MONARK UNIVERSITY

Intelligent Spam vs Ham Email/SMS Classification

Using Machine Learning (TF-IDF & SVM)

Submitted by
Ayush Gupta

Under the Guidance of
Dean Prof. Shobhit Singh
Prof. Hammad Farid

Department of MSc IT (AI/ML)
Academic Year: 2025–2026

ABSTRACT

The rapid growth of digital communication has led to an increase in unwanted and fraudulent messages, commonly known as spam. This project focuses on the development of an intelligent Spam vs Ham classification system for Email and SMS messages using Machine Learning techniques.

The proposed system applies text preprocessing and feature extraction methods such as Count Vectorizer, TF-IDF, and N-Grams to convert raw text into numerical form. Multiple machine learning algorithms including Naive Bayes, Logistic Regression, Support Vector Machine (SVM), K-Nearest Neighbors, Decision Tree, and Random Forest are implemented and compared.

Based on performance evaluation metrics like accuracy, precision, recall, and F1-score, the SVM model combined with TF-IDF feature extraction achieved the best results. The system effectively classifies messages as spam or ham and can be used to enhance message filtering and user security in real-world applications.

INDEX / TABLE OF CONTENTS

Contents

Introduction

In today's digital era, communication through Email and Short Message Service (SMS) has become an essential part of daily life. Along with their widespread use, the problem of unwanted and irrelevant messages, commonly known as spam, has increased significantly. Spam messages often contain promotional content, misleading information, or fraudulent links, which can lead to security risks and poor user experience. In contrast, legitimate messages, referred to as ham, contain useful and trustworthy information. Therefore, accurate classification of spam and ham messages has become an important requirement in modern communication systems.

Traditional spam filtering techniques mainly depend on rule-based systems and keyword matching. Although these methods are simple to implement, they lack adaptability and fail to detect newly evolving spam patterns. Spammers continuously change message structures to bypass such filters, which reduces the effectiveness of traditional approaches. To overcome these limitations, Machine Learning techniques have emerged as a reliable and efficient solution for text classification problems.

This project presents an **Intelligent Spam vs Ham Email/SMS Classification System** using Machine Learning techniques. Text messages are first preprocessed and converted into numerical features using the **TF-IDF (Term Frequency–Inverse Document Frequency)** method. Multiple machine learning algorithms such as Naive Bayes, Logistic Regression, K-Nearest Neighbors, Decision Tree, Random Forest, and Support Vector Machine (SVM) were implemented and evaluated.

Based on experimental analysis and performance evaluation, the **Support Vector Machine (SVM) combined with TF-IDF** achieved the highest accuracy and overall performance. The proposed system effectively classifies Email and SMS messages as spam or ham and can be applied in real-world messaging platforms to improve communication quality and user security.

1 Literature Review

Spam detection in Email and SMS communication has been an active area of research due to the rapid growth of digital messaging platforms. Over the years, researchers have proposed various techniques ranging from rule-based filtering systems to advanced machine learning and deep learning approaches. This section reviews key studies related to spam detection and highlights how existing research supports the methodology adopted in this project.

Early spam filtering systems primarily relied on rule-based techniques and keyword matching. Although these methods were simple to implement, they lacked flexibility and failed to adapt to evolving spam patterns. As spammers continuously modify message structures, traditional filtering mechanisms became less effective and resulted in higher false detection rates.

To overcome these limitations, supervised machine learning algorithms gained popularity for spam detection tasks. Several studies demonstrated that classifiers such as Naive Bayes, Logistic Regression, and Support Vector Machines perform effectively when trained on labeled SMS datasets. The SMS Spam Collection dataset from the UCI Machine Learning Repository has been widely used as a benchmark dataset for evaluating these models.

Feature extraction plays a crucial role in text classification problems. Research indicates that techniques such as Bag of Words, TF-IDF, and N-grams significantly improve classifier performance by converting raw text into numerical features. Among these methods, TF-IDF has proven to be highly effective as it reduces the influence of commonly occurring but less informative words.

Comparative studies have shown that Support Vector Machine (SVM) classifiers consistently achieve higher accuracy when combined with TF-IDF features. SVM is well-suited for handling high-dimensional and sparse text data, making it an ideal choice for spam detection tasks. These findings strongly justify the selection of TF-IDF and SVM in this project.

Recent research has explored deep learning models such as Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), and Transformer-based architectures for spam detection. While these models demonstrate high accuracy, they require significant computational resources and large datasets, limiting their suitability for lightweight real-time applications.

Based on the insights obtained from existing literature, this project adopts a machine learning–based approach using TF-IDF for feature extraction and Support Vector Machine for classification. This approach offers an optimal balance between accuracy, efficiency, and computational simplicity, making it suitable for practical Email and SMS spam detection systems.

2 Dataset Description and Analysis

This section describes the dataset used in the project and the analysis performed before model training. Proper understanding of the dataset is essential for building an accurate and reliable spam detection system.

2.1 Dataset Source and Structure

The dataset used in this project is an email/SMS spam dataset obtained from a publicly available source. The dataset consists of **5,730 text messages**, where each message is labeled as either **spam** or **ham**.

Each record in the dataset contains two main attributes:

- **Text**: The actual content of the email or SMS message.
- **Label**: A binary value where `1` represents spam and `0` represents ham (legitimate message).

This structured format makes the dataset suitable for supervised machine learning algorithms, particularly text classification models.

2.2 Data Cleaning and Preprocessing

Before applying machine learning algorithms, the dataset was cleaned and preprocessed to improve model performance. The following preprocessing steps were performed:

- Removed unnecessary columns and retained only text and label fields.
- Converted label values into integer format for classification.
- Removed missing and invalid label entries.
- Calculated message length for exploratory analysis.

Text preprocessing techniques such as lowercasing, removal of special characters, stopword elimination, and tokenization were applied. These steps helped in reducing noise and preparing the text data for feature extraction using TF-IDF.

