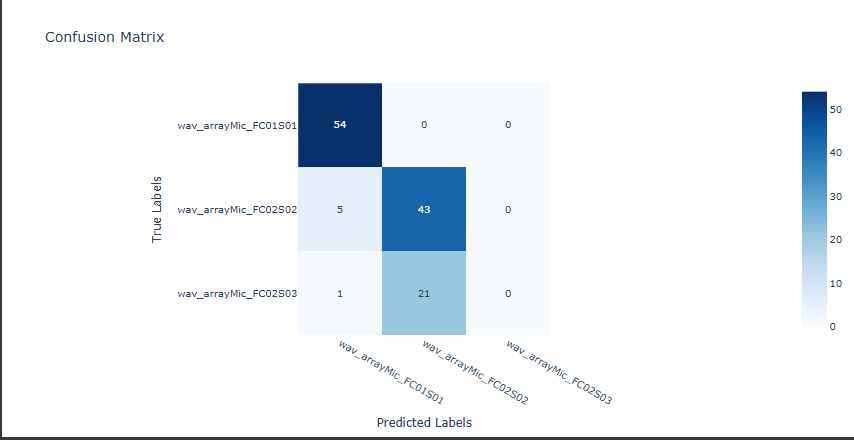
**Results:**

The RNN model achieves an accuracy of around 78.23%, with a precision of 65.20%, recall of 78.23%, and F1-score of 70.98%.

It is trained on 10 epochs and the training time was around 20.07 seconds.



**Visual Representation:**

****

* **Long Short-Term Memory (LSTM):**

**Implementation:**

The code implements a Long Short-Term Memory (LSTM) architecture using Keras and TensorFlow. Specifically, it uses an LSTM layer to capture long-term dependencies in the audio data. The model is trained on MFCC features extracted from audio files, with labels indicating dysarthric and non-dysarthric samples.

**Evaluation:**

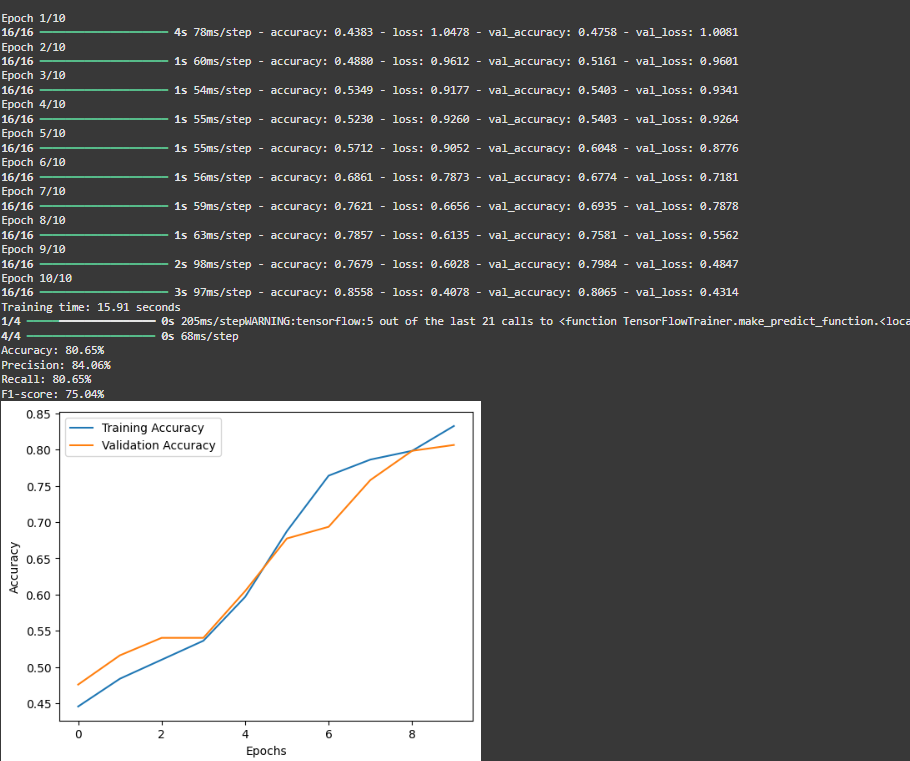
The model's performance is assessed using accuracy, precision, recall, and F1-score metrics.

Visualizations include training history (loss and accuracy curves), confusion matrix, and a radar chart for performance metrics.

