EMOTION DETECTION

Group-10

INTRODUCTION

Problem Statement :

The motivation behind this work is to develop a deep learning model for text-based emotion classification.This project aims to classify short-form text into one of six distinct emotional categories: ***joy***, ***sadness***, ***anger***, ***love***, ***fear***, and ***surprise***.

Objective :

The primary objective is to build, train, and compare several **Recurrent Neural Network (RNN)** architectures to identify the most effective model for this classification task.

Dataset :

The models were trained on the ***'Kaggle Emotion Dataset'***, which contains text entries, each labeled with an emotion(Number 1 to 6).

Tools & Technologies:

1. **Pandas & NumPy:** Used for loading, manipulating, and structuring the data.
2. **NLTK (Natural Language Toolkit):** Used for text preprocessing tasks like stopword removal and stemming.
3. **TensorFlow & Keras:** The main libraries used to build, train, and evaluate all the deep learning models (LSTM, GRU, etc.).
4. **Scikit-learn:** Used for splitting the data and generating final evaluation metrics (like the classification report and confusion matrix).
5. **Matplotlib & Seaborn:** Used to plot all the graphs and visualize the results.

Methodology

Our approach to this classification problem was centered on a two-part process: first, a **rigorous data preprocessing pipeline** to clean and structure the text, and second, the systematic **implementation and comparison of six different Recurrent Neural Network (RNN) architectures.**

Data Preprocessing :

To transform the raw text data into a format suitable for our deep learning models, we executed the following preprocessing steps:

1. **Text Cleaning:** The raw text was first cleaned to remove any residual HTML tags. All text was converted to lowercase, and all punctuation was removed. We then filtered out common English stopwords (e.g., "is", "the", "a") using the NLTK library to reduce noise. Finally, we applied Porter Stemming to reduce words to their root form (e.g., "loving" and "loved" both became "love").
2. **Tokenization:** The cleaned text was tokenized using the Keras Tokenizer, which built a vocabulary of all unique words. Each text was then converted from a string of words into a sequence of integers, where each integer represents a specific word in the vocabulary.
3. **Padding:** Since neural networks require inputs of a uniform length, all integer sequences were post-padded with zeros to ensure they all had a uniform length of 50 tokens
4. **Label Encoding:** The categorical text labels (e.g., "joy", "sadness") were converted into one-hot encoded vectors (e.g.,[0, 1, 0, 0, 0, 0]), making them suitable for training with a categorical_cross_entropy loss function.

Model Architectures and Training :

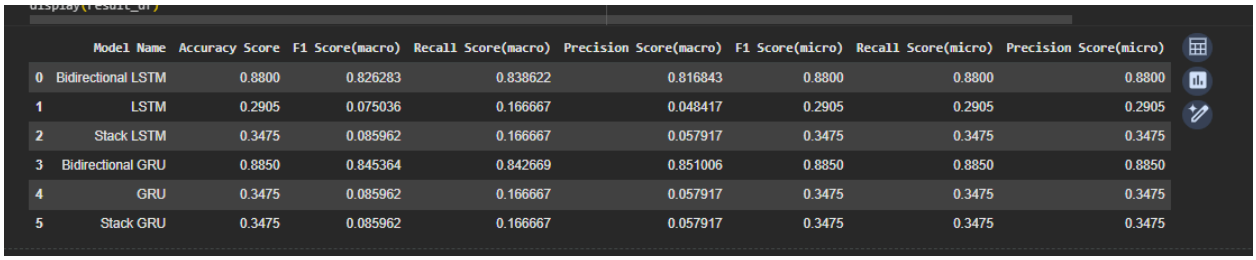
We trained 6 models :

1. **LSTM**
2. **GRU**
3. **Bidirectional LSTM**
4. **Bidirectional GRU**
5. **Stacked LSTM**
6. **Stacked GRU**

All models were built using the TensorFlow/Keras Sequential API. They shared a common structure: an Embedding layer as the input, the specific recurrent layers in the middle, and a final Dense layer with 6 units and a softmax activation function to output the probability for each of the six emotions.

