**Shri Ramdeobaba College of Engineering & Management, Nagpur**

**Department of Electronics and Computer Science Engineering**



**Spam Email Detection Model using Deep Learning**

**Group ID: DLP-4**

1. **Ojas Rai Rollno 10**
2. **Kshitij Halmare Rollno 11**

**GitHub:**[**https://github.com/Kshitij-Halmare/Spam-Email-Detection**](https://github.com/Kshitij-Halmare/Spam-Email-Detection)

**Abstract**

This project presents a spam email detection system using a Bidirectional LSTM (BiLSTM) neural network architecture. By processing preprocessed text sequences through word embeddings and recurrent layers, the model achieves an accuracy of 97% on test data. It demonstrates robust performance with a high AUC-ROC score (0.9947), showing strong generalization to unseen data. The system is also tested on custom examples to validate real-world applicability.

**Introduction**

With the exponential rise in email communication, spam emails pose a significant threat by clogging inboxes, spreading phishing links, and compromising user data. Traditional keyword-based filters are no longer sufficient due to the evolving nature of spam. This project aims to leverage deep learning, specifically a BiLSTM model, to accurately detect spam emails.

**Dataset**

The dataset used is the widely known SMS Spam Collection dataset from UCI, comprising 5,572 messages labeled as "ham" (legitimate) or "spam". Preprocessing steps included tokenization, lowercasing, stopword removal, stemming, and padding for uniform sequence length.

**"ham"** – legitimate, non-spam messages (approx. 86.6%)

**"spam"** – unsolicited messages typically sent for advertising or phishing (approx. 13.4%)

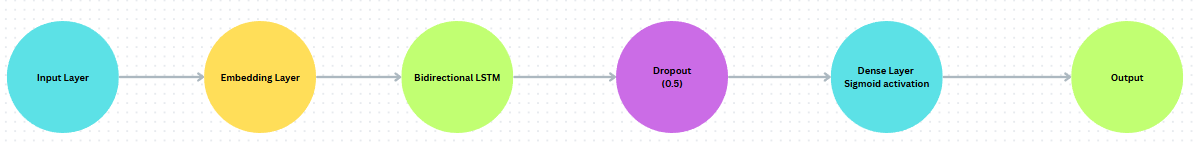
**Preprocessing Steps:**

* Lowercasing all text.
* Removal of punctuation and special characters.
* Tokenization using Keras’ Tokenizer.
* Stopword removal using NLTK.
* Stemming using the Porter Stemmer to reduce words to root forms.
* Padding sequences to a fixed length (e.g., 50 tokens) to ensure uniform input to the model

**Methodology**

* Preprocessing: Tokenization, stemming (Porter), padding sequences to max length.
* Model: Embedding Layer → Bidirectional LSTM → Dropout → Dense (Sigmoid).
* Training Details: Binary cross-entropy loss, Adam optimizer, early stopping and learning rate reduction.
* Evaluation Metrics: Accuracy, Precision, Recall, F1-score, Confusion Matrix, AUC-ROC.

**Model Architecture**

****

* Embedding Layer: Converts each token into a dense 100-dimensional vector. Vocabulary size determined by the top 5000 most frequent words.
* Bidirectional LSTM Layer: Allows the model to learn patterns considering context from both past and future words.
* Dropout Layer: Applied with a rate of 0.5 to prevent overfitting by randomly disabling 50% of neurons during training.
* Dense Layer (Sigmoid): Final binary output layer producing a probability between 0 and 1 for the "spam" class.

#### **Training Details:**

* Optimizer: Adam (adaptive learning rate)
* Loss Function: Binary Cross-Entropy (with optional class weighting to handle imbalance)
* Metrics: Accuracy, Precision, Recall, F1-score
* Callbacks:  
  + EarlyStopping: To prevent overfitting by halting training when validation loss stops improving.
  + ReduceLROnPlateau: Automatically reduces the learning rate when the model hits a plateau.

