IJRAR.ORG E-ISSN: 2348-1269, P-ISSN: 2349-5138

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS (IJRAR) | IJRAR.ORG

An International Open Access, Peer-reviewed, Refereed Journal

Spam Watch: Real-Time SMS Spam Detection Using LSTM and Hugging Face Hosting

[1]Yakshith K D

[1]Student

[1]Artificial intelligence and machine learning

[1]Mangalore Institute of technology and Engineering, Moodbidri, Karnataka

Abstract**—** In the digital communication era, spam messages pose a significant threat to privacy and productivity. Traditional rule-based or keyword-driven spam filters often struggle with high false positives and evolving spam tactics. To solve this, we present SpamWatch, real-time SMS spam detection system that makes use of networks with Long Short-Term Memory (LSTM), a sophisticated deep learning architecture that works well with sequential text data.

Model has been trained on a preprocessed and vectorized dataset of SMS messages, effectively learning the contextual patterns that distinguish spam from legitimate text. Our system leverages tokenization, embedding layers, and recurrent structures to extract semantic features and classify messages with high accuracy.

To enhance accessibility and practical utility, SpamWatch is deployed on the Hugging Face Inference API, providing a simple, Users can enter messages into an interactive web interface and get immediate feedback on their spam classification. The deployment integrates Gradio for frontend interaction and LSTM-based backend logic, offering real-time inference without the need for local execution.

Comprehensive evaluations demonstrate that the model achieves strong performance metrics, encompassing high precision, accuracy, and recall. This project contributes to growing field of AI-driven text classification and provides a scalable, user-friendly solution for combating SMS spam in real-world scenarios.

Keywords**—** *LSTM, SMS Spam Detection, Deep Learning, Natural Language Processing, Real-Time Text Classification, Hugging Face Deployment, Gradio Interface, Neural Networks, Recurrent Neural Networks, AI Text Filtering*

Introduction

The rapid growth of mobile communication has brought convenience and connectivity to millions worldwide. However, this technological advancement has also led to an increase in unwanted, potentially harmful messages, described as spam. These spam messages not only clutter users' inboxes but may also contain malicious links, phishing attempts, or fraudulent schemes that can compromise user data and security. Hence, efficient and real-time SMS spam detection has become an essential task in modern mobile ecosystems.

Traditional spam filters depend on manually crafted rules or classical machine learning models, which frequently encounter difficulties in adjusting to challenging nature of spam content. As improvements in “deep learning” and “natural language processing” (NLP), more sophisticated models namely “Long Short-Term Memory” (LSTM) networks highlighted superior performance in sequence-based text classification tasks, including spam detection. LSTM models, by design, can learn long-term dependencies in text, enabling them ideal in capturing contextual patterns found in spam messages.

This research presents **SpamWatch**, a real-time SMS spam detection system powered by an LSTM model. System was developed for classifying incoming text messages as either "spam" or "ham" (legitimate) with high accuracy. It is further enhanced with a user-friendly frontend interface built using **Gradio**, allowing users to interact with the model easily. To ensure wide accessibility and deployment scalability, the entire application is hosted on **Hugging Face Spaces**, enabling seamless web-based use without local installation.

Through this work, we aim to close the gap between machine learning models at research level and real-world applications by demonstrating practicality of deploying deep learning-based spam detection tools on cloud platforms. Present paper outlines development, deployment, design, and assessment of SpamWatch system, emphasizing on model performance, usability, and accessibility.

2. Related Work

The challenge of detecting spam messages was active field of study, especially with the increasing dependence on mobile communication. Early spam detection systems relied heavily on rule-based filtering and keyword matching. Although these methods were simple to implement, they lacked the adaptability and context-awareness required to handle evolving spam techniques.

As machine learning progressed, traditional classifiers like “Decision Trees,” “Naive Bayes,” as well as “Support Vector Machines” (SVM) became popular for spam detection. These models improved detection accuracy by analyzing patterns in labeled datasets, but they often required extensive feature engineering and were less effective at capturing sequential relationships in text.

RNNs and LSTM networks in particular became potent tools for “natural language processing” (NLP) applications, like spam classification, with advent of deep learning. Capability of LSTM to remember long-term dependence in sequences made it suitable for understanding the context and semantics of SMS messages.

Recent research has investigated hybrid strategies that combine NLP preprocessing methods like tokenization and embedding with deep learning models to boost spam detection performance. Additionally, with the growth of real-time deployment needs, platforms like Hugging Face and tools like Gradio has enabled researchers to serve their models in production environments easily, allowing for public interaction, accessibility, and scalability.