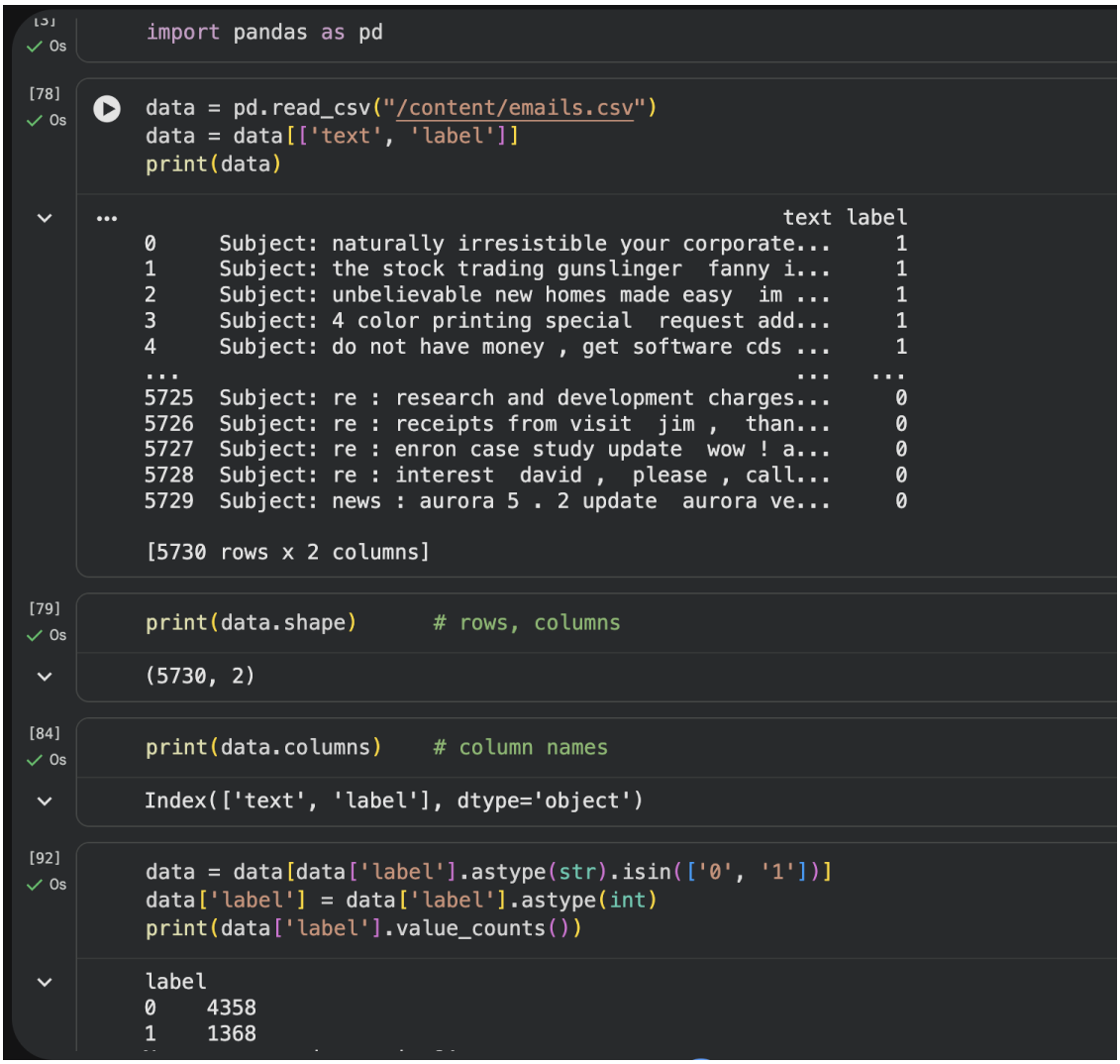

Figure 1: Dataset Loading and Initial Inspection Code

2.3 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) was performed to understand the characteristics of the dataset. Various visualizations were used to analyze message distribution and length patterns.

The analysis showed that ham messages are more frequent compared to spam messages. Additionally, spam messages generally tend to be longer and contain promotional or suspicious content, while ham messages are usually shorter and conversational in nature.

Histograms and count plots were used to visualize message length distribution and class imbalance. These insights helped in selecting appropriate feature extraction and classification techniques.

2.4 Spam vs Ham Distribution

The distribution of spam and ham messages was analyzed using bar charts and pie charts. The dataset contains approximately **76% ham messages** and **24% spam messages**, indicating a class imbalance.

Despite this imbalance, the TF-IDF feature extraction method combined with the Support Vector Machine (SVM) classifier handled the data effectively. SVM performed well in separating spam and ham messages by learning clear decision boundaries in high-dimensional text feature space.

This distribution analysis played a crucial role in selecting SVM as the final model due to its robustness and high accuracy in text classification tasks.