#### **Data Split:**

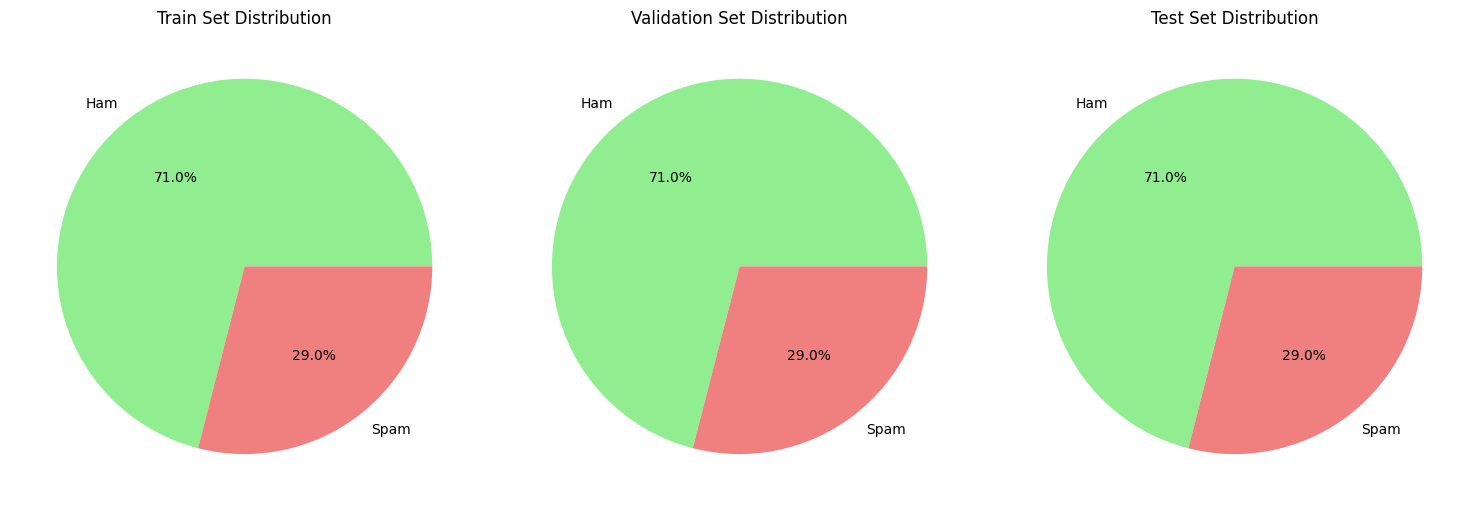
* Training Set: 80% of the data
* Validation Set: 10% of the training set
* Test Set: 10% held out for final evaluation