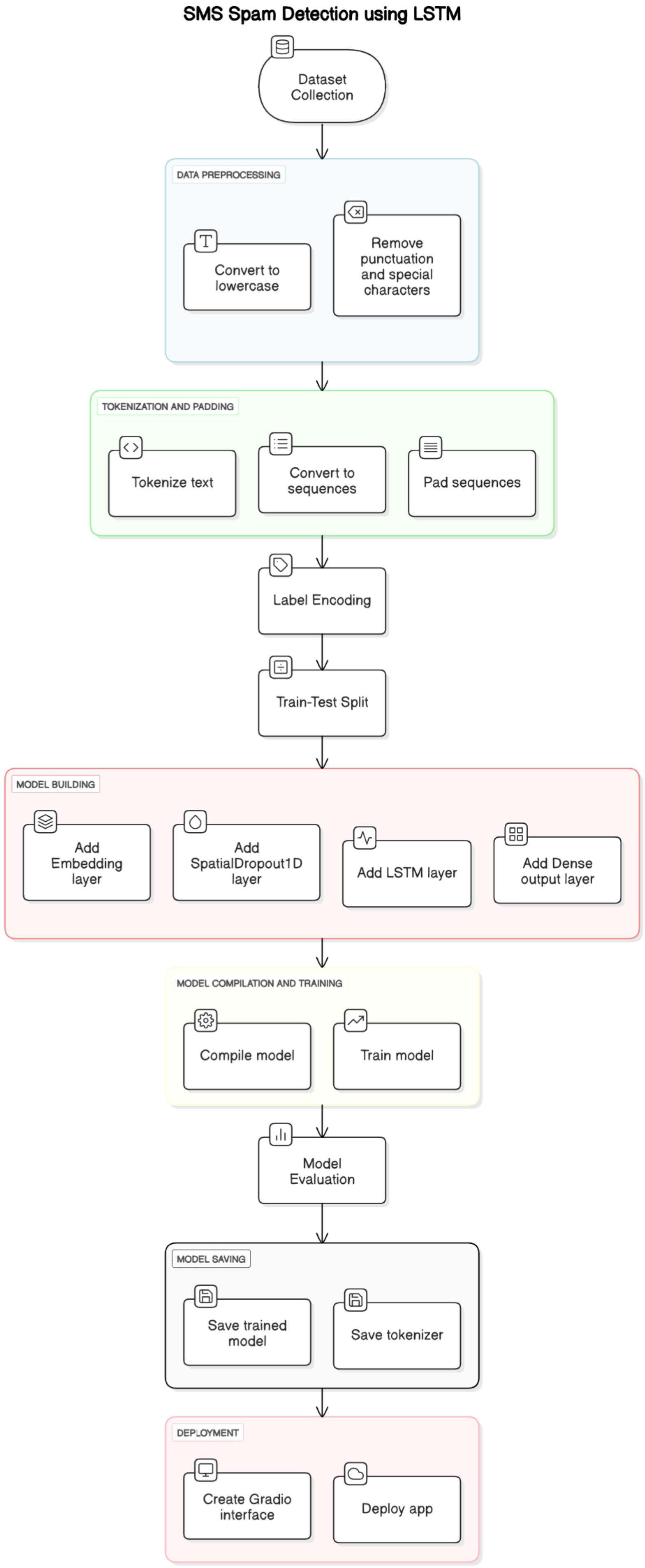
However, most existing works either focus purely on model accuracy or lack a streamlined approach to deployment. Our work bridges this gap by not only applying LSTM-based spam detection but also providing a deployable real-time solution using Hugging Face and Gradio. This practical end-to-end integration adds significant value to real-world applications, particularly in mobile environments where immediate filtering is essential.

3. Methodology

The development of the spam detection model was carried out through a structured process involving several important steps. First, we obtained a dataset containing SMS messages labeled as either spam or ham. The data underwent preprocessing to clean the text, convert it into a usable format, and ensure uniform input size for the model. This included techniques such as text normalization, tokenization, and padding.

Once the data was prepared, we designed an LSTM-based deep learning model, which is particularly effective in understanding sequential data like text. The model was trained on the processed dataset across multiple epochs, allowing it to learn the patterns and characteristics of spam and ham messages. After training, we evaluated the model’s performance using various metrics and visual tools like confusion matrix, accuracy and loss plots, and standard evaluation scores such as precision, recall, and F1-score.