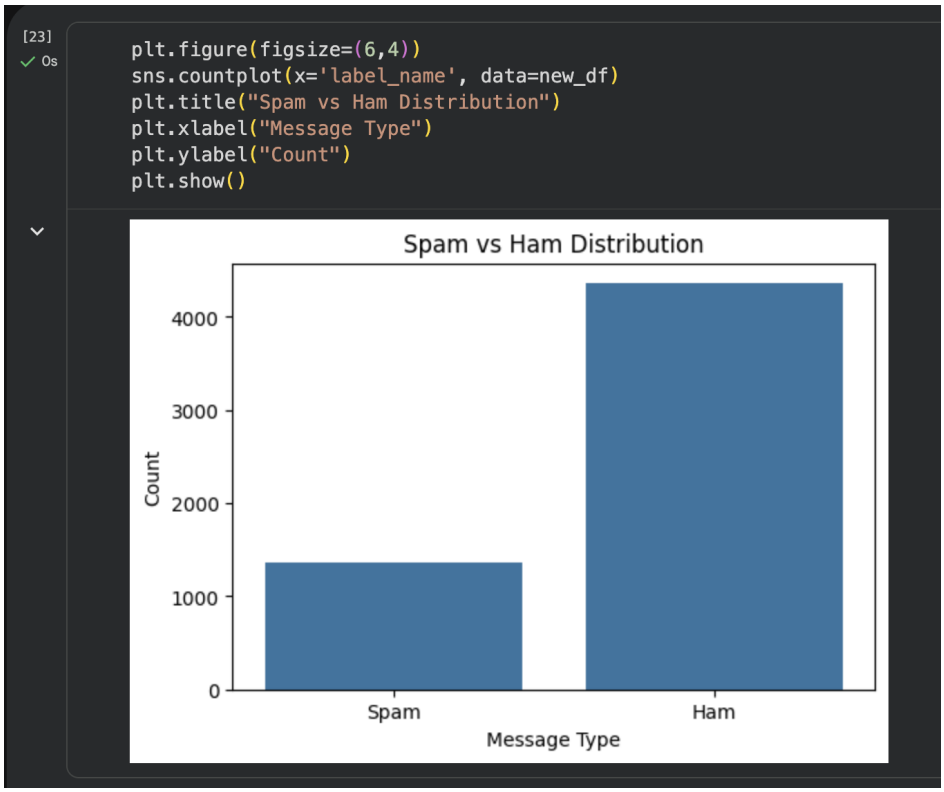

Figure 2: Spam vs Ham Message Distribution

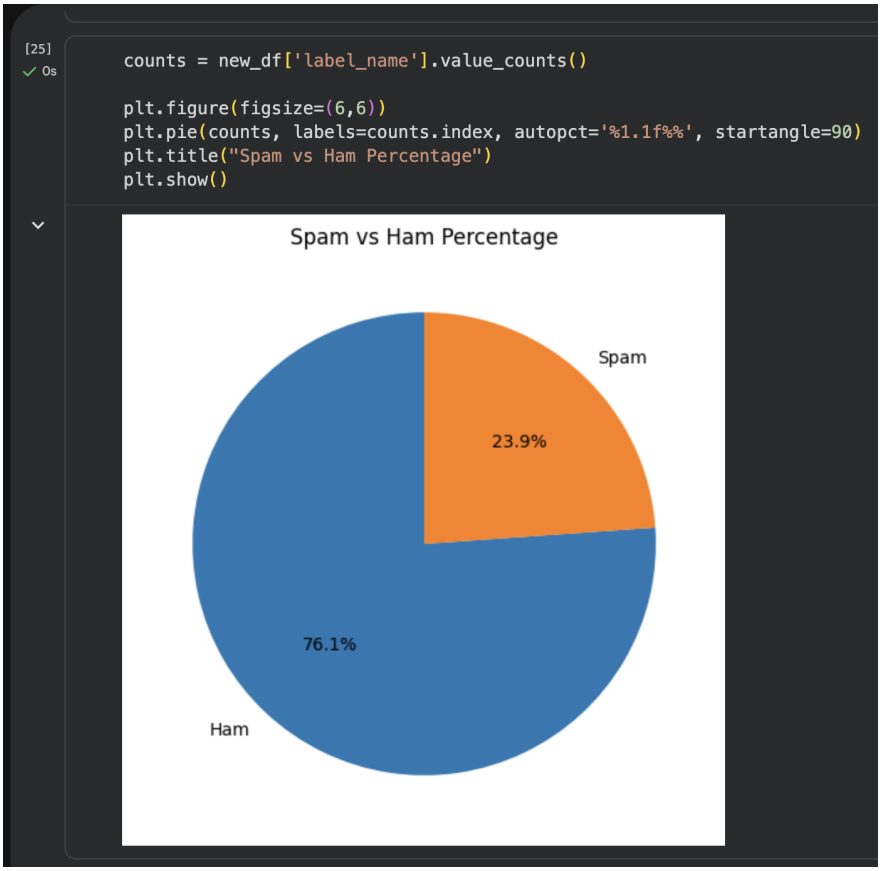

Figure 3: Spam vs Ham Percentage Distribution

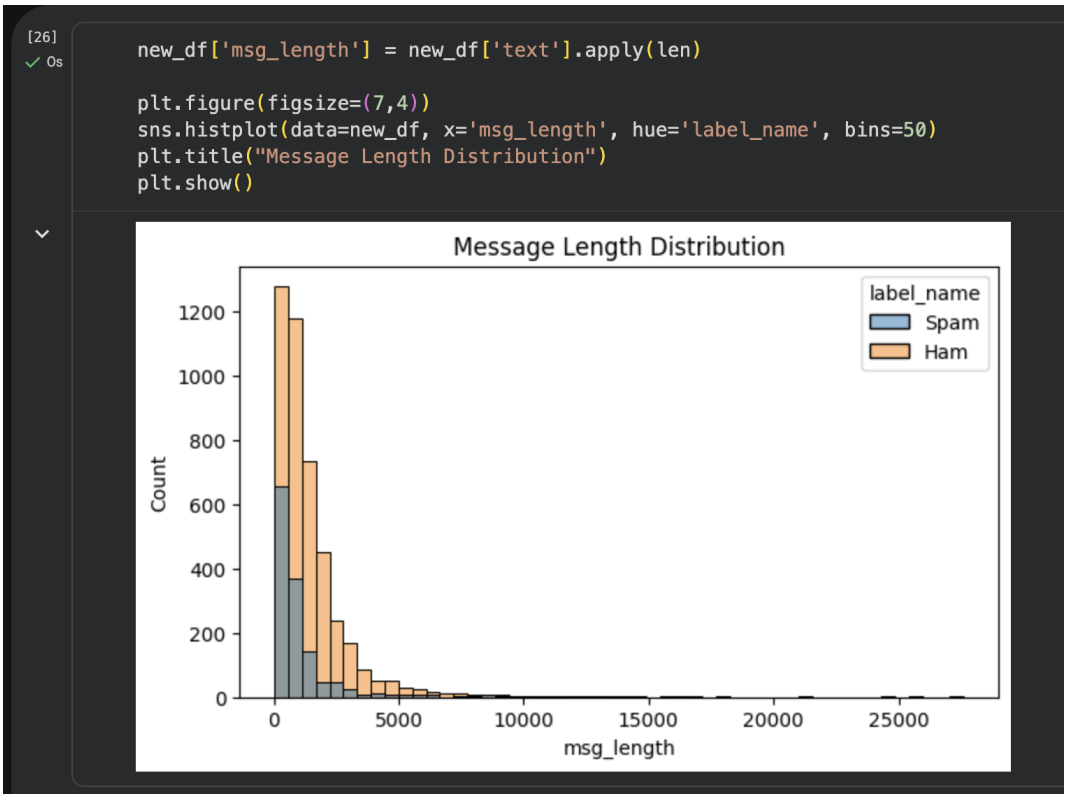

Figure 4: Message Length Distribution for Spam and Ham

3 Feature Extraction Techniques

Machine learning algorithms cannot process raw textual data directly. Therefore, it is necessary to convert text messages into numerical representations known as features. In this project, multiple feature extraction techniques were implemented and evaluated to identify the most effective method for Spam vs Ham classification. The techniques used include Bag of Words (Count Vectorizer), TF-IDF Vectorizer, and N-Gram based features.

