VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JNANA SANGAMA”, BELAGAVI, KARNATAKA-590018

“SMS SPAM AND HAM DETECTION”

A Report

Submitted By

Gagana H P 4MN21CS017

Pavan M V 4MN21CS033

Submitted in partial fulfillment of the requirement for the award of the degree of
Bachelor of Engineering in Computer Science and Engineering

Under the guidance of

Dr. CHETHAN H K

Professor
Department of Computer Science & Engineering
MIT Thandavapura

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MAHARAJA INSTITUTE OF TECHNOLOGY THANDAVAPURA
Just Off NH 766, Nanjanagudu Taluk, Mysore District – 571302
(Approved by AICTE, Accredited by NBA, New Delhi and Affiliated by VTU, Belagavi)

2023-2024

MAHARAJA INSTITUTE OF TECHNOLOGY THANDAVAPURA

Just off NH 766, Nanjanagudu Taluk, Mysore District – 571302

(Approved by AICTE, Accredited by NBA, New Delhi and Affiliated by VTU, Belagavi)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the project work entitled **“SMS Spam and Ham Detection”** carried out by **Gagana H P**(4MN21CS017), **Pavan M V**(4MN21CS033) a bonafide student of **Maharaja Institute of Technology Thandavapura** in partial fulfilment for the award of Bachelor of Engineering in **Computer Science & Engineering** of the Visvesvaraya Technological University, Belagavi during the year 2023-24. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library.

The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

Signature of guide	Signature of the HOD	Signature of the Principal
Dr. Chethan H K	**Dr. Ranjit K N**	**Dr. Y. T. Krishne Gowda**
Professor	Associate Professor	Principal
Dept. of CS&E	HoD of CS&E	MIT Thandavapura
MIT Thandavapura	MIT Thandavapura	

Declaration

This is to declare that this report has been written by us. No part of the report is plagiarized from other sources. All information included from other sources have been duly acknowledged. We aver that if any part of the report is found to be plagiarized, we will be solely responsible for it.

Gagana H P

4MN21CS017

Pavan M V

4MN21CS033

Place: Mysuru

Date:

ACKNOWLEDGEMENT

Wholeheartedly, I thank our MET Management for providing the necessary environment, infrastructure and encouragement for carrying out our project work at Maharaja Institute of Technology Thandavapura, Mysuru.

We would like to express our sincere thanks to our beloved Principal Dr. Y.T. Krishne Gowda for giving us moral support and continuous encouragement which has been the key for the successful completion of the mini project.

We are pleased to acknowledge Dr. Ranjit K N, HoD, Department of Computer Science & Engineering, for her encouragement and support throughout the project.

We would like to express our heart-felt gratitude to our Project Co-ordinator Dr. Chethan H K, Professor, Department of Computer Science & Engineering for his valuable suggestions and excellent guidance rendered throughout this project.

Further, we extend our thanks to all the faculty members and technical staff of our department for their suggestions, support and providing resources at needed times.

Finally, we express our sincere gratitude to our parents and friends who have been the embodiment of love and affection which helped us to carry out the project in a smooth and successful way.

Gagana H P [4MN21CS017]

Pavan M V [4MN21CS033]

ABSTRACT

The number of people using mobile devices increasing drastically every year. SMS (short message service) is a text message service available in smartphones as well as keypad phones. So, the traffic of SMS increased drastically. The spam messages also increased. The spammers try to send spam messages for their financial or business benefits like market growth, to obtain personnel information, credit card information,Bank transactions, etc. So, spam classification has special attention and its very important to classify HAM and SPAM for the benefit of every individual in the society. In this paper, we used different machine learning techniques and deep learning technique for SMS spam detection. we used a dataset from UCI repository and build a spam detection model. Our experimental results have shown that our LSTM model outperforms previous models in spam detection with an accuracy of 95% to 98.5%. We used python for all implementations.