Each stage of this process—from data handling to model evaluation—has been presented below along with relevant diagrams to illustrate the outcomes clearly.


SMS Spam Detection using LSTM
Dataset Collection
DATA PREPROCESSING
Convert to lowercase
Remove punctuation and special characters
TOKENIZATION AND PADDING
Tokenize text
Convert to sequences
Pad sequences
Label Encoding
Train-Test Split
MODEL BUILDING
Add Embedding layer
Add SpatialDropout1D layer
Add LSTM layer
Add Dense output layer
MODEL COMPILATION AND TRAINING
Compile model
Train model
Model Evaluation
MODEL SAVING
Save trained model
Save tokenizer
DEPLOYMENT
Create Gradio interface
Deploy app

3.1 Data Collection

Dataset utilized in this investigation is **SMS Spam Collection Dataset**, obtained from Kaggle. This corpus consists of "5,**574 English-language SMS messages**," each of which is classified as either **spam or ham(legitimate)**. Research on SMS spam detection makes extensive use of dataset and was originally compiled from various open-access sources.

Each entry in the dataset contains two fields:

- **v1**: Label, either "ham" or "spam".
- **v2**: Raw SMS message text.

3.2 Data Preprocessing

To assure efficacy of LSTM-based spam classification model, raw SMS messages underwent number of preprocessing procedures. To transform textual data into numerical format that deep learning models may understand, several procedures are crucial. Below is detailed breakdown of the preprocessing pipeline:

3.2.1 Text Normalization

Each message was first converted to lowercase to maintain consistency and reduce redundancy in the vocabulary:

$Message_{normalized}=lowercase(message_{original})$

3.2.2 Removal of Noise

Unwanted characters such as punctuation marks, special symbols, and numeric digits were removed using regular expressions:

$Message_{clean}=re.sub(pattern,'',message_{normalized})$

Where:

pattern = '[^a-zA-Z]'

This step ensures only alphabetic tokens remain, reducing input noise.

3.2.3 Tokenization

Each message is split into a sequence of individual words (tokens):

$tokens=Tokenizer(message_{clean})$

We define a vocabulary size V representing the number of unique tokens in the dataset:

$$V=/\{w_1,w_2,...,w_n\}/$$

3.2.4 Text-to-Sequence Conversion

Each tokenized message is converted to a numerical sequence, where each word is replaced with its corresponding index in the vocabulary:
$sequence=[i_1,i_2,...,i_n]$,where $i_k=index(w_k)$

3.2.5. Padding Sequences

Since LSTM networks require input of the same length, we padded all sequences to a fixed maximum length LLL:
$padded\_sequence=\{[i_1,i_2,...,i_l]\ \ if\ length>L\ \ ,\ \ [i_1,i_2,...,i_n,0,...,0]\ \ if\ \ n<L$

Padding ensures uniformity and prevents truncation of short messages.

3.2.6. Label Encoding

Binary numeric format for labels spam and ham was generated using:

labelencoded={0 if ham, 1 if spam

This preprocessing pipeline transforms the dataset into structured format appropriate for feeding into LSTM network, preserving text's sequential and semantic structure.

4. Model Development

To effectively classify SMS messages as spam or ham, we employed LSTM model. For sequential data, like text, LSTM networks— kind of Recurrent Neural Network (RNN)—are perfect because they can learn long-term dependencies.

4.1 Long Short-Term Memory (LSTM) model

LSTM-based spam detection model comprises **seven key layers**, each with a distinct role in processing and classifying SMS messages:

1. ***Input Layer****:*

   This is 1st layer where raw SMS text was fed into the model. The text is tokenized into a sequence of word indices. Mathematically, each input at time step t is represented as:

   X_t*=Tokenized input sequence*

2. ***Embedding Layer****:*

   This layer converts input tokens into dense vector representations that capture semantic meaning of words. The transformation is done via an embedding matrix EEE, and the output is:

   $x_t^{(emb)=}E \cdot x_t$

3. ***LSTM Layer (Long Short-Term Memory)****:*

   The core of model, LSTM layer processes sequence data and retains important information over time using memory cells. It involves several gates:

   - ***Forget Gate****: Identifies information that should be discarded from cell state:*
     $ft = \sigma (W_f \cdot [h_{t-1}, x_t] + b_f)$
   - ***Input Gate****: Determines new information to be stored in cell state:*
     $it = \sigma (W_i \cdot [h_{t-1}, x_t] + b_i)$
   - ***Cell Candidate****: Produces new candidate values that must be integrated into state:*
     $\sim C_t = \tanh (W_C \cdot [h_{t-1}, x_t] + b_C)$
   - ***Cell State Update****: Updates internal memory:*
     $C_t = f_t * C_{t-1} + i_t * \sim C_t$
   - ***Output Gate and Hidden State****: Computes output on basis of cell state:*
     $o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$
     $h_t = o_t * \tanh (C_t)$

4. ***Dropout Layer**:*
   Fraction p of input units is set to zero at random by this layer during training to avoid overfitting.