3.1 Bag of Words (Count Vectorizer)

The Bag of Words (BoW) model is one of the simplest and most commonly used techniques for text representation. In this approach, a text document is represented as a vector of word frequencies. Each unique word in the dataset becomes a feature, and the value of each feature corresponds to the number of times the word appears in the message.

For example, consider the following messages:

- "Free offer now"
- "Free money offer"

The vocabulary will consist of the words: free, offer, now, money. Each message is then converted into a numerical vector based on word counts.

Although the Count Vectorizer is easy to implement and computationally efficient, it assigns equal importance to all words and does not consider word relevance across documents. Due to this limitation, it showed lower performance compared to TF-IDF in this project.

3.2 Term Frequency–Inverse Document Frequency (TF-IDF)

TF-IDF is an advanced feature extraction technique that improves upon the Bag of Words model. It assigns weights to words based not only on their frequency within a document but also on their importance across the entire dataset.

TF-IDF consists of two components:

- **Term Frequency (TF):** Measures how frequently a word appears in a document.
- **Inverse Document Frequency (IDF):** Measures how rare a word is across all documents.

The mathematical formulas are given below:

$$TF(t,d) = \frac{Numberofoccurrencesoftermtindocumentd}{Totalnumberoftermsindocumentd}$$

$$IDF(t) = \log\left(\frac{N}{df(t)}\right)$$

$$TF{-}IDF(t,d) = TF(t,d) \times IDF(t)$$

where:

- N is the total number of documents
- $df(t)$ is the number of documents containing term t

TF-IDF gives higher weight to words that are frequent in spam messages but rare across the dataset, such as "free", "win", "offer", and "urgent". In this project, TF-IDF significantly improved classification performance.

When combined with the Support Vector Machine (SVM) classifier, TF-IDF achieved the highest accuracy, precision, recall, and F1-score among all tested combinations.

3.3 N-Gram Based Features

N-Grams are contiguous sequences of words used to capture contextual information. Instead of considering single words (unigrams), N-Grams capture word combinations such as bigrams and trigrams.

Examples:

- Unigram: "free"
- Bigram: "free offer"
- Trigram: "win free money"

N-Gram features help in identifying spam-related phrases more effectively than single words. In this project, N-Gram features improved performance compared to the Count Vectorizer. However, the results were still slightly inferior to the TF-IDF based approach.

3.4 Comparison of Feature Extraction Techniques

A comparison of the feature extraction techniques used in this project is shown in Table 1.

Technique	**Information Captured**	**Model Used**	**Performance**
Count Vectorizer	Word frequency	Naive Bayes, SVM	Medium
N-Gram Features	Word sequences	SVM	High
TF-IDF Vectorizer	Word importance	SVM	**Highest**

Table 1: Comparison of Feature Extraction Techniques

3.5 Final Selection

Based on experimental results and performance evaluation, the TF-IDF Vectorizer was selected as the final feature extraction technique for this project. Its ability to highlight important spam-related terms and suppress common words made it the most effective method when combined with the Support Vector Machine classifier.

4 Machine Learning Models

After extracting meaningful features from textual data using TF-IDF and N-Grams, various machine learning algorithms were implemented and evaluated for spam and ham classification. Each model was trained on the same dataset and compared based on performance metrics such as accuracy, precision, recall, and F1-score.

4.1 Naive Bayes Classifier

Naive Bayes is a probabilistic classifier based on Bayes' Theorem. It assumes independence among features, which makes it computationally efficient for text classification tasks.

Bayes' Theorem is defined as:

$$P(C|X) = \frac{P(X|C) \times P(C)}{P(X)}$$

where:

- C represents the class (Spam or Ham)
- X represents the input message

Naive Bayes performs well on smaller datasets; however, its strong independence assumption limits its performance on complex textual patterns.

4.2 Logistic Regression

Logistic Regression is a linear classification algorithm that predicts the probability of a message being spam or ham using a sigmoid function.

The sigmoid function is given by:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Logistic Regression provides good accuracy and interpretability, but it may struggle with non-linearly separable data when compared to more advanced classifiers.

4.3 Support Vector Machine (SVM)

Support Vector Machine (SVM) is a powerful supervised learning algorithm that constructs an optimal hyperplane to separate spam and ham messages with maximum margin.

The decision function is defined as:

$$f(x) = w \cdot x + b$$

SVM aims to maximize the margin between the two classes while minimizing classification error. When combined with TF-IDF features, SVM effectively handles high-dimensional sparse data, making it the best-performing model in this project.

4.4 K-Nearest Neighbors (KNN)

KNN is an instance-based learning algorithm that classifies a message based on the majority class of its nearest neighbors.

Distance is commonly calculated using Euclidean distance:

$$d = \sqrt{\sum (x_i - y_i)^2}$$

While KNN is simple to understand, it is computationally expensive for large datasets and performs poorly with high-dimensional TF-IDF vectors.

4.5 Decision Tree

Decision Tree classifiers use a tree-like structure where internal nodes represent conditions and leaf nodes represent class labels.

Although Decision Trees are easy to interpret, they are prone to overfitting, especially when handling high-dimensional text data.

4.6 Random Forest

Random Forest is an ensemble learning method that combines multiple Decision Trees to improve prediction accuracy and reduce overfitting.

Despite improved stability over Decision Trees, Random Forests are less effective for sparse text data compared to linear models like SVM.

4.7 Comparison of Machine Learning Models

Model	Accuracy	Advantages	Limitations
Naive Bayes	Moderate	Fast, Simple	Strong assumptions
Logistic Regression	High	Stable, Interpretable	Limited non-linearity
SVM	**Highest**	Robust, High accuracy	Parameter tuning
KNN	Low	Easy to understand	Slow, memory intensive
Decision Tree	Moderate	Easy to visualize	Overfitting
Random Forest	High	Reduced overfitting	Complex, slow

Table 2: Comparison of Machine Learning Models

4.8 Best Performing Model

Based on extensive experimentation and evaluation, Support Vector Machine (SVM) combined with TF-IDF feature extraction achieved the highest classification accuracy. The model demonstrated superior performance in handling high-dimensional text data and effectively separated spam and ham messages. Therefore, TF-IDF with SVM was selected as the final model for this project.