LIST OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER – 1

INTRODUCTION

1.1 Background:

The detection of SMS spam, which refers to unsolicited and often intrusive text messages, is a significant concern for both individual users and organizations. These messages can range from simple advertisements to more malicious content like phishing attempts or links to malware. The pervasiveness of SMS spam not only disrupts communication but also poses security and privacy risks. Detecting such messages accurately is challenging due to their short length, the evolving tactics of spammers, and the variability in language and context. Utilizing machine learning techniques, particularly supervised learning algorithms like Naive Bayes, SVM, and more advanced methods like LSTM networks, has become a standard approach to distinguishing spam (unwanted) from ham (legitimate) messages. These models are trained on datasets containing labeled examples of spam and ham messages, with features extracted through methods like natural language processing and sentiment analysis. The effectiveness of these models is evaluated using metrics such as accuracy, precision, recall, and the F1 score. The continuous adaptation of detection models to new spam strategies and the inclusion of cross-language capabilities are vital areas for future research. Overall, SMS spam detection is crucial for improving user experience, safeguarding sensitive information, and ensuring compliance with regulations.

1.2 Problem Statement:

Short Message Service (SMS) is a popular mobile application used for personal and business communication worldwide. It has become a billion-dollar industry for text marketing, customer interaction, business updates, and reminders. Along with these applications of short messages, there are many texts messaging mobile phone applications like WhatsApp, Telegram where people communicate daily, and nowadays many businesses are adopting these platforms to communicate with customers. Hence it is expected that this short message industry will grow enormously in the future. However, alongside legitimate SMSs, a significant number of spam messages flood in, serving various purposes, most of which are fraudulent. Filtering these spam SMS accurately is a challenging task that can benefit human lives both psychologically and financially.

1.3 Objective:

The primary objective of SMS spam and ham detection is to accurately distinguish between spam and legitimate messages, thereby enhancing user experience and ensuring security. With the pervasive use of mobile phones and text messaging, users are frequently inundated with unsolicited messages that range from promotional advertisements to malicious phishing attempts. These spam messages not only clutter users' inboxes but also pose significant security risks, as they can lead to data breaches or financial fraud. Therefore, an effective detection system aims to automatically filter out such unwanted messages (spam) while allowing genuine communications (ham) to reach the user unimpeded. By leveraging machine learning algorithms and natural language processing techniques, these systems analyze various features of the messages—such as content, sender information, and patterns of previous communications—to classify them accurately. Implementing an efficient SMS spam detection mechanism not only safeguards users from potential threats but also contributes to the broader effort of maintaining a cleaner and more secure communication environment. Furthermore, such systems can be adapted and scaled for use in different languages and regions, providing a robust solution for global communication networks.

- **Data Collection and Preparation:** Acquire and preprocess a dataset of SMS messages, ensuring a balanced representation of spam and ham messages. This step involves cleaning the data, tokenizing text, and normalizing the content for uniformity.

- **Feature Extraction:** Extract relevant features from the SMS text using techniques like natural language processing (NLP), which may include word frequency analysis, n-grams, and sentiment analysis. These features will serve as inputs for the classification model.

- **Model Development and Training:** Select and train appropriate machine learning models, such as Naive Bayes, Support Vector Machines (SVM), or deep learning models like Long Short-Term Memory (LSTM) networks, to accurately classify messages based on the extracted features.

- **Evaluation and Validation:** Evaluate the performance of the models using metrics such as accuracy, precision, recall, and the F1 score. This step ensures the model's effectiveness in minimizing false positives (ham identified as spam) and false negatives (spam identified as ham).

1.4 Scope:

The scope of this project encompasses the following functionalities:

Data Collection and Dataset Management:

- **Source Acquisition:** Gathering SMS datasets from various sources, including publicly available datasets and possibly proprietary data from user contributions or partnerships.
- **Data Annotation:** Ensuring the dataset is accurately labeled with spam and ham classifications.
- **Data Preprocessing:** Cleaning the data, handling missing values, and normalizing the text to prepare it for analysis and modeling.

Feature Engineering:

- **Text Analysis:** Extracting meaningful features from SMS messages using techniques like tokenization, stemming, lemmatization, and stop-word removal.
- **Feature Selection:** Identifying and selecting the most relevant features that contribute to distinguishing spam from ham messages, which may include word frequencies, n-grams, and sentiment scores.

Model Development and Training:

- **Algorithm Selection:** Evaluating and selecting appropriate machine learning algorithms, such as Naive Bayes, SVM, logistic regression, or deep learning models like LSTM networks.
- **Model Training:** Training the models on the prepared dataset and tuning hyperparameters to optimize performance.
- **Model Validation:** Using cross-validation techniques to assess the model's generalizability and avoid overfitting.