   $H_t^{drop}=Dropout(h_{t,}p)$

5. ***Dense Layer (Fully Connected Layer)**:*
   Output from this layer is taken from LSTM and maps it to single unit representing the classification result.

   $Z=W.h_t^{drop}+b$

6. ***Sigmoid Activation Layer**:*
   This layer squashes the output between 0 and 1, converting it into a probability.

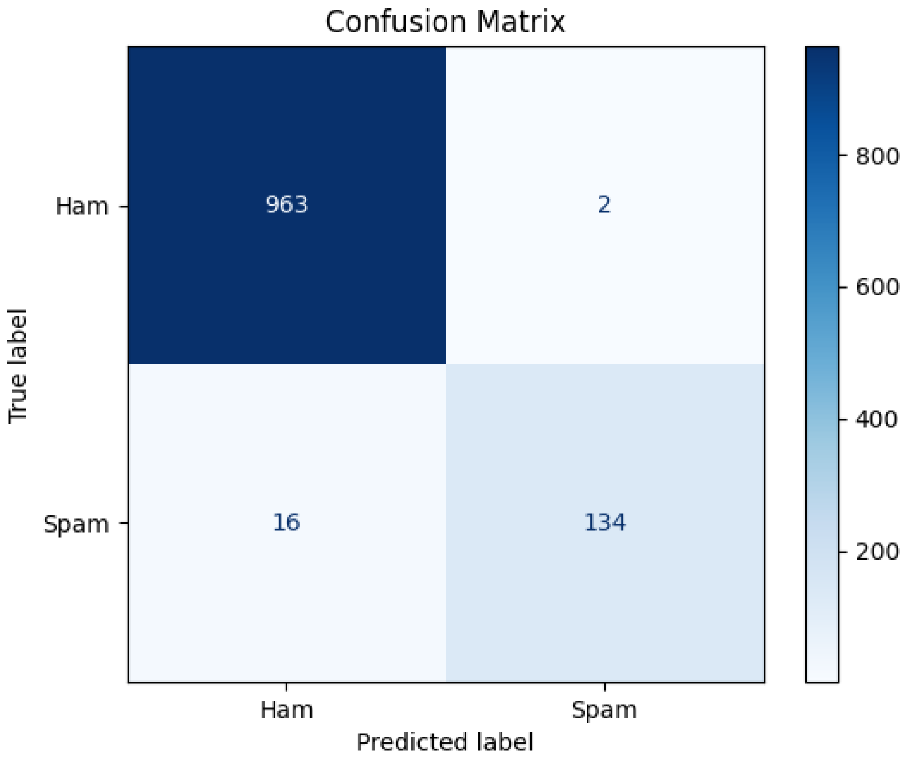
   $^{\wedge}Y= \sigma(z)=1/(1+e^{-z1})$

7. ***Output Layer**:*
   *Finally, the prediction is made. If the output probability ^y≥ 0.5 message is categorized as **spam**, or else as **ham(not spam)**.*

5.Model Evaluation

For ensuring effectiveness of LSTM-based spam detection model, numerous evaluation techniques has been used. These include the confusion matrix, accuracy and loss trends over epochs, and core performance metrics like recall, F1-score, precision, and accuracy. Together, these evaluations help measure how well model performs to classify SMS messages as spam or not spam. By analyzing these metrics, we can understand the model's strengths, readiness, and generalization capability for real-world deployment.