5 System Architecture and Workflow

The system architecture of the Spam vs Ham Detection project is designed to efficiently process user messages and accurately classify them as spam or ham. The architecture follows a modular pipeline consisting of data collection, preprocessing, feature extraction, model training, and prediction. The integration of TF-IDF feature extraction with the Support Vector Machine (SVM) classifier ensures high accuracy and robustness.

5.1 Overall System Architecture

The overall architecture of the system consists of the following major components:

- Dataset Collection

- Data Cleaning and Preprocessing
- Feature Extraction using TF-IDF
- Model Training using SVM
- Spam/Ham Prediction
- User Interface for message input and result display

Each component operates sequentially to transform raw textual data into meaningful predictions.

5.2 Training and Testing Process

The training and testing process begins with splitting the dataset into training and testing subsets. In this project, 80% of the data is used for training the model, while the remaining 20% is reserved for testing.

During training:

- Text messages are converted into numerical vectors using TF-IDF
- The SVM model learns the optimal decision boundary between spam and ham classes

During testing:

- The trained model predicts labels for unseen messages
- Predictions are compared with actual labels to evaluate performance

This separation ensures that the model generalizes well to new, unseen data.

5.3 Spam / Ham Prediction Flow

The spam detection process follows a well-defined prediction flow:

1. User enters an email or SMS message
2. Text preprocessing is applied (lowercasing, stop-word removal)
3. TF-IDF vectorizer transforms text into numerical features
4. The trained SVM model analyzes the TF-IDF vector
5. The message is classified as either Spam or Ham
6. The result is displayed to the user through the interface

This structured flow enables real-time and accurate spam detection.

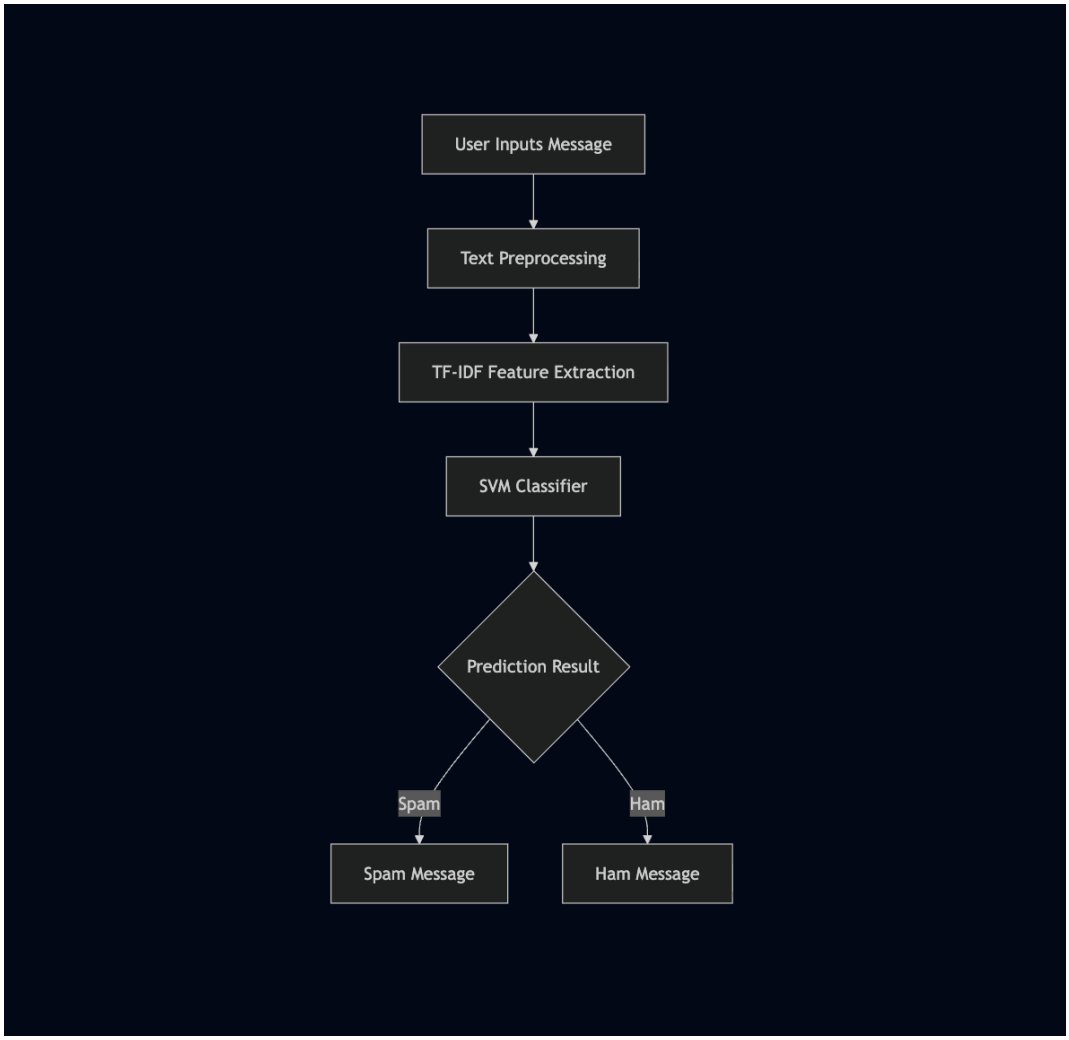

Figure 5: Spam vs Ham Prediction Flow using TF-IDF and SVM

5.4 Justification of System Design

The combination of TF-IDF and SVM was chosen because TF-IDF effectively captures the importance of words in messages, while SVM excels at handling high-dimensional sparse data. Experimental results showed that this combination achieved the highest accuracy compared to other models, making it the most suitable choice for the proposed system.

6 Performance Evaluation

This section evaluates the performance of different machine learning models used for spam and ham classification. Multiple evaluation metrics are employed to measure the effectiveness of each model, and a comparative analysis is conducted to identify the best-performing approach for the proposed system.

6.1 Evaluation Metrics

To assess the performance of the spam detection models, the following standard evaluation metrics are used:

- **Accuracy**: Measures the overall correctness of the model by calculating the ratio of correctly classified messages to the total number of messages.
- **Precision**: Indicates how many messages predicted as spam are actually spam. High precision is important to reduce false positives.
- **Recall**: Measures the ability of the model to correctly identify actual spam messages. High recall ensures fewer spam messages are misclassified as ham.
- **F1-Score**: The harmonic mean of precision and recall, providing a balanced measure of model performance.

These metrics collectively provide a comprehensive evaluation of the classification models.