All models were compiled with the adam optimizer and categorical_crossentropy loss. They were trained for 5 epochs with a batch_size of 32, using the validation set to monitor for overfitting.

We implemented and trained the following six models to compare their performance:



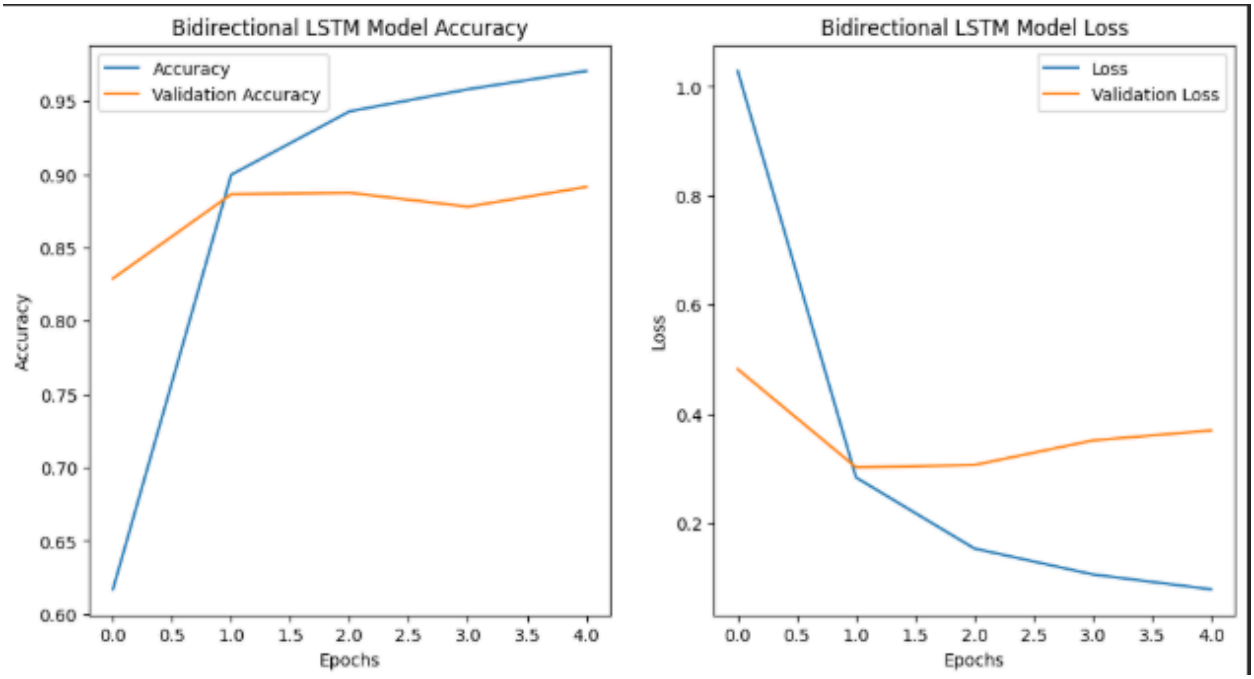
	Model Name	Accuracy Score	F1 Score(macro)	Recall Score(macro)	Precision Score(macro)	F1 Score(micro)	Recall Score(micro)	Precision Score(micro)
0	Bidirectional LSTM	0.8800	0.826283	0.838622	0.816843	0.8800	0.8800	0.8800
1	LSTM	0.2905	0.075036	0.166667	0.048417	0.2905	0.2905	0.2905
2	Stack LSTM	0.3475	0.085962	0.166667	0.057917	0.3475	0.3475	0.3475
3	Bidirectional GRU	0.8850	0.845364	0.842669	0.851006	0.8850	0.8850	0.8850
4	GRU	0.3475	0.085962	0.166667	0.057917	0.3475	0.3475	0.3475
5	Stack GRU	0.3475	0.085962	0.166667	0.057917	0.3475	0.3475	0.3475

Result :