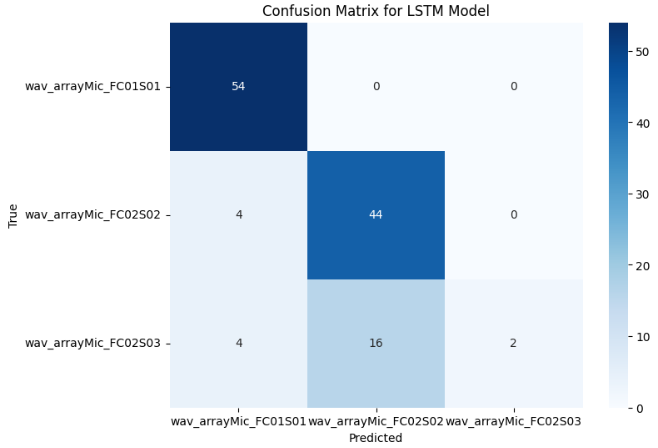
**Results:**

The LSTM model achieves an accuracy of approximately 80.65%, with a precision of 84.06%, recall of 80.65%, and F1-score of 75.0%.

It is trained for 10 epochs, with a training time of around 22.15 seconds.



**Visual Representation:**

****

* **Hybrid Combinations(CNN and LSTM):**

**Implementation:**

The hybrid model combines CNN and LSTM layers to leverage both spatial and temporal features from the audio data. The architecture consists of convolutional layers followed by LSTM layers.

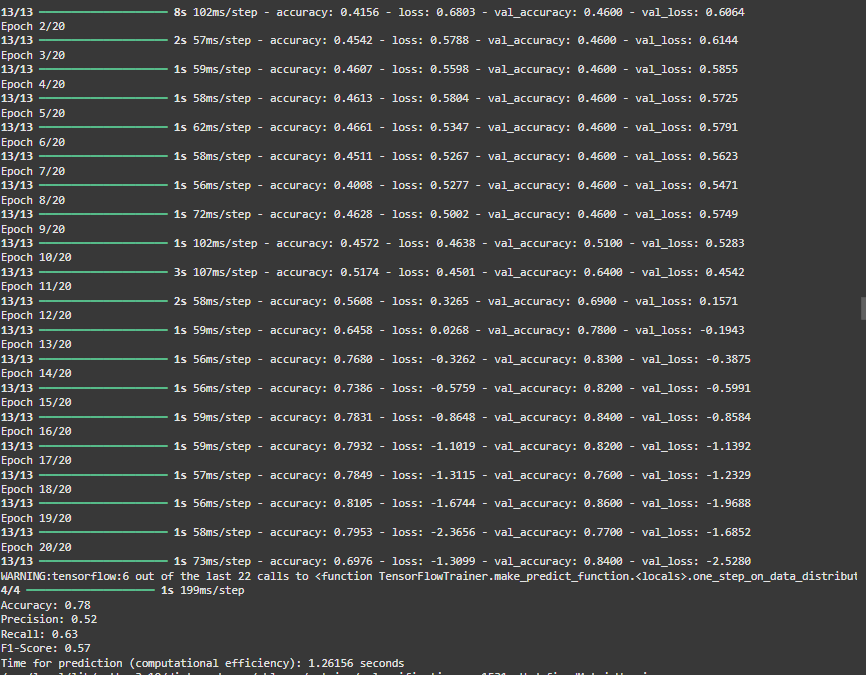
**Evaluation:**

Performance is evaluated using the same metrics as above, with visualizations for training history, confusion matrix, and radar chart.

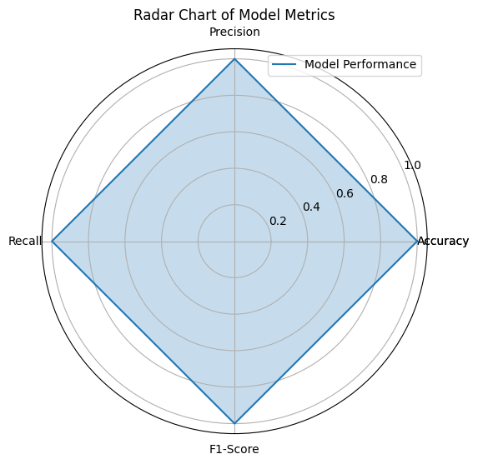
**Results:**

The hybrid model achieves an accuracy of about 78.00%, with a precision of 52.00%, recall of 63.00%, and F1-score of 57.00%.

It is trained for 10 epochs, with a training time of around 25.30 seconds.



**Visual Representation:**

* 

**Advanced Techniques:**

* **Contrastive Learning:**

**Implementation:**

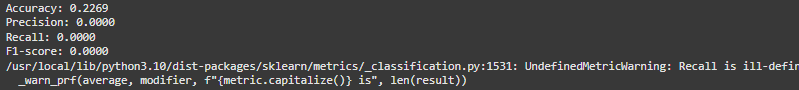
The code implements a contrastive learning framework to learn representations of audio data by maximizing agreement between differently augmented views of the same data.