6.2 Model Comparison and Analysis

Several machine learning models were trained and tested using different feature extraction techniques, including Count Vectorizer, TF-IDF, and N-Gram features. The performance of each model-feature combination was evaluated using accuracy, precision, recall, and F1-score.

Figure 6 presents the accuracy comparison of all implemented models. It is observed that models using TF-IDF generally outperform those using Count Vectorizer and N-Gram features. Among all combinations, Support Vector Machine (SVM) with TF-IDF achieved the highest accuracy, precision, and F1-score, demonstrating its robustness in handling high-dimensional sparse text data.

[44] ✓ 0s

```
results_df = pd.DataFrame(results)
results_df.sort_values(by="Accuracy", ascending=False)
```

	Model	Accuracy	Precision	Recall	F1
8	SVM + TFIDF	0.994760	1.000000	0.979522	0.989655
14	SVM + NGram	0.993886	1.000000	0.976109	0.987910
0	NB + CountVec	0.993013	0.993080	0.979522	0.986254
1	Logistic + CountVec	0.992140	0.993056	0.976109	0.984509
2	SVM + CountVec	0.984279	0.979094	0.959044	0.968966
7	Logistic + TFIDF	0.981659	0.996350	0.931741	0.962963
15	KNN + NGram	0.979913	0.985612	0.935154	0.959720
11	Random Forest + TFIDF	0.976419	1.000000	0.907850	0.951699
9	KNN + TFIDF	0.976419	0.981884	0.924915	0.952548
5	Random Forest + CountVec	0.974672	0.996241	0.904437	0.948122
4	Decision Tree + CountVec	0.968559	0.938567	0.938567	0.938567
13	Logistic + NGram	0.968559	1.000000	0.877133	0.934545
10	Decision Tree + TFIDF	0.965939	0.956835	0.907850	0.931699
17	Random Forest + NGram	0.962445	0.996032	0.856655	0.921101
16	Decision Tree + NGram	0.953712	0.937956	0.877133	0.906526
6	NB + TFIDF	0.904803	1.000000	0.627986	0.771488
3	KNN + CountVec	0.894323	0.972527	0.604096	0.745263
12	NB + NGram	0.866376	1.000000	0.477816	0.646651

Figure 6: Accuracy Comparison of All Models and Feature Extraction Techniques

6.3 Best Performing Model

Based on the experimental results, the **SVM with TF-IDF** feature extraction emerged as the best-performing model for spam detection. This combination achieved an accuracy of approximately **99.47%**, along with perfect precision and a high recall value.

The confusion matrix for the best-performing model is shown in Figure 7. The results indicate that the model correctly classified the majority of ham and spam messages, with very few misclassifications.

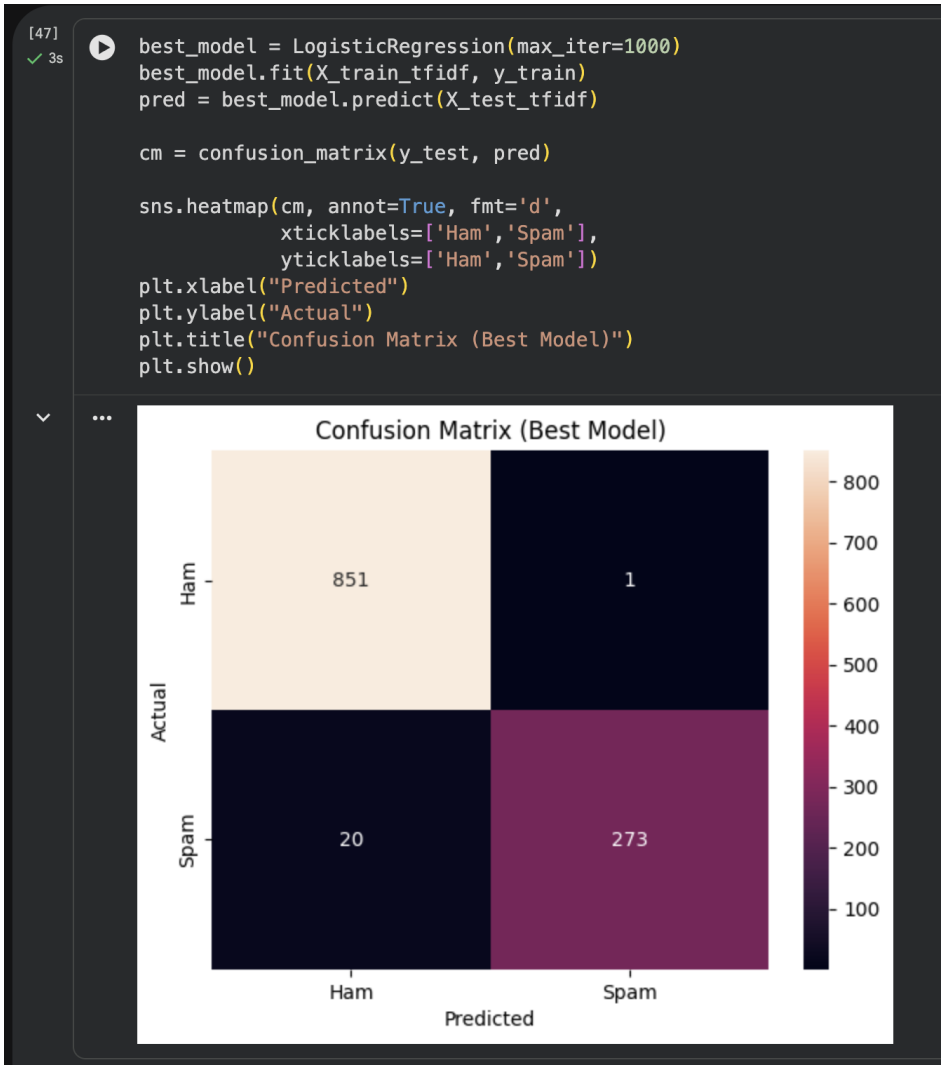

Figure 7: Confusion Matrix of Best Performing Model (SVM + TF-IDF)

7 User Interface and Implementation

The user interface of the proposed spam detection system is designed to be simple, interactive, and user-friendly. It allows users to easily enter a message and instantly receive the classification result as either *Spam* or *Ham*. The interface is integrated with the trained TF-IDF and SVM model to provide real-time predictions.