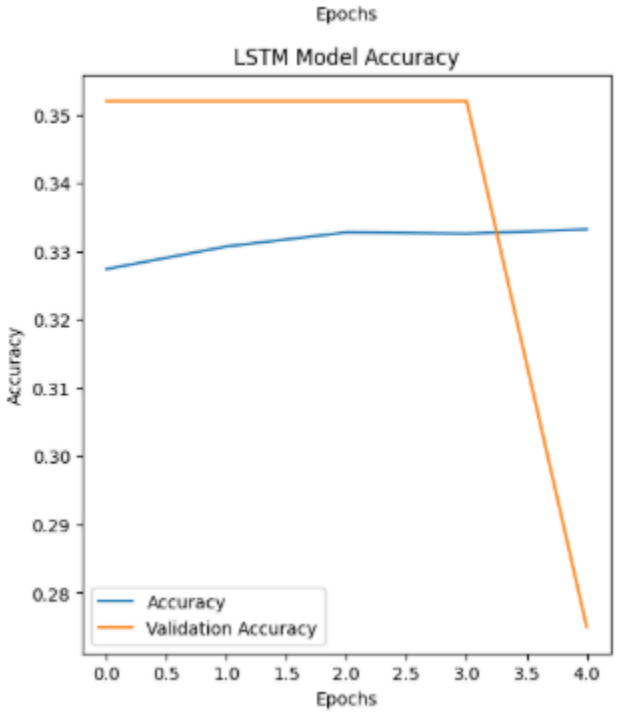
The **Bidirectional LSTM** and **Bidirectional GRU** showed the best accuracy over all other models.

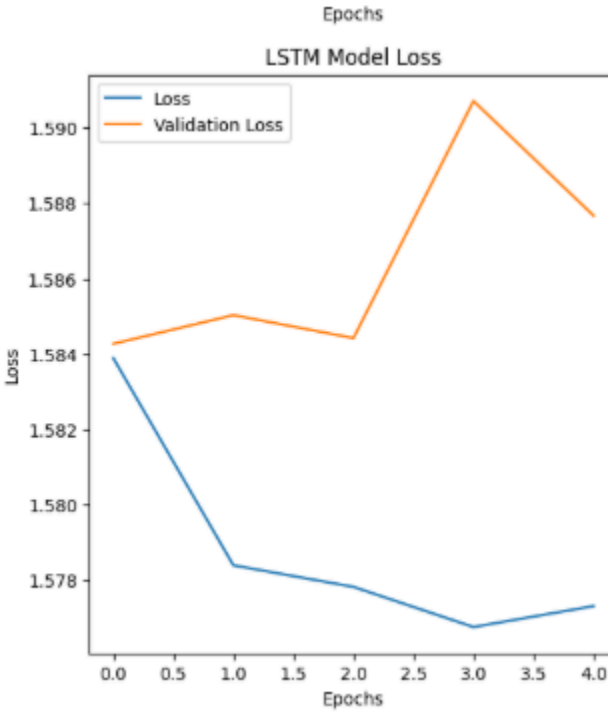
Graphs of Accuracy & Loss of the models :