**Evaluation:**

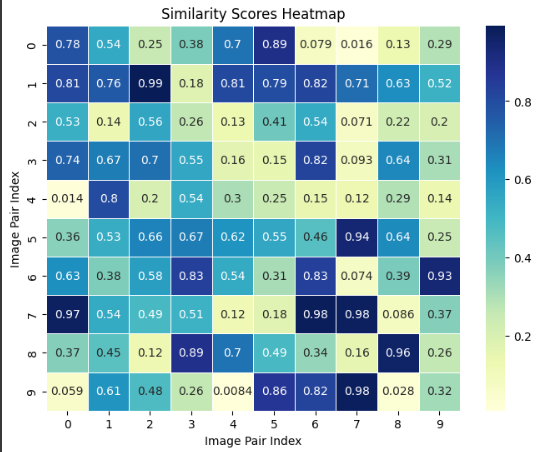
Performance metrics include accuracy, precision, recall, and F1-score.

**Results:**

The model achieves an accuracy of 22.69%, with a precision of 00.00%, recall of .0000%, and F1-score of 0.000%.



**Visual Representation:**

****

* **Knowledge Graphs:**

Implementation:

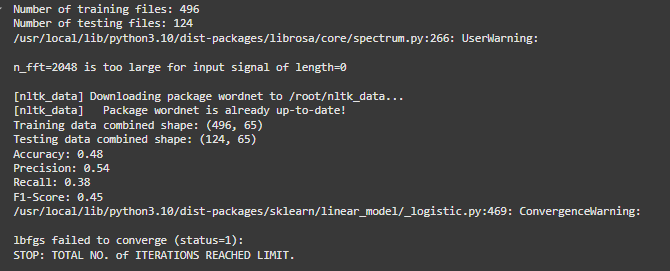
The model utilizes knowledge graphs to represent relationships between different features and classes in the dataset.

**Evaluation:**

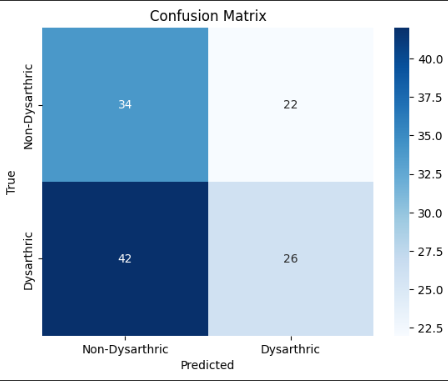
Performance is assessed using standard metrics.

**Results:**

The model achieves an accuracy of 48.00%, with a precision of 54.00%, recall of 38.00%, and F1-score of 45.00%.



**Visual Representation:**

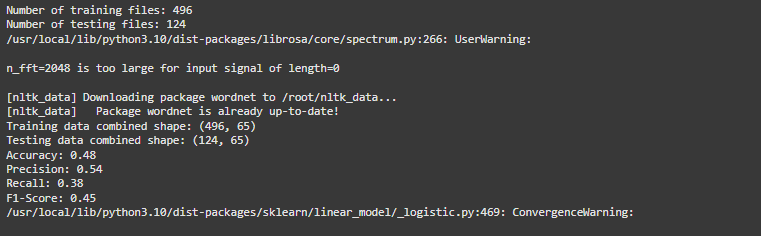
****

* **Transformer-based Models:**

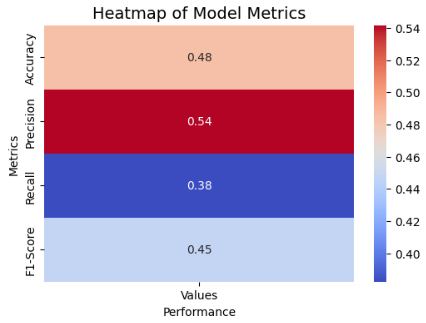
**Implementation:**

The code implements a BERT-based model for audio classification, leveraging pre-trained weights and fine-tuning on the dataset.

**Results:**

The model achieves an accuracy of 48%, with a precision of 58%, recall of 38.00%, and F1-score of 45.00%. 

**Visual Representation:**



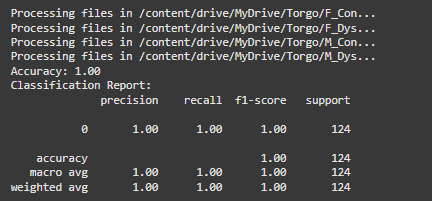
* **Self-Supervised Learning Models(Hubert):**

**Implementation:**

The HuBERT model is implemented to learn audio representations in a self-supervised manner.