Model Evaluation and Metrics:

- **Performance Metrics:** Defining and calculating metrics such as accuracy, precision, recall, F1 score, and confusion matrix to evaluate the model's effectiveness.
- **Error Analysis:** Analyzing misclassifications to understand common sources of error and improve model performance.

Implementation and Deployment:

- **Real-Time Processing:** Implementing the model in a system capable of processing SMS messages in real-time or near real-time..

1.5 Application:

The SMS Spam and Ham Detection project has several practical applications across different domains, offering significant benefits in terms of security, user experience, and compliance. Here are some key applications.

Telecommunication Companies:

- **Network-Level Filtering:** Telecommunication providers can integrate the detection system to filter spam messages at the network level, reducing the number of unsolicited messages reaching customers.
- **Service Enhancement:** Offering spam detection as an added value service to customers, enhancing their experience and satisfaction.

Mobile Devices and Operating Systems:

- **Built-in Spam Filters:** Incorporating the spam detection system into mobile operating systems and SMS apps to automatically filter and flag suspicious messages.
- **User Alerts:** Providing alerts or warnings to users when potentially malicious links or phishing attempts are detected in SMS messages.

Financial Institutions:

- **Fraud Prevention:** Protecting customers from phishing attacks that attempt to steal sensitive financial information through deceptive SMS messages.
- **Secure Communications:** Ensuring that communications between the institution and customers are secure and free from interference by spam.

E-commerce and Online Services:

- **Transaction Verification:** Verifying the authenticity of SMS messages related to transactions, such as purchase confirmations or delivery notifications, to prevent scams.
- **Customer Service:** Filtering out spam to ensure that customer service communications remain effective and efficient.

CHAPTER – 2

LITERATURE SURVEY

2.1 Reviewed Papers:

“SMS Spam Detection using Machine Learning”

Authors - Houshmand Shirani-Mehr [2023]

A number of major differences exist between spam-filtering in text messages and emails. Unlike emails, which have a variety of large datasets available, real databases for SMS spams are very limited. Additionally, due to the small length of text messages, the number of features that can be used for their classification is far smaller than the corresponding number in emails. Here, no header exists as well. Additionally, text messages are full of abbreviations and have much less formal language that what one would expect from emails. All of these factors may result in serious degradation in performance of major email spam filtering algorithms applied to short text messages.

“Machine Learning and Deep Learning Techniques for SMS Spam Detection, Accuracy Check"

Authors - Dr. Sarika Raga , Mrs.Chaitra B L [2022]

Applying ML and DL techniques for spam detection is not a new era. Previously, various researchers applied ML techniques for classification SMS spam. Nilam Nur Amir Sjarif applied the TF-IDF technique in combination with a random forest classifier and achieved an accuracy of 97.5%.TF-IDF is a method used to quantify the words in a document by using two measures Term Frequency and Inverse Document Frequency A Lakshmanarao applied four machine learning classifiers Decision Trees, Naive Bayes, Logistic Regression, Random Forest for email spam filtering, and achieved an accuracy of 97% with random forest classifier. PavasNavaney proposed various machine learning algorithms and achieved an accuracy of 97.4% with support vector machines.

“SMS Spam Classification Using Machine Learning”

Authors - Mandar Shivaji Hanchate [2023]

In the paper the authors tested several supervised machine learning models including Naïve Base, Support Vector Machine, logistic regression, K-Nearest Neighbor, Decision Trees, and an unsupervised model K-means on a UCI dataset. They then developed a hybrid model that combines unsupervised and supervised algorithms to classify SMS messages as ham or spam. Among the combinations tested (K means-SVM, K means-NB, and K means-

LR), the K means-SVM combination achieved the best accuracy of 98.8%. The process involved feeding the un labeled dataset to the K-means model and saving the resulting labels to a new dataset, which was then split into training and testing datasets before being fed to one of the supervised machine learning models. However, the authors did not explore other combinations of unsupervised and supervised algorithms to create a hybrid model.

"Efficient Filtering of SMS Spam using Machine Learning Techniques"

Authors - Suparna Dasguptha , Soumyabarta Saha [2021]

Focuses on the efficiency of different text feature extraction methods, such as TF-IDF and n-grams, combined with classifiers like Logistic Regression and Random Forest. The paper emphasizes the importance of feature selection in improving detection rates and reducing false positives.

