 A green and black sign with text

AI-generated content may be incorrect.

**Module Code & Module Title**

**6CS012 – Artificial Intelligence and Machine Learning**

**SMS Spam & Ham Detection using RNN, LSTM, and Pretrained Word2Vec Embeddings**

**Student ID : 2329200**

**Student Name: Aashan Bajgain**

**College ID : np03cs4a220002**

**Tutor : Mr. Siman Giri**

**Module Leader : Mr. Siman Giri**

**Word count : 1451**

**Submitted on : 2025/05/13**

# Abstract

In this fast-paced digital times, one might say being unsolicited has gained a bad name on one particular platform: SMS messaging. The project intends to implement a deep learning pipeline for the binary text classification: spam vs ham. There are three different model architectures that the team tries to tackle: Simple RNN, LSTM, and LSTM with pretrained Word2Vec embeddings from the GloVe corpus. The data are then prepared for the input through tokenization and percentile-based sequence padding. Each model was trained and then evaluated for classification accuracy, losses, confusion matrices, and F1-scores. A GUI based on Gradio was set throughout the project for making predictions in real-time. Our experimental results confirm that LSTM with pretrained embeddings is a clear winner over baseline models. Since the project brings to the fore the embedding layer capability for learning semantic context, it provides a scalable and efficient method for real-time SMS spam detection.

Table of Contents

[1.Introduction 4](#_Toc197971014)

[2. Dataset Description 5](#_Toc197971015)

[2.1. Dataset Statistics 5](#_Toc197971016)

[2.2. Preprocessing Steps 5](#_Toc197971017)

[3. Methodology 6](#_Toc197971018)

[3.1. Preprocessing 6](#_Toc197971019)

[3.2. Model 1: Simple RNN with Trainable Embedding 6](#_Toc197971020)

[3.3. Model 2: LSTM with Trainable Embedding 6](#_Toc197971021)

[3.4. Model 3: LSTM with Pretrained Word2Vec Embeddings 6](#_Toc197971022)

[4. Training and Evaluation 7](#_Toc197971023)

[4.1. Training Parameters 7](#_Toc197971024)

[4.2. Evaluation Metrics 7](#_Toc197971025)

[4.3. Results Summary 7](#_Toc197971026)

[5. GUI for Real-Time Prediction 8](#_Toc197971027)

[6. Model Comparison & Visualizations 8](#_Toc197971028)

[7. Discussions 9](#_Toc197971029)

[7.1. RNN vs. LSTM 9](#_Toc197971030)

[7.2. Word2Vec Benefits 9](#_Toc197971031)

[7.3. Computational Efficiency 9](#_Toc197971032)

[8. Conclusion and Future Work 10](#_Toc197971033)

[9. References 11](#_Toc197971034)

# 1.Introduction

Since SMS is currently one of the prominent communication mediums, spam messages of threatening nature-phishing and malware-have risen, posing an intrusion. Hence, spam detection is the primary focus of securing digital communication. Traditional techniques followed rule-based filtering and keyword matching but toward advancement, with deep learning, in particular, deep learning coupled with NLP techniques, have developed more prominent and precise models for spam detection.

Text classification is a core NLP problem with a multitude of applications ranging from sentiment analysis to spam detection. Among various models available, RNNs, and LSTMs are especially useful in processing sequential data. Nonetheless, training everything from scratch disregards the rich semantics of words. Using pretrained embeddings such as Word2Vec has the advantage of providing context without needing to undergo time-consuming training.

In this project, we have compared and applied three models:

* Simple RNN with trainable embedding layer.
* LSTM with trainable embedding layer.
* LSTM with pretrained GloVe (Word2Vec) embeddings.

All the models were verified by using accuracy, precision, recall, F1-score, and confusion matrix to measure its performance. A Gradio-powered interface also allows real-time testing of the final model, enhancing usability and deployment readiness.

# 2. Dataset Description

The data collection for this project is the UCI SMS Spam Collection Dataset, a highly sought-after corpus for spam detection problems. It consists of 5,574 SMS messages labeled as either "spam" or "ham" (not spam). The data collection has two principal columns:

* v1: Label that is 'ham' or 'spam'
* v2: Unprocessed SMS message

## 2.1. Dataset Statistics

* Total Messages: 5,574
* Spam Messages: 747
* Ham Messages: 4,827

Given the binary nature of the classification task, we map 'ham' to 0 and 'spam' to 1.

## 2.2. Preprocessing Steps

* Making the text lowercase
* Remove the punctuation, URLs, special characters, and numbers
* Tokenize the text using Keras Tokenizer
* Pad the sequences using the 95th percentile padding

The above pipeline leaves the standardized input with required semantic structures in place.

# 3. Methodology

## 3.1. Preprocessing

To preprocess the data, we preprocessed the text, ignoring characters such as stopwords and punctuation by re and nltk libraries of Python. We turned each of messages into an integer list by tokenizing the text using Keras’ Tokenizer. The indexes of word frequency are integers. After tokenisation, we padded the sequences to a common input size for all models, and we computed the 95th percentile of sequence length.