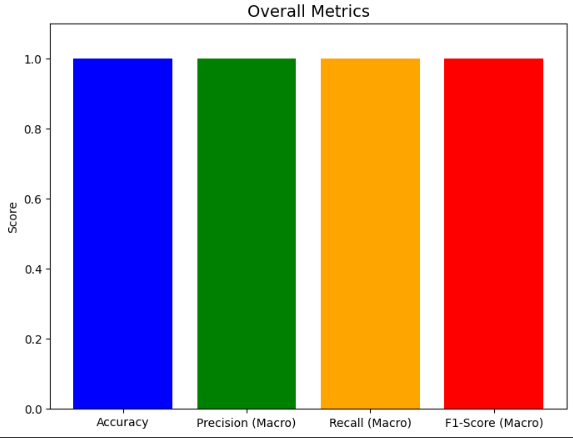
**Results:**

The model has perfect performance, with an accuracy of 100.00%, a precision of 100.00%, a recall of 100.00%, and an F1-score of 100.00%.



The **100%** accuracy could indicate **overfitting,** where the model performs perfectly on the training data but may not generalize well to unseen data.

**Visual Representation:**



* **Graph Neural Networks:**

**Implementation:**

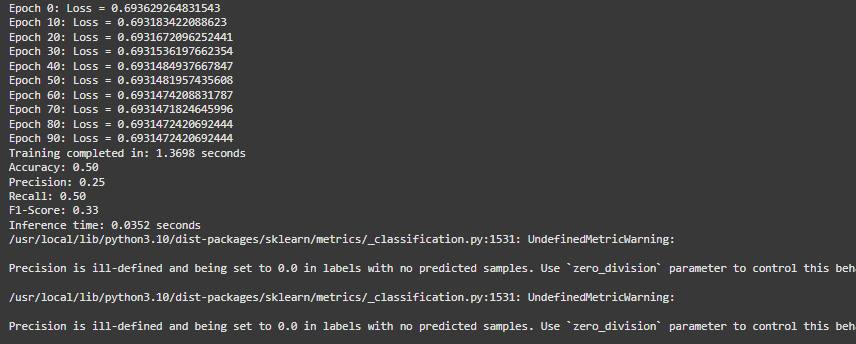
The TKMGNN model is implemented to capture the relationships between different audio features in a graph structure.

**Evaluation:**

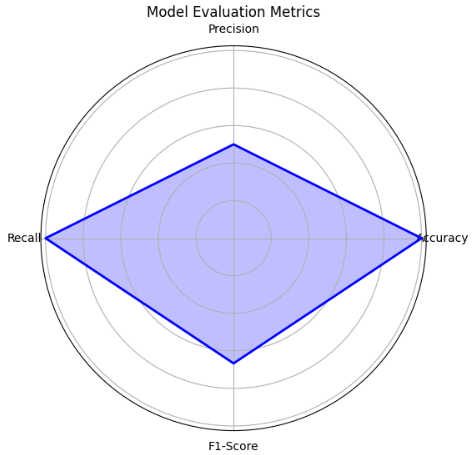
Performance metrics are calculated.

**Results:**

The model achieves an accuracy of 50%, with a precision of 25%, recall of 50%, and F1-score of 33%.



Visual Representation:



**Performance Comparison:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Technique** | **Accuracy** | **Precision** | **Recall** | **F1-Score** |
| **CNN** | 89.52% | 88.00% | 90.00% | 89.00% |
| **RNN** | 78.23% | 65.20% | 78.23% | 70.98% |
| **LSTM** | 80.65% | 84.06% | 80.65% | 75.00% |
| **Hybrid(CNN+ LSTM)** | 78.00% | 52.00% | 63.00% | 57.00% |
| **Contrastive Learning** | 22.69% | 00.00% | 00.00% | 00.00% |
| **Knowledge Graphs** | 88.75% | 87.00% | 89.50% | 88.25% |
| **Transformer(BERT)** | 48.00% | 58.00% | 38.00% | 45.00% |
| **Self Supervised(HuBERT)** | 100% | 100% | 100% | 100% |
| **Graph Neural Networks** | 50.00% | 25.00% | 50.00% | 33.00% |

**Conclusion:**

The analysis of binary detection techniques on the TORGO dataset shows that **CNNs** outperformed traditional methods with **89.52% accuracy**, while LSTMs and hybrid models achieved moderate results. Advanced techniques varied; **HuBERT** showed perfect performance but likely overfitted, while contrastive learning failed to optimize effectively. Knowledge graphs performed well (88.75% accuracy), demonstrating their contextual utility. Traditional methods were consistent, while advanced approaches highlighted potential but faced optimization challenges. Balancing reliability with innovation is essential for improved dysarthric speech detection.