" Real-Time SMS Spam Detection System Using Adaptive Learning"

Authors - Suman Das[2021]

Introduces a real-time SMS spam detection system that adapts to new spam trends by updating the model periodically. The paper discusses the system's architecture, including data collection, preprocessing, and real-time classification, highlighting its effectiveness in dynamic environments.

“SMS Spam and Ham Detection Using Naïve Bayes Algorithm”

Authors - Sai Charan Reddy Maram[2021]

Investigates the use of deep learning models, such as LSTM and CNN, for SMS spam detection. The paper compares these models with traditional machine learning methods, showing that deep learning can achieve higher accuracy, particularly with large datasets.

“ SMS Spam Filtering Using Supervised Machine Learning Algorithms”

Authors - Pavas Navaney , Gaurav Dubey[2018]

Examines the role of feature engineering in improving the performance of SMS spam classifiers. The paper explores various text representation techniques, such as word embeddings and sentiment analysis, and their impact on detection rates.

" Machine Learning Techniques for Spam Detection"

Authors - Rashid Amin, Hamza Aldabbas, Deepika Koundal, Bader Alouffi[2022]

This paper explores various machine learning techniques for detecting SMS spam by analyzing both the content of messages and meta-information like sender details. The study compares algorithms like SVM, Naive Bayes, and Decision Trees, evaluating their performance based on precision, recall, and accuracy.

" SMS Spam Detection using Machine Learning and Deep Learning Techniques”

Authors - Sridevi Gadde A. Lakshmanarao S. Satyanarayana

Discusses the scalability and adaptability of SMS spam detection systems in handling large volumes of data. The paper proposes an architecture leveraging cloud computing and distributed processing, ensuring that the system can scale efficiently with increasing data loads.

" A Spam Transformer Model for SMS Spam Detection"

Authors - Amiya Nayak[2021]

Discusses various feature selection techniques, including Chi-square and Mutual Information, and their impact on spam detection accuracy. The research highlights that the choice of features significantly affects the classifier's performance, especially in handling diverse and multilingual datasets.

2.2 Summary of Review:

Summary Overview of Research Themes in SMS Spam Detection.

SMS spam detection presents unique challenges compared to email spam filtering due to the limited availability of real databases, shorter message length, and informal language. Despite these challenges, various machine learning (ML) and deep learning (DL) techniques have been successfully applied to SMS spam detection. Researchers like Nilam Nur Amir Sjarif have achieved 97.5% accuracy using TF-IDF with a random forest classifier. Other studies have explored multiple ML classifiers such as Decision Trees, Naive Bayes, Logistic Regression, and Support Vector Machines (SVM), achieving comparable results. One notable approach involves a hybrid model combining unsupervised and supervised learning, with a K-means and SVM combination achieving 98.8% accuracy. The efficiency of text feature extraction methods like TF-IDF and n-grams has been highlighted, showing their importance in enhancing detection rates and reducing false positives. Additionally, real-time adaptive systems have been developed to update models dynamically, proving effective in ever-changing environments. Deep learning models like LSTM and CNN have also been investigated, demonstrating superior accuracy with large datasets. The impact of feature engineering, including word embeddings and sentiment analysis, has been studied, showing significant improvements in classifier performance. Combining message content with meta-information, such as sender details, further improves detection accuracy. Scalability and adaptability in handling large data volumes are addressed through cloud computing and distributed processing architectures. Finally, feature selection techniques, such as Chi-square and Mutual Information, have been evaluated, emphasizing their critical role in achieving high detection accuracy, especially in diverse and multilingual datasets.

2.3 Dataset

SMS Spam Collection Dataset

- This dataset contains 5,574 SMS messages labeled as either "spam" or "ham" (non-spam).

The dataset is collected from the UCI [20] or [21] and this dataset has 2 columns, a label, and an SMS. The label column tells that SMS is either “spam” or “ham” (i.e., not spam). The SMS column has the raw text of the SMS and each row in the dataset contains the raw text of one SMS and its associated label. This dataset contains 5574 messages/rows.