5.1. Confusion Matrix Visualization



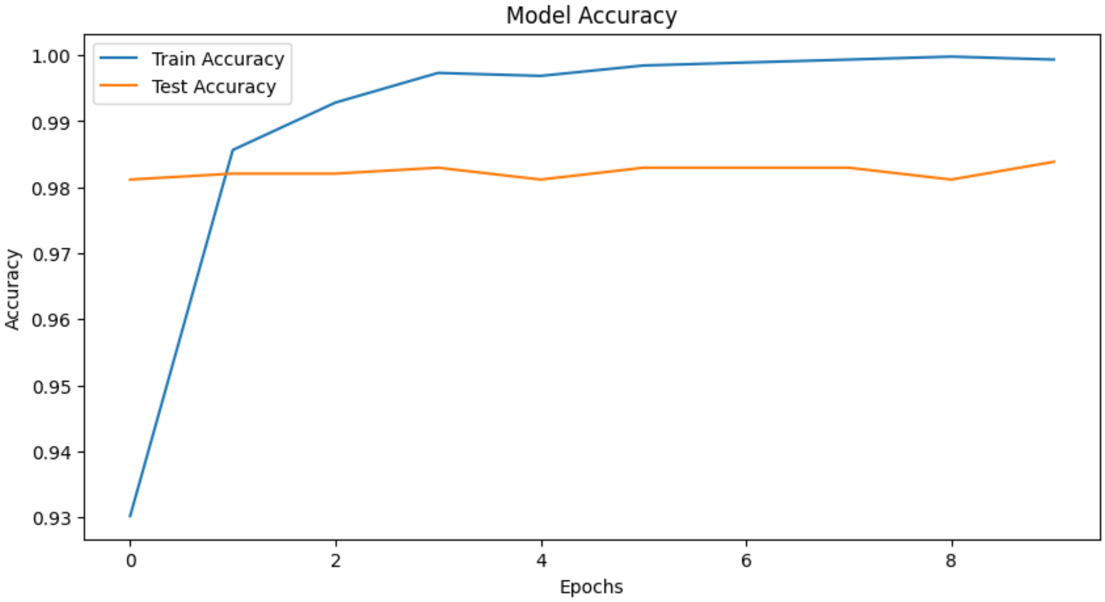
Ham (Not Spam) and Spam are two classes for which this matrix illustrates model's classification performance.

	Predicted Ham	Predicted Spam
Actual Ham	963	2
Actual Spam	16	134

- True Positives (Spam predicted as Spam) : 134
- True Negatives (Ham predicted as Ham) : 963
- False Positives (Ham predicted as Spam) : 2
- False Negatives (Spam predicted as Ham) : 16

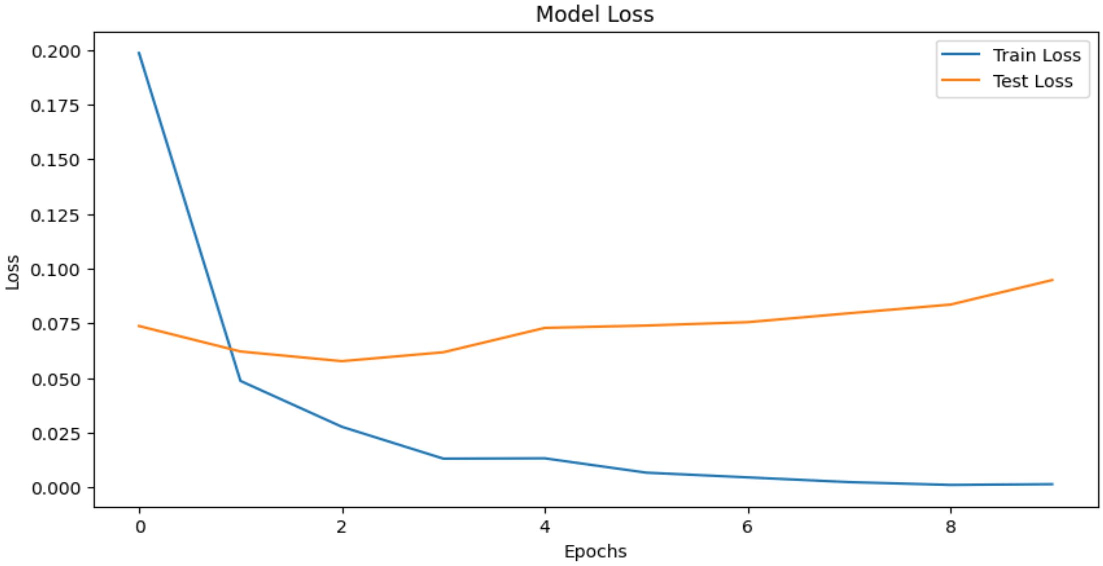
The matrix confirms that LSTM model have been quite accurate in differentiating between spam as well as ham messages, with a low false positive rate (0.2%) and a moderately low false negative rate (10.7%).

