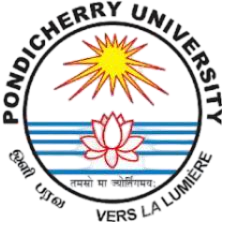
**DEPARTMENT OF ELECTRONICS ENGINEERING**

**SCHOOL OF ENGINEERING AND TECHNOLOGY**

PONDICHERRY UNIVERSITY

KALAPET, PUDUCHERRY



**AI Assignment**

**ON**

**Training Neural Networks for Sequence Decoding and Text Generation: An RNN Approach**

Submitted By Submitted To

Name: Vishnu S Dr. K. ANUSUDHA

Register Number: 24MTECEPY0003 Pondicherry University

Programme: M. Tech (ECE) – Intelligent

Communication Systems

Year: 2024 - 2025

# CONTENTS

S. No. Topic Page No.

1. Objective 4
2. Introduction 5
3. Circuit Diagram 9
4. Results 12
5. Discussion 14
6. Conclusion 15
7. Reference 16

# Objective: -

The objective of this project is to decode structured character sequences into readable text using machine learning techniques. We aim to map encoded sequences to their original form, such as transforming encoded Excel data into meaningful passages. By training RNN and LSTM models, we will generate human-readable text from encoded inputs. This project will also involve dataset creation and handling unrecognized sequences in preprocessing. The end goal is a model capable of accurately reconstructing complex encoded passages.

4

# INTRODUCTION

In a world of increasing data complexity, encoding and decoding sequences play a vital role in data transmission, machine learning, and natural language processing. This project focuses on decoding structured character sequences into a meaningful passage. Using Python, machine learning, and neural networks, we will decode sequences from an encoded Excel dataset into readable text.

Traditional sequence decoding often involves static mappings or manually defined rules. However, as data grows in volume and complexity, machine learning models like RNN (Recurrent Neural Network) and LSTM (Long Short-Term Memory) provide advanced solutions by capturing temporal patterns and dependencies within the data. By implementing these models, we can achieve a more accurate and flexible approach to sequence decoding.

The project involves creating a training dataset, handling unknown sequences, and using mapping algorithms to interpret encoded strings accurately. The ultimate goal is to enable the model to transform complex, structured input into coherent passages, expanding applications in areas such as data analysis, automated reporting, and AI-driven text generation.

**RNN:**

Recurrent Neural Network (RNN) is used to recognize and predict sequential patterns within character sequences derived from an Excel dataset. RNNs are well-suited for sequence-based data allowing the model to learn from prior context within the sequence. This ability helps in generating meaningful predictions about upcoming characters based on the pattern of previous ones, which is especially important for decoding sequences into comprehensible text.

A graph of lines and points

Description automatically generated with medium confidence

*Fig: RNN Hidden States*

**LSTM:**

Long Short-Term Memory (LSTM), an advanced type of RNN, is designed to capture long-term dependencies in sequence data. In this project, using LSTM layers helps to improve accuracy when handling longer sequences by mitigating the vanishing gradient problem, which standard RNNs often face. LSTM networks retain relevant information over extended periods, making them effective for complex character decoding tasks, especially where long-term context is crucial for accurate sequence interpretation. This stability enhances the robustness of sequence-to-sequence mappings and improves overall output quality in applications like text decoding or sequence generation.