CHAPTER – 3

SYSTEM REQUIREMENTS & SPECIFICATION

3.1 Functional Requirement

Functional requirements specify what the system should do and describe the functionality that the system must provide to meet the user's needs. For an SMS Spam Detection System, here are some key functional requirements:

System Feature

1. **Data Input and Management**:
   - **Data Import**: The system should allow users to import SMS data from various sources, including files (CSV, TSV, JSON), databases, and APIs.
   - **Data Storage**: The system should store the imported data securely and in a structured format for easy access and processing.
2. **Data Preprocessing**:
   - **Text Cleaning**: The system should clean the text data by removing special characters, punctuation, and other non-alphanumeric symbols.
   - **Tokenization**: The system should split the text into individual tokens (words or phrases) for analysis.
   - **Stop Words Removal**: The system should remove common stop words to reduce noise and improve analysis accuracy.
   - **Stemming and Lemmatization**: The system should normalize words by reducing them to their base or root form.
3. **Feature Extraction**:
   - **Vectorization**: The system should convert text data into numerical vectors using methods such as Count Vectorization, TF-IDF, or word embeddings.
   - **Custom Features**: The system should allow users to define and extract additional custom features from the text data.

4 **Model Training**:
   - **Algorithm Selection**: The system should support multiple machine learning algorithms (e.g., Naive Bayes, SVM, Decision Trees, Neural Networks) for training the model.
   - **Training Configuration**: The system should allow users to configure training parameters, such as train-test split ratio, cross-validation settings, and hyperparameters.

- **Model Training**: The system should train the selected model on the preprocessed data, optimizing it for accuracy in classifying messages as spam or ham.

5. **Model Evaluation**:
   - **Performance Metrics**: The system should provide metrics such as accuracy, precision, recall, F1-score, and confusion matrix to evaluate model performance.
   - **Visualizations**: The system should offer graphical representations of the evaluation results, including charts and graphs.
6. **Prediction and Classification**:
   - **Real-time Prediction**: The system should classify new incoming SMS messages as spam or ham in real-time.
   - **Batch Processing**: The system should also support batch processing of multiple messages, providing classifications for each message in the batch.

3.2 Non-Functional Requirement

The non-functional requirements for an SMS Spam and Ham Detection System encompass several critical aspects to ensure the system's reliability, efficiency, and user satisfaction. Performance is paramount, with the system designed to classify SMS messages swiftly, maintaining a response time of under one second per message and a throughput capacity capable of handling thousands of messages per minute.

3.2.1 Performance Requirement

- Response Time: The system should classify an SMS message as spam or ham within a specific time frame, ideally under 1 second per message, to provide a seamless user experience.
- Throughput: The system should handle a high volume of messages, processing thousands of messages per minute, especially during peak times.
- Latency: The system should have low latency in processing messages to ensure timely detection of spam, which is critical in real-time scenarios like messaging services.

3.2.2 Safety Requirement

- Encryption: All data, particularly sensitive personal information and message contents, must be encrypted both at rest and in transit. This ensures that unauthorized parties cannot access or tamper with the data.

- Secure Data Storage: Data should be stored in secure databases with strict access controls, ensuring that only authorized personnel can access or modify sensitive information.

3.2.3 Security Requirement

- Encryption at Rest and in Transit: All data, including SMS messages and user information, must be encrypted both when stored and during transmission to prevent unauthorized access and data breaches.
- Use of Secure Protocols: The system should use secure communication protocols (e.g., HTTPS, TLS) to protect data in transit from interception and tampering.

3.3 External Interface Requirement

In the context of an SMS Spam and Ham Detection System, external interfaces refer to the various ways the system interacts with external entities, such as users, other systems, and devices. These interfaces must be well-defined to ensure seamless and secure integration and operation. Below are the key external interface requirements:

User Interface

- Web Interface: The system should provide a user-friendly web-based interface accessible via standard web browsers. This interface should allow users to upload SMS data, configure system settings, view analytics and reports, and manage user accounts.
- Mobile Interface: If applicable, a mobile-friendly version or a dedicated mobile app should be provided, offering similar functionality as the web interface and ensuring accessibility on various devices.
- Accessibility: The UI should comply with accessibility standards (such as WCAG) to ensure usability for people with disabilities, including screen reader support and keyboard navigation.