5. 2. Model Accuracy Over Epochs



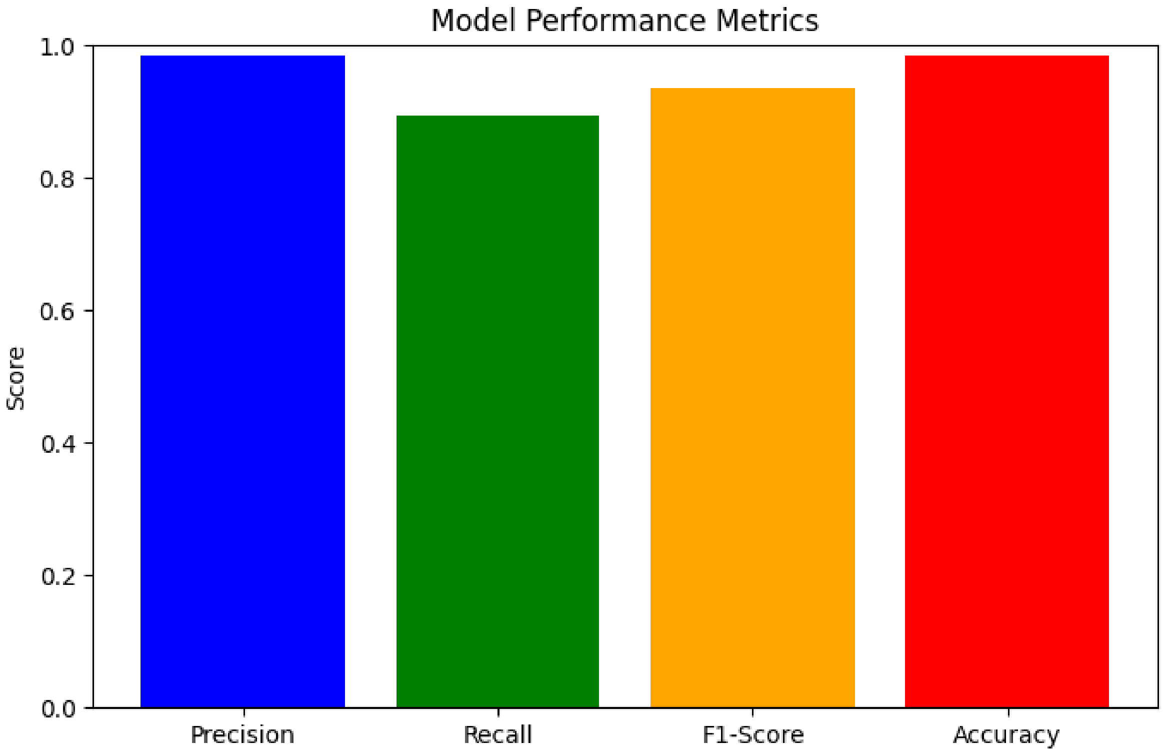
This graph shows LSTM model testing and training accuracy over 10 epochs. Training accuracy increases steadily and converges near 0.998, while test accuracy remains consistent around 0.98, indicating minimal overfitting and strong generalization performance

5. 3. Model Loss Over Epochs



This chart visualizes the loss curves for training and testing datasets. Training loss drops significantly & approaches zero, while test loss stabilizes around 0.07–0.08, confirming the model has learned the patterns well without severe overfitting.

5. 4. Model Performance Metrics



A bar chart summarizing the model's key performance metrics:

- Recall: ~ 0.89
- Precision: ~ 0.98
- Accuracy: ~ 0.98
- F1-Score: ~ 0.93

These scores demonstrate model's outstanding capacity to accurately detect spam messages while limiting false negatives and positives

<u>Final Model Evaluation Summary</u>

The LSTM-based spam detection model performs well on every important assessment metric:

Metric	Score
Precision	0.9853
F1-Score	0.9371
Recall	0.8933
Accuracy	0.9839

These values show reliability and efficacy model in handling real-world SMS spam detection. The consistent accuracy and high precision make it highly reliable for deployment in applications like mobile spam filters and secure messaging platforms.

6.Deployment of the Spam SMS Detection Model

To ensure real-time accessibility and user interaction, the trained LSTM-based spam detection model was deployed using **Gradio** on **Hugging Face Spaces**. This deployment method allowed the system to be hosted on the cloud and accessed via a simple web interface, requiring no installations or technical setup from end users.

1. Platform Selection

Hugging Face Spaces was chosen for deployment due to its:

- Free and reliable hosting for machine learning models.
- Seamless integration with Gradio, a Python-based UI toolkit.
- Support for public and collaborative machine learning applications.
-

2. Model Preparation

After the model was trained on SMS data, it was saved in a serialized format. This preserved the model's learned parameters and allowed it to be reused for inference during deployment.