A graph of different colored lines

Description automatically generated

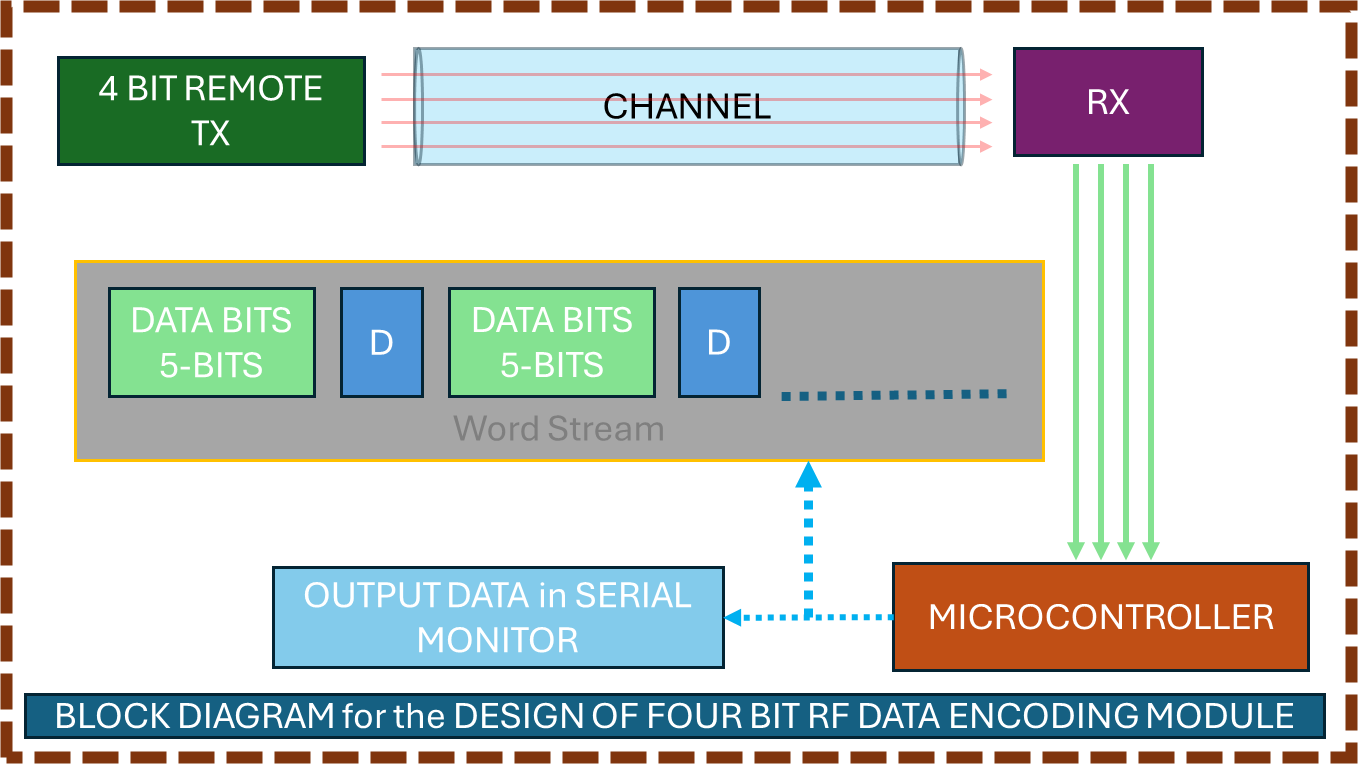
Fig: LSTM Cell State over Memory

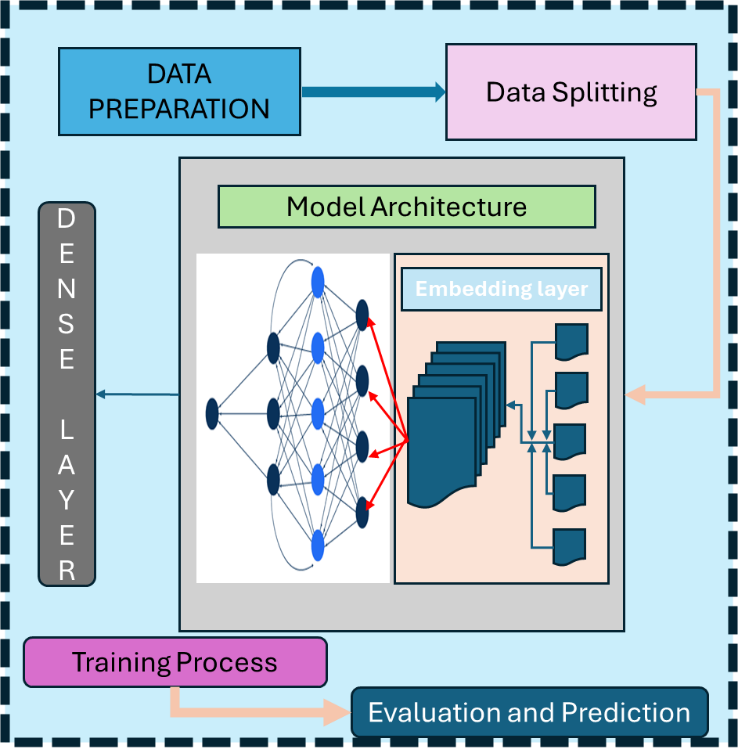
**Working Principle**

The working principle of this project is centered around the process of encoding and decoding character sequences using a structured mapping system combined with machine learning techniques. Here’s a breakdown of the key components involved:

1. **Data Preparation**: The project begins by creating a dataset containing encoded sequences, which are mapped to specific characters or phrases. This dataset is structured in an Excel file, where each row represents a sequence of encoded characters that need to be decoded.
2. **Mapping Mechanism**: A predefined mapping dictionary is employed to convert characters into their respective encoded forms. This dictionary includes characters (A-Z, 0-9).