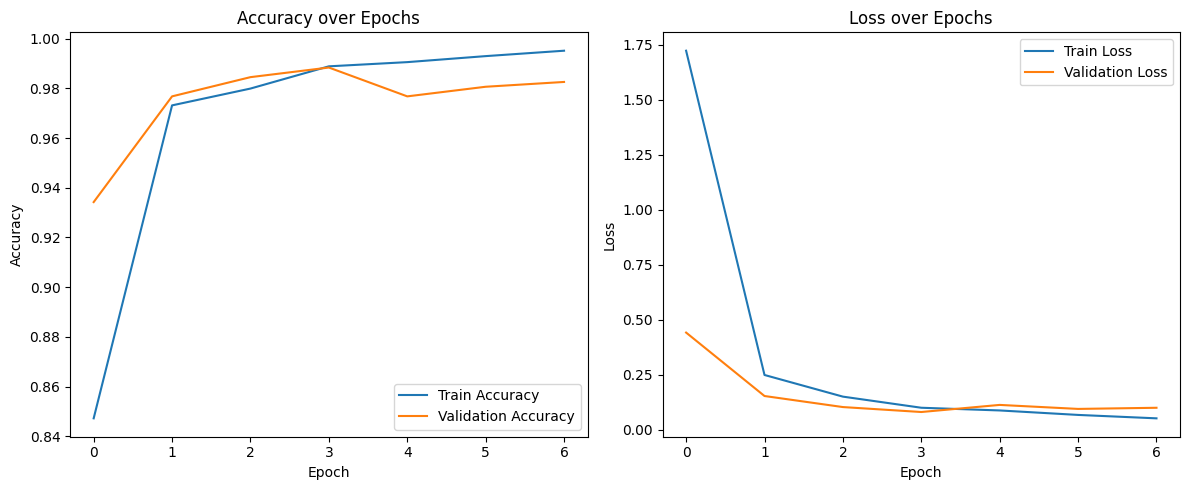
**Result:**

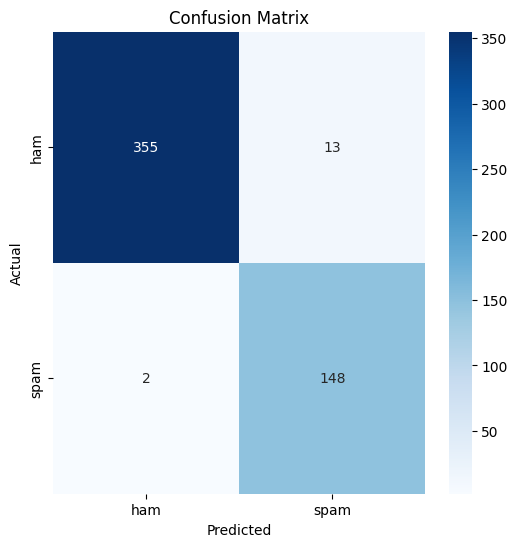
* Training Accuracy: ~99.13%
* Validation Accuracy: ~98.26%

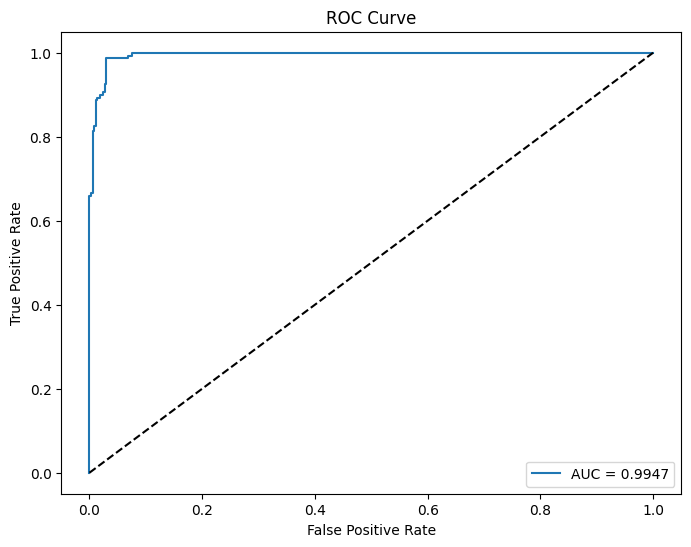
Classification Metrics:

| Class | Precision | Recall | F1-Score | Support |
| --- | --- | --- | --- | --- |
| Ham | 0.99 | 0.96 | 0.98 | 368 |
| Spam | 0.92 | 0.99 | 0.95 | 150 |

****

****

****

****

**AUC-ROC Score: 0.9947**

**Contribution:**

**Kshitij Halmare**

* Led the data preprocessing pipeline, including text cleaning, tokenization, stemming, and padding.
* Designed and implemented the Bidirectional LSTM architecture using Keras.
* Handled model training, tuning hyperparameters, and implementing callbacks for optimization (e.g., EarlyStopping, ReduceLROnPlateau).
* Created the model evaluation pipeline, including classification metrics, confusion matrix, and AUC-ROC analysis.

**Ojas Rai**

* Focused on dataset exploration, visualizations, and statistical analysis of spam vs. ham messages.
* Conducted testing on custom email examples and analyzed misclassifications.
* Drafted and formatted the project report, including sections on methodology, results, and discussion.
* Assisted with preparing the architecture diagram, formatting references, and writing the conclusion.

**References**

1)Hochreiter, S., & Schmidhuber, J. (1997).

*Long Short-Term Memory*. Neural Computation, 9(8), 1735–1780.

2) Graves, A., & Schmidhuber, J. (2005).

*Framewise phoneme classification with bidirectional LSTM networks*.

In Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN).

3) Almeida, T.A., Hidalgo, J.M.G., & Yamakami, A. (2011).

*Contribution to the study of SMS spam filtering: new collection and results*.

In Proceedings of the 11th ACM Symposium on Document Engineering (DocEng).