3. Interface Development

A user interface was developed using Gradio, which allowed users to:

- Enter SMS messages into a text box.
- Receive real-time classification outcomes, highlighting whether message is **spam** or **not spam**.

The interface was designed to be simple, clean, and intuitive, making it accessible even to users without technical knowledge.

4. Dependency Management

All necessary libraries and dependencies (such as those used for text preprocessing and model prediction) were listed in a configuration file. This ensured that the correct environment would be automatically set up during deployment.

5. Deployment Workflow

The application files—including the user interface logic, model file, and configuration—were uploaded to a Hugging Face Space repository. Once uploaded, the Hugging Face platform automatically built the environment and launched the application.

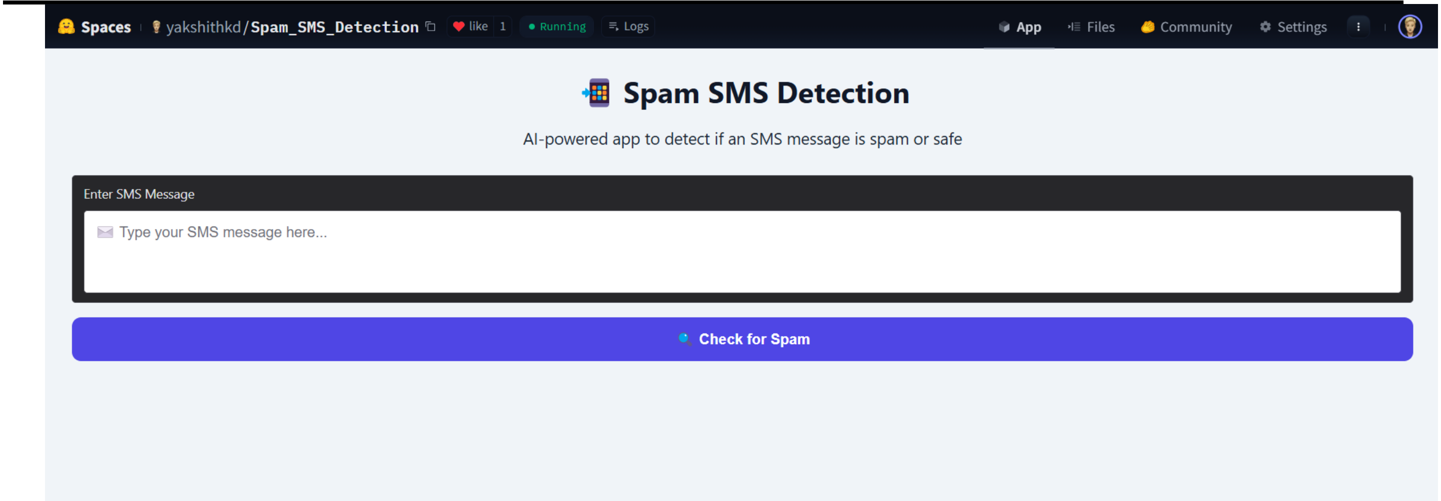
6. Public Accessibility

The final deployed model is hosted online and available for public use at: **https://huggingface.co/spaces/yakshithkd/Spam_SMS_Detection**

Users can visit this link, type in an SMS message, and instantly see whether it's flagged as spam.

7. User Interface Snapshot

The image below displays the live user interface of the deployed model



Spam SMS Detection

AI-powered app to detect if an SMS message is spam or safe

Enter SMS Message

Type your SMS message here...

Check for Spam

8. Conclusion

This project successfully developed and deployed an intelligent spam detection system employing LSTM neural network. Model proved efficacy of deep learning methods for natural language-based classification problems by correctly identifying SMS messages as spam or legitimate (ham).

Integration of Gradio for building front-end and Hugging Face Spaces for hosting enabled the creation of an interactive, real-time web application. This made the system easily accessible to users through a simple browser-based interface without any additional setup. By combining robust backend processing with user-friendly deployment, the system proves to be both technically sound and practically usable.

This work contributes to the ongoing efforts in combatting spam communication and highlights how modern machine learning models can be operationalized efficiently for real-world applications. Future improvements may involve multilingual support, integration with messaging platforms, or continuous learning from live user input.