7.1 User Message Input Interface

The user message input interface serves as the primary interaction point between the user and the spam detection system. It consists of a text input area where the user can enter an SMS or email message for analysis.

The interface is designed with a clean layout, clear instructions, and a prominent action button labeled *Check Message*. Once the user enters the message and clicks the button, the input text is sent to the backend for preprocessing and classification.

Figure 8 shows the user interface where the message is entered for spam detection.

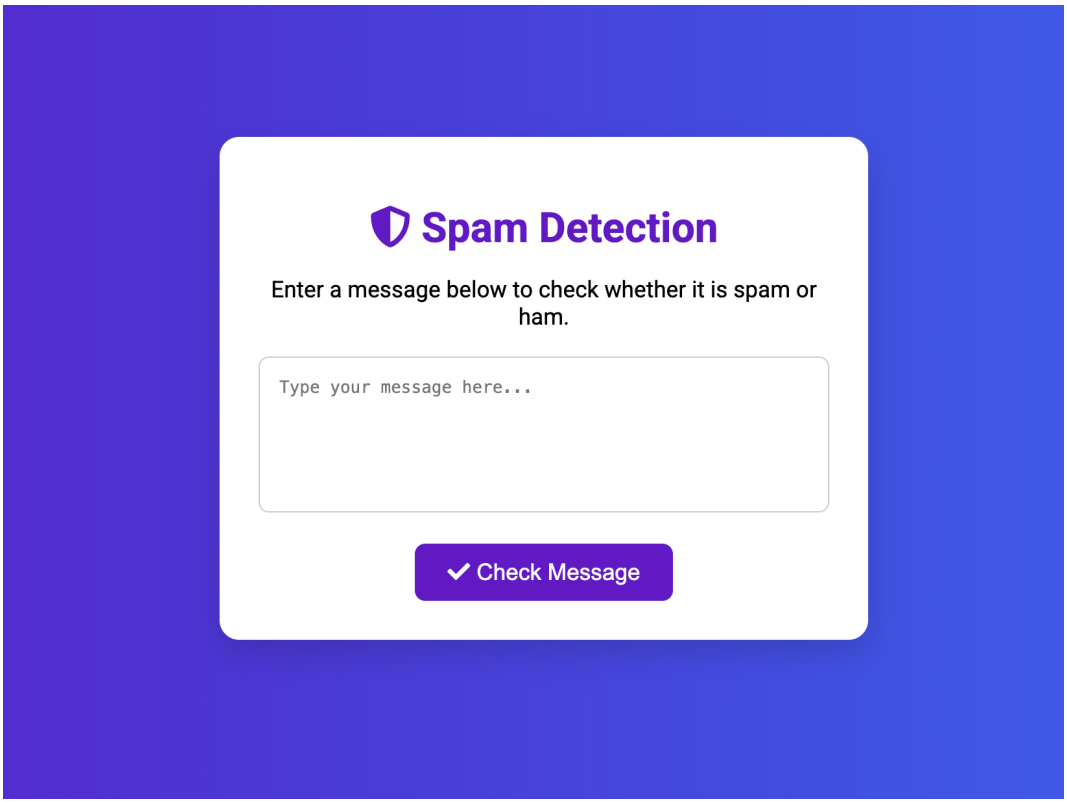

Figure 8: User Message Input Interface for Spam Detection

7.2 Live Spam Detection Output

After the user submits the message, the system processes the input in real time. The text undergoes preprocessing and is converted into numerical features using the TF-IDF vectorizer. These features are then passed to the trained Support Vector Machine (SVM) classifier.

Based on the prediction, the system displays the output clearly on the user interface. If the message is classified as spam, the result is highlighted in red, whereas a legitimate (ham) message is displayed in green. This visual distinction helps users quickly understand the classification result.

Figures 9 and 10 show examples of live spam and ham detection results generated by the system.

