IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2025

DOI: 10.17148/IJIREEICE.2025.131031

SMS Spam Classifier

Sura Reddy[1], Sriram K[2], Brunda G[3], Dr. Golda Dilip[4]

Student, Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India[1]

Student, Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India[2]

Student, Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India[3]

Guide, Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India[4]

Abstract: This project focuses on classifying SMS messages as spam or ham (not spam) using machine learning models. The system collects a labelled dataset of SMS messages, performs text preprocessing (tokenization, stop-word removal, lemmatization), converts text into numerical features using TF-IDF or Word Embeddings, and trains classifiers such as Logistic Regression, Naive Bayes, and SVM. The model achieving the highest accuracy is used for deployment through a Streamlit web app.

Keywords: Machine Learning, NLP, Spam Detection, SMS Classification, TF-IDF, Streamlit

I. INTRODUCTION

Classifying spam messages is a key challenge in modern communication due to the rapid increase in unsolicited texts. Traditional rule-based systems fail to adapt to new spam patterns and languages. Machine Learning and Natural Language Processing techniques address this issue by learning patterns from text data.

This project focuses on developing an SMS Spam Classifier using machine learning algorithms to identify whether a message is spam or legitimate. The system uses text preprocessing and TF-IDF vectorization for feature extraction and is deployed as a user-friendly web app using Streamlit.

II. SYSTEM WORKFLOW

The application follows a modular structure with five key components: data collection, preprocessing, feature extraction, model training, and prediction.

A. Data Collection

The system uses the **SMS Spam Collection Dataset** from the UCI Machine Learning Repository.
It contains labelled messages classified as *spam* or *ham* (not spam).

B. Preprocessing

The text data is cleaned by removing punctuation, numbers, and special symbols.
All words are converted to lowercase, tokenized, and stop words are removed.
Lemmatization is applied to reduce words to their base form.

C. Feature Extraction

The cleaned text is converted into numerical vectors using the **TF-IDF Vectorizer**, which assigns importance to words based on frequency and relevance.

D. Model Training

Machine learning models such as **Naive Bayes**, **Logistic Regression**, and **Support Vector Machine (SVM)** are trained on the feature vectors.

E. Prediction and Deployment

The best-performing model is integrated into a **Streamlit web app** that allows users to enter an SMS message and instantly view whether it is spam or ham.

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

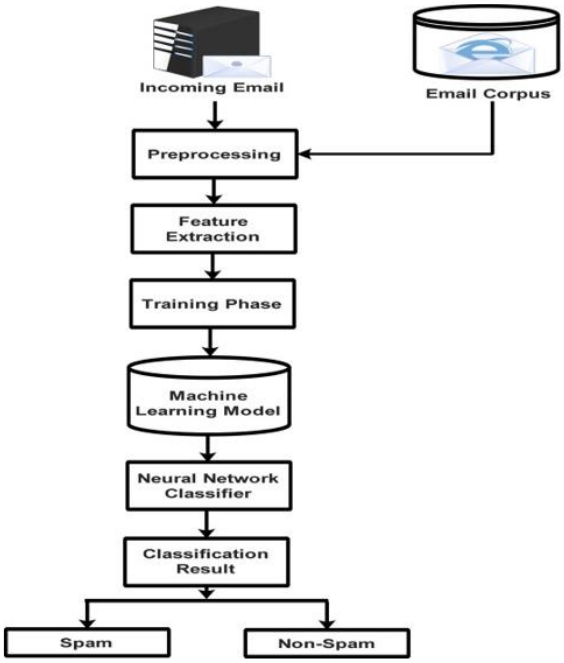
Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2025

DOI: 10.17148/IJIREEICE.2025.131031

III. FEATURES

1. User-friendly **web interface** built using Streamlit.
2. **Real-time message classification** into *Spam* or *Ham*.
3. **Text preprocessing and cleaning** using NLP techniques.
4. **TF-IDF based feature extraction** for better accuracy.
5. **Multiple ML models** (Naive Bayes, Logistic Regression, SVM) for comparison.
6. **Performance metrics** such as Accuracy, Precision, Recall, and F1-Score.

IV. ARCHITECTURE DIAGRAM



V. TECHNICAL STACK

S.No.	Component	Technology Used	Description
1	Frontend	Streamlit	Provides an interactive web interface for entering SMS text and displaying results.
2	Backend (ML Model)	Scikit-learn	Trains and runs models like Naive Bayes and Logistic Regression for classification.
3	Programming Language	Python	Handles preprocessing, model training, and deployment tasks.
4	Libraries	Pandas, NumPy, NLTK, Scikit-learn	Used for data cleaning, text processing, and evaluation.
5	Dataset	SMS Spam Collection (UCI Repository)	Contains labeled SMS messages categorized as spam or ham.
6	Feature Extraction	TF-IDF Vectorizer	Converts cleaned text into weighted numerical vectors.
7	Evaluation Metrics	Accuracy, Precision, Recall, F1-Score	Measures and compares model performance.

VI. CONCLUSION

The SMS Spam Classifier successfully identifies spam and legitimate messages using machine learning and NLP techniques. The combination of TF-IDF feature extraction and algorithms like Naive Bayes and Logistic Regression delivers high accuracy and efficiency. The project proves the potential of ML in automating text classification tasks and can be further enhanced with deep learning models or multilingual dataset support.

DOI: 10.17148/IJIREEICE.2025.131031

REFERENCES

[1]. T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami, "Contributions to the study of SMS spam filtering," *Proc. 11th ACM Symp. Document Eng.*, 2011, pp. 259–262.
[2]. M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, "A Bayesian approach to filtering junk e-mail," *AAAI Tech. Rep. WS-98-05*, 1998.
[3]. UCI Machine Learning Repository, "SMS Spam Collection Dataset," 2017. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/sms+spam+collection
[4]. Scikit-learn Developers, "Scikit-learn: Machine Learning in Python," 2025. [Online]. Available: https://scikit-learn.org/
[5]. GitHub, "SMS Spam Detection using ML," 2025. [Online]. Available: https://github.com/
[6]. Scikit-learn Developers, "Scikit-learn: Machine Learning in Python," 2025. [Online]. Available: https://scikit-learn.org/
[7]. TensorFlow Developers, "Text classification with TensorFlow," 2024. [Online]. Available: https://www.tensorflow.org/text/tutorials
[8]. GitHub, "SMS Spam Detection using ML," 2025. [Online]. Available: https://github.com/