1. Bidirectional LSTM

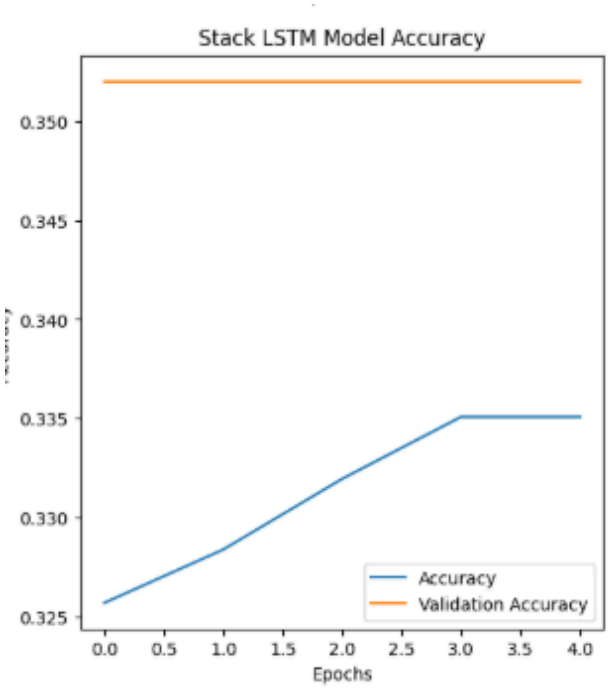


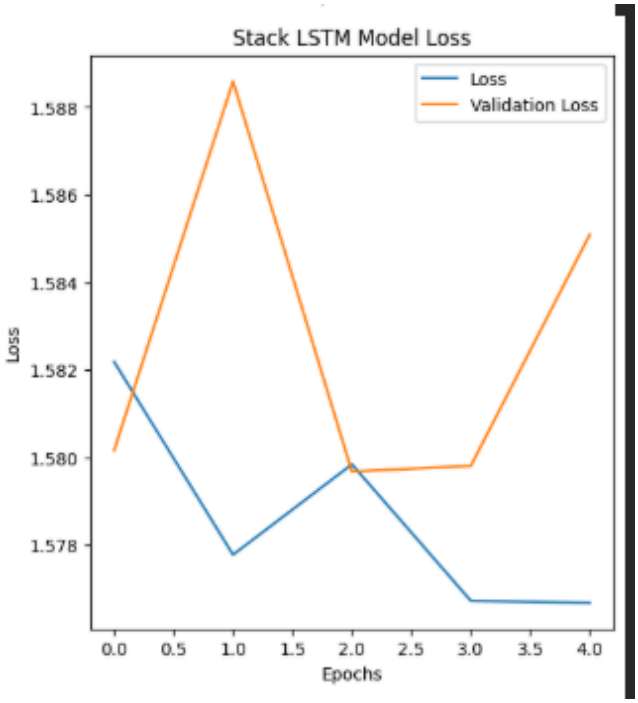
2. LSTM



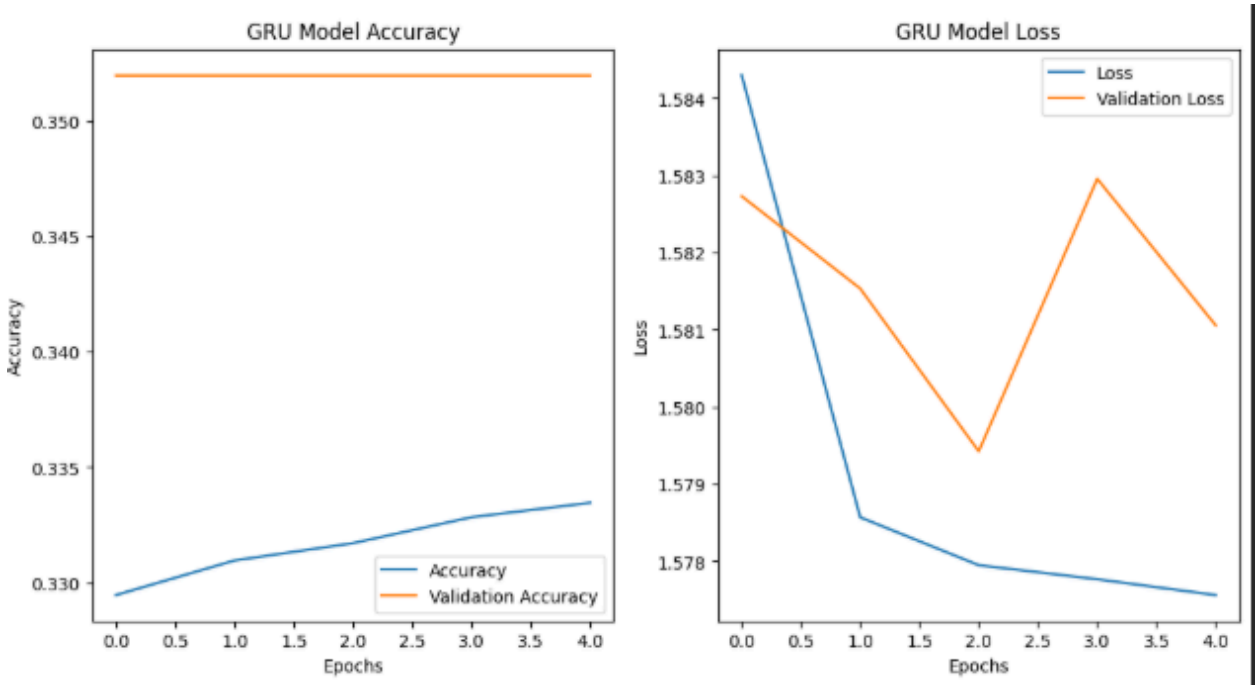


3. Stack LSTM

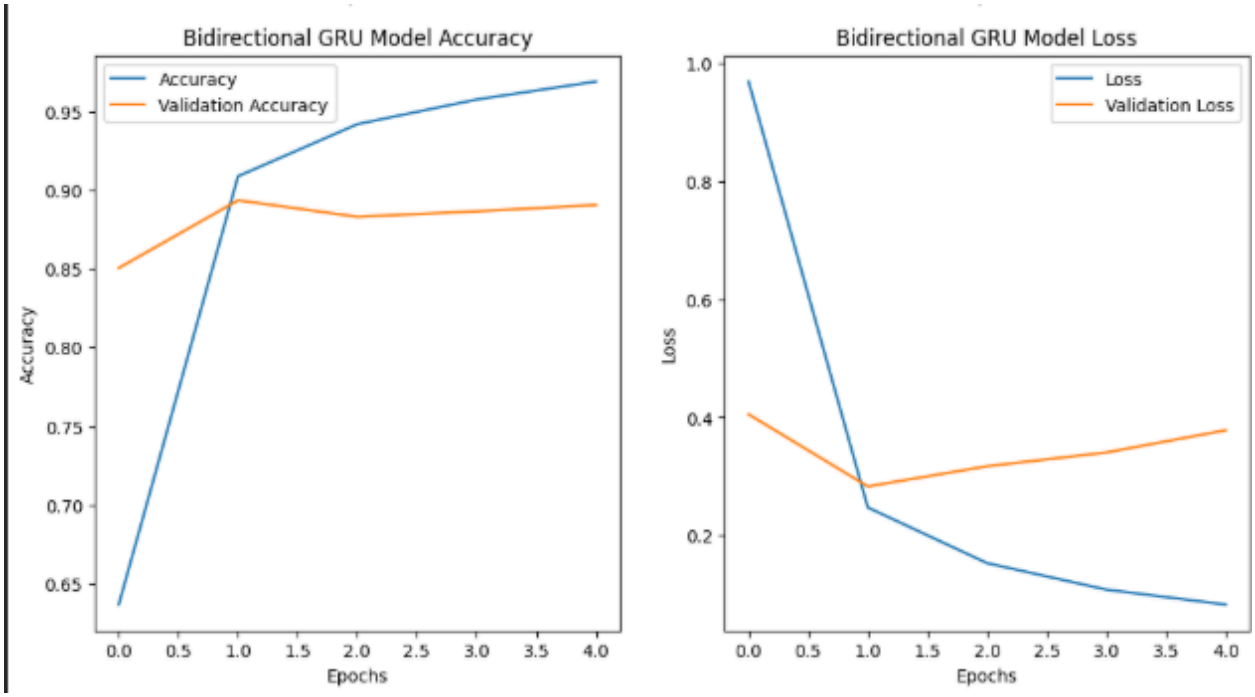




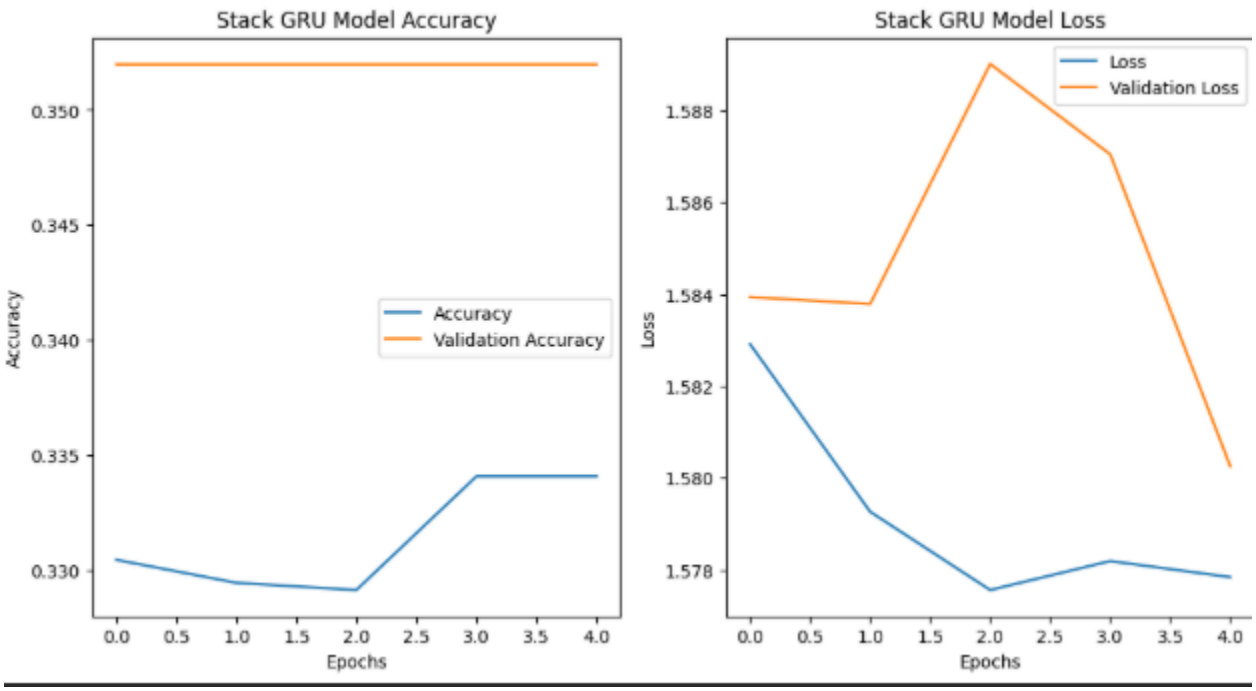
4. GRU



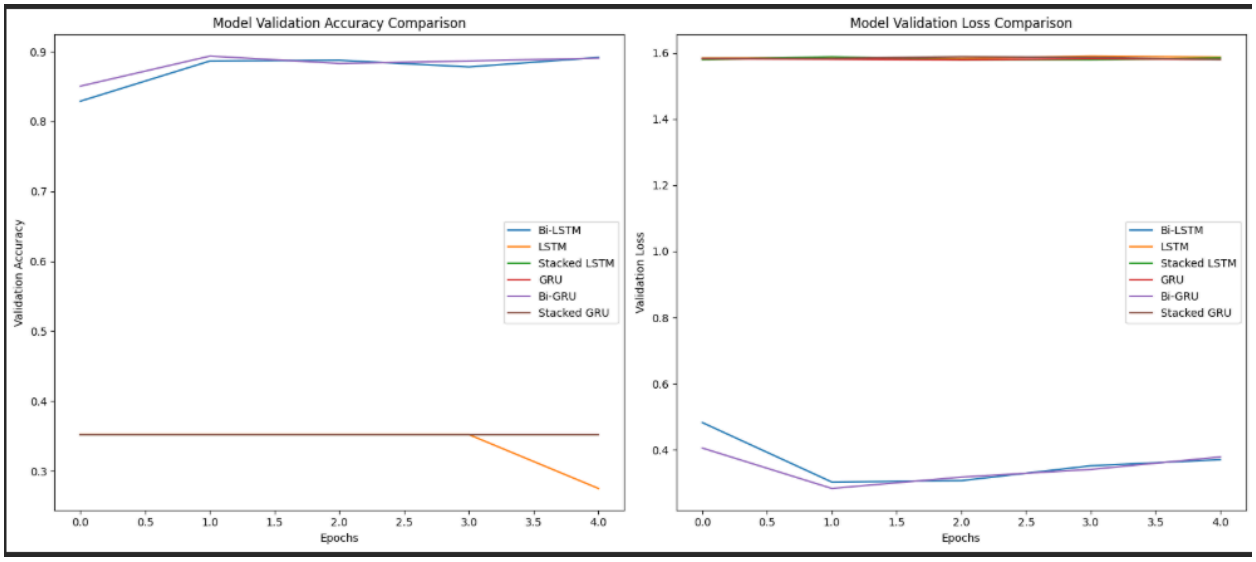
5. Bi-directional GRU

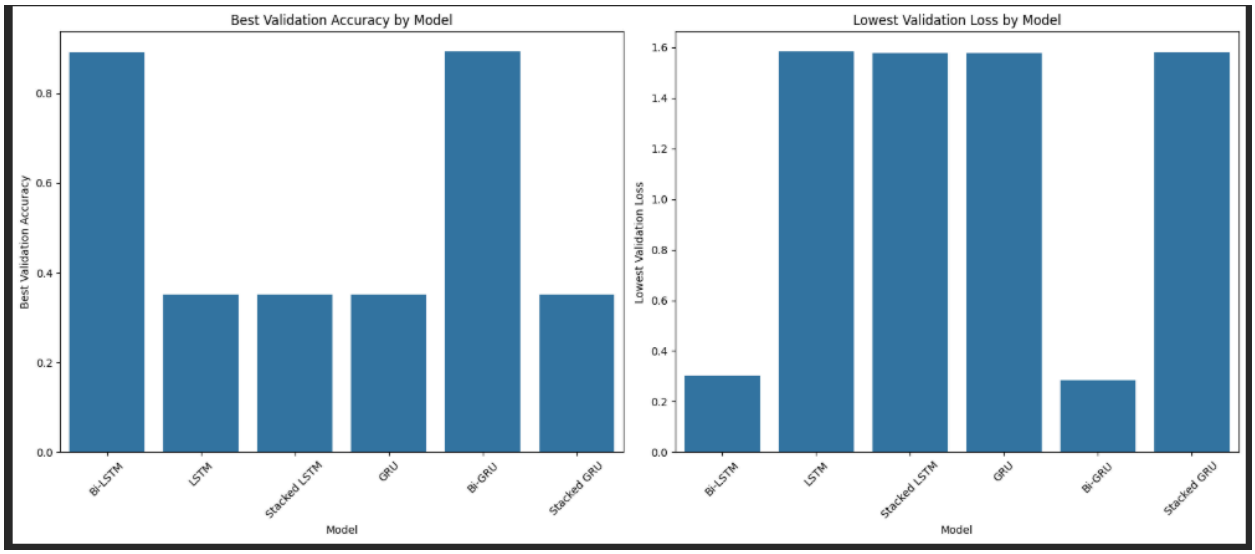


6. Stack GRU



Comparative Accuray and Loss across models :





Some example inputs and the output :

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step
Input:  I got the job! I am absolutely ecstatic!
Output:  joy


1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step
Input:  This is the third time my order has been wrong, I'm furious.
Output:  anger


1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step
Input:  I feel so warm and happy when I'm with my family.
Output:  joy


1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step
Input:  I've been feeling so down and empty all week.
Output:  sadness


1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
Input:  I heard a strange noise downstairs, I'm too scared to look.
Output:  fear


1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
Input:  I can't believe he would do that to me.
Output:  joy


1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step
Input:  The report is due by 5 PM on Friday.
Output:  joy


1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
Input:  I finally finished that difficult project, what a relief.
Output:  joy


1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step
Input:  I hope my presentation goes well tomorrow.
Output:  joy


1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step
Input:  He quit? Without any notice? I'm stunned.
Output:  surprise


1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step
Input:  Oh, great. Another meeting. Just what I needed today.
Output:  anger


1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step
Input:  I'll miss you so much, but I'm happy you're following your dreams.
Output:  sadness
```