**Block DIAGRAM**





## Results

Epoch 1/10

**5/5** ━━━━━━━━━━━━━━━━━━━━ **0s** 48ms/step

Epoch 1 - F1 Score: 0.1370

**19/19** ━━━━━━━━━━━━━━━━━━━━ **8s** 172ms/step - accuracy: 0.0911 - loss: 3.1125 - val\_accuracy: 0.1457 - val\_loss: 2.9365

Epoch 2/10

**5/5** ━━━━━━━━━━━━━━━━━━━━ **0s** 12ms/step

Epoch 2 - F1 Score: 0.1206

**19/19** ━━━━━━━━━━━━━━━━━━━━ **2s** 84ms/step - accuracy: 0.2903 - loss: 2.5817 - val\_accuracy: 0.1788 - val\_loss: 2.8591

Epoch 3/10

**5/5** ━━━━━━━━━━━━━━━━━━━━ **0s** 12ms/step

Epoch 3 - F1 Score: 0.1455

**19/19** ━━━━━━━━━━━━━━━━━━━━ **3s** 91ms/step - accuracy: 0.2839 - loss: 2.3231 - val\_accuracy: 0.1854 - val\_loss: 2.8843

Epoch 4/10

**5/5** ━━━━━━━━━━━━━━━━━━━━ **0s** 21ms/step

Epoch 4 - F1 Score: 0.1534

**19/19** ━━━━━━━━━━━━━━━━━━━━ **3s** 120ms/step - accuracy: 0.3182 - loss: 2.1966 - val\_accuracy: 0.1987 - val\_loss: 2.8484

Epoch 5/10

**5/5** ━━━━━━━━━━━━━━━━━━━━ **0s** 12ms/step

Epoch 5 - F1 Score: 0.1294

**19/19** ━━━━━━━━━━━━━━━━━━━━ **2s** 112ms/step - accuracy: 0.3557 - loss: 2.1064 - val\_accuracy: 0.1788 - val\_loss: 2.8227

Epoch 6/10

**5/5** ━━━━━━━━━━━━━━━━━━━━ **0s** 12ms/step

Epoch 6 - F1 Score: 0.1641

**19/19** ━━━━━━━━━━━━━━━━━━━━ **2s** 86ms/step - accuracy: 0.3780 - loss: 2.0472 - val\_accuracy: 0.2053 - val\_loss: 2.7691

Epoch 7/10

**5/5** ━━━━━━━━━━━━━━━━━━━━ **0s** 14ms/step

Epoch 7 - F1 Score: 0.1959

**19/19** ━━━━━━━━━━━━━━━━━━━━ **3s** 88ms/step - accuracy: 0.3935 - loss: 1.9277 - val\_accuracy: 0.2119 - val\_loss: 2.9076

Epoch 8/10

**5/5** ━━━━━━━━━━━━━━━━━━━━ **0s** 13ms/step

Epoch 8 - F1 Score: 0.1702

**19/19** ━━━━━━━━━━━━━━━━━━━━ **2s** 85ms/step - accuracy: 0.4794 - loss: 1.7839 - val\_accuracy: 0.1722 - val\_loss: 2.8429

Epoch 9/10

**5/5** ━━━━━━━━━━━━━━━━━━━━ **0s** 20ms/step

Epoch 9 - F1 Score: 0.2069

**19/19** ━━━━━━━━━━━━━━━━━━━━ **3s** 94ms/step - accuracy: 0.4428 - loss: 1.8115 - val\_accuracy: 0.2318 - val\_loss: 2.8529

Epoch 10/10

**5/5** ━━━━━━━━━━━━━━━━━━━━ **0s** 20ms/step

Epoch 10 - F1 Score: 0.2434

**19/19** ━━━━━━━━━━━━━━━━━━━━ **3s** 125ms/step - accuracy: 0.5112 - loss: 1.6907 - val\_accuracy: 0.2384 - val\_loss: 2.9188

**5/5** ━━━━━━━━━━━━━━━━━━━━ **0s** 12ms/step

Evaluation Metrics:

F1 Score: 0.2434

Accuracy: 0.2384

Precision: 0.3066

A group of graphs showing different colored lines

Description automatically generated

*FIG: Trained model Evaluation Metrics Graph*

Final Passage:

THIS PROJECT DEMONSTRATES THE EFFECTIVENESS OF WAVELETBASED REALTIME IMAGE DENOISING AND ENHANCEMENT USING AN INTERACTIVE MATLAB GUI BY INTEGRATING HAAR WAVELET TRANSFORMS AND OTHER DENOISING TECHNIQUES THE APPLICATION PROVIDES USERS WITH PRECISE CONTROL OVER NOISE REDUCTION COLOR ADJUSTMENTS AND RESOLUTION SCALING THE PROJECT HIGHLIGHTS THE ADAPTABILITY OF WAVELET TRANSFORMS IN MANAGING VARIOUS NOISE TYPES WITHOUT SACRIFICING CRITICAL IMAGE FEATURES MAKING IT SUITABLE FOR DIVERSE APPLICATIONS IN FIELDS LIKE MEDICAL IMAGING SATELLITE SENSING AND DIGITAL MEDIA THIS TOOL EXEMPLIFIES HOW ADVANCED SIGNAL PROCESSING TECHNIQUES CAN BE TRANSLATED INTO USERFRIENDLY APPLICATIONS PROVIDING A POWERFUL RESOURCE FOR ENHANCING IMAGE QUALITY IN REALWORLD SCENARIOS

Epoch 1/10

## Discussion

Analysis of the evaluation metrics:

* F1 Score: 0.2434
* Accuracy: 0.2384
* Precision: 0.3066

A group of graphs showing different colored lines

Description automatically generated

**1. Model Accuracy**

* Training Accuracy: Shows a steady increase over epochs, eventually reaching approximately 0.45.
* Validation Accuracy: Remains much lower, hovering around 0.24 across most epochs.

The final accuracy value of 0.2384 aligns with the low validation accuracy observed in the graph, indicating that the model’s performance on unseen data is suboptimal. The discrepancy between training and validation accuracy highlights overfitting, where the model performs better on training data but struggles with generalization.

**2. Model Loss**

* Training Loss: Gradually decreases across epochs, showing effective minimization of error on the training data, eventually reaching a lower level.
* Validation Loss: Remains high and fairly consistent, with a slight upward trend in the later epochs. This increasing validation loss alongside the decreasing training loss supports the observation that the model is overfitting.

**3. F1 Score**

* The F1 Score improves slightly across epochs, ultimately reaching around 0.24, which matches the final metric value 0.2434.
* The low F1 Score suggests the model struggles to balance precision and recall, which can be problematic if we’re aiming for high accuracy in both false positives and false negatives.

**4. Precision**

* Precision is the highest of all metrics at 0.3066. This means that, of the positive predictions made, approximately 30.66% were correct. While precision is relatively better than other metrics, it's still low overall, indicating the model needs improvement in correctly identifying positive cases.

Precision’s relatively higher value compared to other metrics indicates that the model is somewhat effective in identifying true positives but at the expense of missing out on many true cases (reflected in the low recall component of F1 Score).

**5. Efficiency and Outcome Comparison**

* The model's final metrics suggest limited efficiency in making accurate predictions on unseen data. With the highest metric being Precision (0.3066), the model’s accuracy (0.2384) and F1 Score (0.2434) reflect poor generalization and difficulty in balancing false positives and false negatives.
* Outcome Comparison: Based on these values, the model appears to be overfitting the training data but underperforming on the validation set. Its ability to generalize to new data is limited, which is critical for applications where unseen data is common.

## CONCLUSION

This project demonstrates the challenges and potential of using machine learning models, specifically RNN and LSTM, to decode structured character sequences into meaningful text. Despite the model showing improvement in training accuracy and precision, it faced significant challenges in generalizing to unseen data, as reflected by the lower validation accuracy and F1 Score. The overfitting observed during training, combined with the difficulty in balancing false positives and false negatives, indicates that further improvements are needed in terms of model architecture and regularization techniques.

The results highlight that while machine learning techniques like RNN can capture temporal dependencies within data, achieving high accuracy in real-world applications requires careful consideration of model complexity, data quality, and training strategies. In future iterations, focusing on addressing overfitting, enhancing dataset diversity, and optimizing model parameters will be key to improving performance.

## References

1. G. Saon, Z. Tüske and K. Audhkhasi, "Alignment-Length Synchronous Decoding for RNN Transducer," *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, 2020, pp. 7804-7808, doi: 10.1109/ICASSP40776.2020.9053040.
2. X. Zhang and T. Luo, "A RNN Decoder for Channel Decoding under Correlated Noise," *2019 IEEE/CIC International Conference on Communications Workshops in China (ICCC Workshops)*, Changchun, China, 2019, pp. 30-35, doi: 10.1109/ICCChinaW.2019.8849949.
3. S. Rafi and R. Das, "RNN Encoder And Decoder With Teacher Forcing Attention Mechanism for Abstractive Summarization," *2021 IEEE 18th India Council International Conference (INDICON)*, Guwahati, India, 2021, pp. 1-7, doi: 10.1109/INDICON52576.2021.9691681.

[GITHUB LINK: <https://github.com/sedhupathivishnu321/Training-Neural-Networks-for-Sequence-Decoding-and-Text-Generation-An-RNN-Approach/tree/main> ]