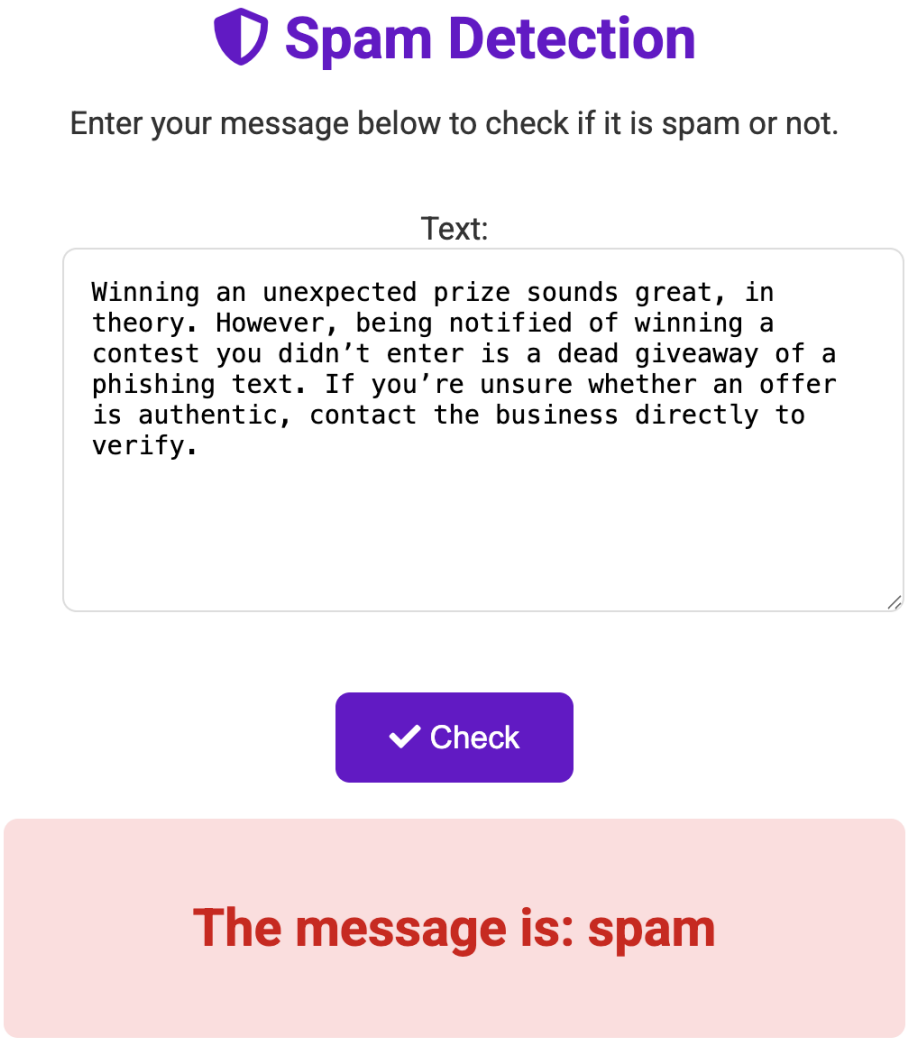

Figure 9: Live Spam Detection Output

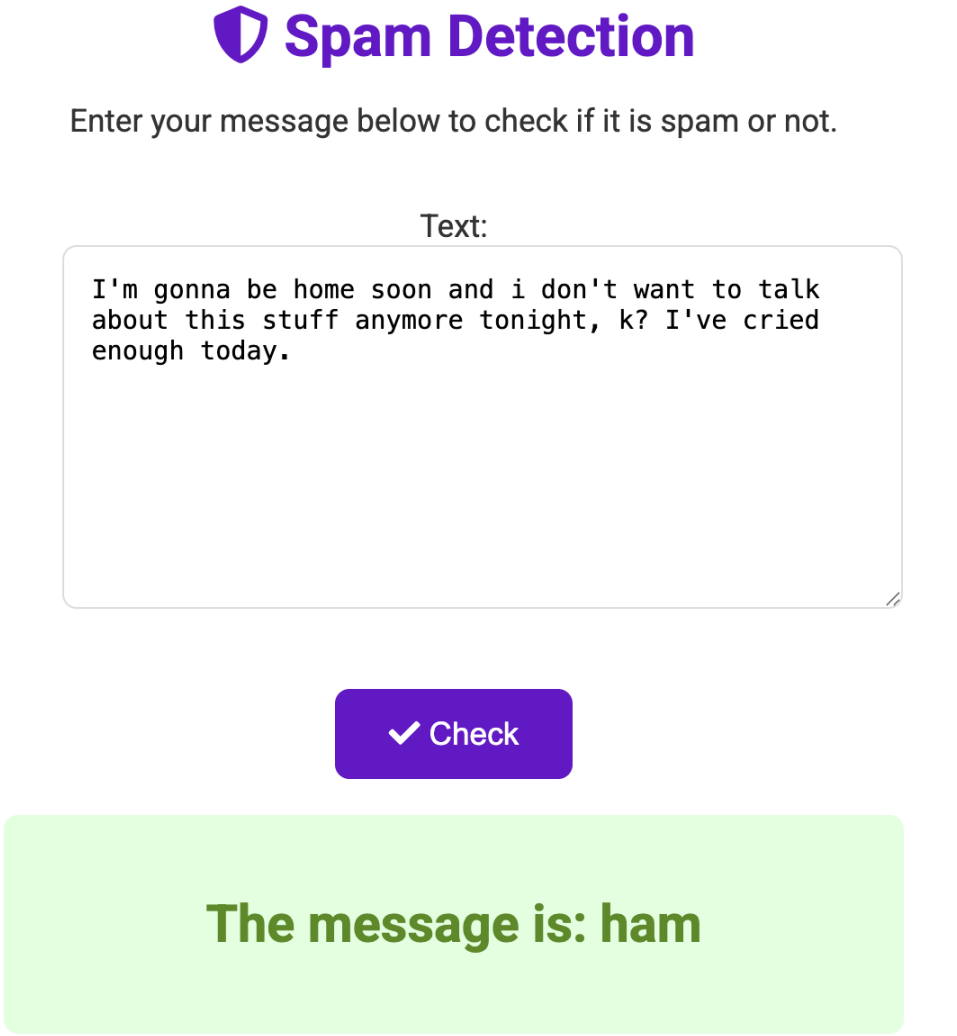

Figure 10: Live Ham Detection Output

8 Conclusion

This project presented an intelligent and efficient approach for Spam vs Ham classification using machine learning techniques. The primary objective was to accurately distinguish between spam and legitimate (ham) SMS/email messages by leveraging Natural Language Processing (NLP) methods and supervised learning algorithms. Various feature extrac-

tion techniques such as Bag of Words (Count Vectorizer), N-Gram features, and Term Frequency–Inverse Document Frequency (TF-IDF) were implemented and analyzed. Multiple machine learning models including Naive Bayes, Logistic Regression, K-Nearest Neighbors, Decision Tree, Random Forest, and Support Vector Machine (SVM) were trained and evaluated on the dataset. Experimental results demonstrated that models using TF-IDF features

consistently outperformed those using Count Vectorizer and N-Gram techniques. Among all evaluated models, the Support Vector Machine (SVM) combined with TF-IDF achieved the highest accuracy, precision, recall, and F1-score. This indicates its strong capability in handling high-dimensional and sparse text data commonly found in spam detection tasks. The

developed system also includes a user-friendly interface that allows users to input messages and receive real-time spam or ham predictions. The successful integration of the trained SVM model with the interface validates the practical applicability of the proposed system. In conclusion, the combination of TF-IDF feature extraction and SVM classification proved

to be an effective and reliable solution for spam detection. The system can be further enhanced in the future by incorporating larger and multilingual datasets, advanced deep learning models, and real-time deployment for large-scale applications.

References

1. Machine Learning for Text. A comprehensive book on machine learning applied to text data, including text classification (which is what spam vs ham does). It explains algorithms, feature extraction like TF-IDF, and classification models. Available at: `https://link.springer.com/book/10.1007/978-3-319-73531-3`

2. Kuldeep Yadav, Ponnurangam Kumaraguru, Atul Goyal, Ashish Gupta, Vinayak Naik. *SMSAssassin: Crowdsourcing Driven Mobile-Based System for SMS Spam Filtering.* Proceedings of the ACM Conference, Published on 01 March 2011. Available at: `https://dl.acm.org/doi/10.1145/2184489.2184491`

3. Deepak Dharrao, Pratik Gaikwad, Shailesh V. Gawai, Anupkumar M. Bongale, Kishan Patel, Aniket Singh. *Classifying SMS as Spam or Ham: Leveraging NLP and Machine Learning Techniques.* International Journal of Safety and Security Engineering, Published on 29 February 2024. Available at: `https://www.iieta.org/journals/ijsse/paper/10.18280/ijsse.140128`