9. Acknowledgment

In an effort to convey my heartfelt appreciation to the authors, and researchers of previous works in the fields of deep learning and natural language processing, particularly those who contributed valuable insights into LSTM-based text classification and spam detection systems. Their foundational research has significantly influenced the direction and understanding of this project.

This work, *Spam Message Detection using LSTM and Deployment via Hugging Face*, was developed independently. I would like to acknowledge the open-source community and datasets that provided essential resources for experimentation and implementation.

Their contributions have inspired me to explore the practical applications of theoretical machine learning models and to translate these concepts into a deployable solution accessible to users through a simple and effective web interface

References

[1] https://www.kaggle.com/code/ngawangchoeda/spam-classifier-using-lstm

[2] A Research Paper of SMS Spam Detection Arpita Laxman Gawade1 , Sneha Sagar Shinde2 , Samruddhi Gajanan Sawant3 , Rutuja Santosh Chougule4 , Mrs Almas Amol Mahaldar. 5 1, 2, 3, 4Department of Diploma in Computer Engineering, Third Year, Sharad Institute of Technology, Polytechnic Yadrav,Ichalkaranji, Kolhapur, Maharashtra, India 5Lecturer, Department of Diploma in Computer Engineering, Sharad Institute of Technology, Polytechnic Yadrav, Ichalkaranji, Kolhapur, Maharashtra, India

[3] Collection of SMS messages tagged as spam or legitimate- SMS Spam Collection Dataset

[4] https://www.kaggle.com/code/gadaadhaarigeek/spam-ham-detection-using-lstm

[5] Spam text classification using LSTM Recurrent Neural Network by Yeshwanth Zagabathuni

[6] SMS Spam Detection Based on Long Short-Term Memory and Gated Recurrent Unit Pumrapee Poomka, Wattana Pongsena, Nittaya Kerdprasop, and Kittisak Kerdprasop

[7] Chilukuri Lekhya Sri, D. Dhana Lakshmi, Kodati Ravali, and Shanmugasundaram Hariharan, "Improved Spam Detection Through LSTM-Based Approach," presented at the International Conference on Artificial Intelligence and Smart Energy (ICAIS), Coimbatore, India, March 2024

[8] Syed Md. Minhaz Hossain, Jayed Akbar Sumon, Anik Sen, Md. Iftaker Alam, Khaleque Md. Aashiq Kamal, Hamed Alqahtani, and Iqbal H. Sarker, "Spam Filtering of Mobile SMS Using CNN–LSTM Based Deep Learning Model," in Proceedings of the International Conference on Advanced Data Mining and Applications (ADMA), Springer, 2022. Available: https://www.researchgate.net/publication/358988095 [Accessed: Apr. 7, 2025].

[9] Jain, A.K., Yadav, S.K., Choudhary, N.: A novel approach to detect spam andsmishing SMS using machine learning techniques. IJESMA 12, 21–38 (2020)

[10] ain, A.K., Gupta, B.B.: Feature based approach for detection of smishing messagesin the mobile environment. J. Inf. Technol. Res. 12, 17–35 (2019)14. Jain, A.K., Gupta, B.: Rule-based framework for detection of smishing messagesin mobile environment. Procedia Comput. Sci. 125, 617–623 (2018)15. Almeida, T.A., Silva, T.P., Santos, I., Hidalgo, J.M.G.: Text normalization and semantic indexing to enhance Instant Messaging and SMS spam filtering. Knowl.Based Syst. 108, 25–32 (2016)
(PDF) Spam Filtering of Mobile SMS Using CNN–LSTM Based Deep Learning Model. Available from: https://www.researchgate.net/publication/358988095_Spam_Filtering_of_Mobile_SMS_Using_CNN-LSTM_Based_Deep_Learning_Model [accessed Apr 07, 2025].

[11] Arifin, D.D., Bijaksana, M.A.: Enhancing spam detection on mobile phone ShortMessage Service (SMS) performance using FP-growth and Naive Bayes Classifier.In: Proceedings of the 2016 IEEE Asia Pacific Conference on Wireless and Mobile(APWiMob), Bandung, Indonesia, 13–15 September 2016, pp. 80–84 (2016)
(PDF) Spam Filtering of Mobile SMS Using CNN–LSTM Based Deep Learning Model. Available from: https://www.researchgate.net/publication/358988095_Spam_Filtering_of_Mobile_SMS_Using_CNN-LSTM_Based_Deep_Learning_Model [accessed Apr 07, 2025].

[12] HuggingFace's Transformers: State-of-the-art Natural Language Processing
Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, Alexander M. Rush
https://huggingface.co/docs