## 3.2. Model 1: Simple RNN with Trainable Embedding

The first RNN-based model we propose is the simple RNN model with an embedding layer that is trained from scratch. The architecture includes:

* Embedding layer (trainable)
* SimpleRNN with 128 units
* 64 node Dense layer Activated with ReLU
* Dropout layer
* Final Dense(1) output layer with sigmoid activation

## 3.3. Model 2: LSTM with Trainable Embedding

Taking the RNN model and plugging in LSTM instead of the recurrence layer was necessary to introduce longer dependencies. The rest of the architecture is the same; the only difference is that LSTM is going to be better able to learn characteristics of longer messages with context and across many repetitions due to less vanishing gradient issues.

## 3.4. Model 3: LSTM with Pretrained Word2Vec Embeddings

For this model, we used GloVe's 50-dimensional embeddings. We used gensim to download glove-wiki-gigaword-50, then, we made an embedding matrix (that referenced each word in our vocabulary, to the vectors given by the GloVe model), initializing missing words in the vocabulary with zeros.

The embedding layer, in this model, was frozen (trainable=False) so that its pretrained semantics would not change throughout training. The remaining architecture is identical to Model 2.

# 4. Training and Evaluation

Each model was trained with the Adam optimizer, using binary\_crossentropy as the loss function and accuracy as the evaluation metric. Early stopping was used to avoid overfitting.

## 4.1. Training Parameters

* Epochs: 10
* Batch Size: 64
* Validation Split: 10%

## 4.2. Evaluation Metrics

We evaluated models using:

* Accuracy
* Confusion Matrix
* Precision, Recall, F1-Score

## 4.3. Results Summary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **Recall** | **F1 Score** |
| Simple RNN | ~94.6% | 0.89 | 0.88 | 0.88 |
| LSTM | ~96.2% | 0.92 | 0.91 | 0.91 |
| LSTM+GloVe | ~97.1% | 0.94 | 0.93 | 0.93 |

The GloVe-enhanced LSTM out performed both of the other alternatives statistically with the main advantage being recall and F1-score, which are important measures in reducing false negatives in the detection of spam.

# 5. GUI for Real-Time Prediction

To add interactivity and accessibility to our model we created a Gradio-based GUI. Users can now use a web interface to enter SMS message input and receive an immediate classification output (Spam or Ham), along with confidence scores.

**Key Features:**

* Textbox input for messages
* Output label (Spam/Ham)
* Real-time response output probability

This design closes the loop on our model development to application, proving that the model can be successfully deployed.

# 6. Model Comparison & Visualizations

We plotted the training and validation accuracy and loss for each model, and it is clear from the plots that all LSTM models converge more quickly and generalize much better to unseen data compared to the Simple RNN. Model 3 had the lowest validation loss that we recorded, which indicated it had the lowest error relative to the other models.

Lastly, we also presented confusion matrices and classification reports to provide another illustration of the numerical results and from the heatmap, we saw fewer examples of miscategorization in the GloVe model.

# 7. Discussions

## 7.1. RNN vs. LSTM

Based on the experimental observations, it is clear that the LSTM architecture is superior to SimpleRNN in terms of retaining contextual memory across sequences. It effectively captured patterns for when the length of messages exceeded what most RNN memory could handle.

## 7.2. Word2Vec Benefits

Using pretrained embeddings enhanced the training speed and the model's accuracy. The pretrained embeddings embedded semantic knowledge that was not retrievable from limited training data.

## 7.3. Computational Efficiency

Each model was trained on Jupyter notebook, and the GloVe model took marginally longer to train due to the initialization of the embedding matrix, but the accuracy gains were worth this expense. Memory consumption was similar across models.

# 8. Conclusion and Future Work

In this project we successfully applied and evaluate three deep learning models on SMS spam detection. The results were promising and showed that LSTM with GloVe embeddings performed the best with over 97% accuracy and very good precision and recall. Using pre-trained embedding was substantial benefit.

For the future we want to:

* Use Bidirectional LSTM
* Use transformers such as BERT that can provide more context
* Deploy the model using Streamlit or Flask and create a web application
* Expanding the system to multilingual dataset

This project has laid the groundwork for new and effective real-time spam detection tools that utilize effective deep learning methods.

# 9. References

Chollet. (2025). *Keras Documentation.* Retrieved from Keras: https://keras.io/

Gradio. (2025). *Gradio Documentation*. Retrieved from Gradio: https://www.gradio.app/docs

Mikolov, T. (2013). *Distributed Representations of Words and Phrases and their Compositionality*. Retrieved from Arxiv: https://arxiv.org/abs/1310.4546

Řehůřek, R. (2024). *Gensim Topic Modeling for Humans*. Retrieved from Radim Řehůřek: https://radimrehurek.com/gensim/auto\_examples/index.html

UCI Machine Learning Repository. (2018). *SMS Spam Collection*. Retrieved from UCI ICS Archive: https://archive.ics.uci.edu/dataset/228/sms+spam+collection