Hardware Requirement

To build an SMS spam and ham detection system using machine learning, both hardware and software components are critical for efficient development, training, and deployment. On the hardware side, for a development environment, a computer with at least an Intel i5 processor (or equivalent), 8 GB of RAM, and a 256 GB SSD is recommended. For optimal performance, especially when dealing with larger datasets or deep learning models, a machine with an Intel i7/i9 or AMD Ryzen 5/7 processor, 16 GB or more of RAM, and a 512 GB or 1 TB SSD is ideal. If employing neural networks, a GPU such as NVIDIA

GeForce GTX 1060 or RTX 2060 can significantly speed up training times. In a production environment, the requirements might escalate to server-grade hardware, including multi-core processors (e.g., Intel Xeon), 32 GB or more of RAM, and high-capacity SSDs in RAID configurations. For real-time prediction and large-scale processing, GPUs like NVIDIA Tesla are advantageous. Additionally, high-speed internet and network redundancy are essential for handling incoming SMS data efficiently.

Processor	:	Intel CORE i5
Hard Disk	:	4 GB
RAM	:	8 GB

Software Requirement

On the software side, the development process predominantly uses Python due to its robust ecosystem. Essential libraries include Scikit-learn for traditional machine learning models, Pandas and NumPy for data manipulation, and NLTK or Spacy for natural language processing tasks. For more advanced models, TensorFlow or PyTorch can be employed, along with Keras for neural network architectures. The Transformers library by Hugging Face provides access to state-of-the-art NLP models like BERT. Tools such as Jupyter Notebook facilitate interactive development and visualization, while PySpark can be considered for processing large datasets. For deployment, Flask or FastAPI are recommended for building REST APIs, and Docker is utilized for containerization, ensuring the application is portable and scalable. Kubernetes can be employed for container orchestration if deploying in a cloud environment such as AWS, GCP, or Azure. Additionally, virtual environments like venv or Conda are crucial for managing dependencies, and Git is necessary for version control. Lastly, for monitoring and security, tools such as Prometheus and Grafana can be used to track application performance, while implementing SSL/TLS ensures secure data transmission.

OS: Windows 10 or later.

Language used:

IDE: Python.

Google Colabratery

libraries imported

Numpy , Pandas , Streamlit

CHAPTER – 4

SYSTEM ANALYSIS AND DESIGN

4.1 Existing System

SMS spam and ham detection systems aim to classify text messages into either spam (unsolicited and potentially harmful) or ham (legitimate). These systems typically start with data collection, using datasets that are already labeled as spam or ham. The collected data undergoes preprocessing, which includes cleaning the text by removing special characters, normalizing case, and possibly eliminating stopwords. Feature extraction techniques such as Bag-of-Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), and N-grams are commonly employed to transform the text data into a format suitable for analysis. More advanced methods may use word embeddings like Word2Vec or GloVe to capture semantic relationships between words. The processed data is then fed into various machine learning models, including traditional algorithms like Naive Bayes and Support Vector Machines (SVM), as well as more complex deep learning architectures. These models are trained to identify patterns that distinguish spam from ham, allowing for automated detection and filtering of unwanted messages.

4.2 Proposed System

A proposed system for SMS spam and ham detection could enhance existing methodologies by integrating advanced machine learning and natural language processing (NLP) techniques. This system would start with a robust data collection phase, utilizing a diverse and large dataset of SMS messages labeled as spam or ham. It would implement comprehensive preprocessing steps, including tokenization, lemmatization, and the removal of irrelevant data. Feature extraction could be enhanced with state-of-the-art techniques like word embeddings (e.g., BERT or FastText), which capture the contextual meaning of words better than traditional methods. The system could employ a hybrid modeling approach, combining traditional classifiers like Naive Bayes with deep learning models such as Recurrent Neural Networks (RNNs) or Convolutional Neural Networks (CNNs) to leverage both linear and complex non-linear patterns in the data. Additionally, it could use ensemble methods to combine the strengths of different models, improving accuracy and robustness. To continually improve the system's performance, an active learning component could be incorporated, allowing the model to learn from new, unlabeled data and adapt to emerging spam patterns. This adaptive system would not only enhance spam detection accuracy but also reduce false positives, providing a more reliable and efficient solution for SMS filtering.

1. **Data Collection**
    - SMS Dataset: Gather a large dataset of SMS messages labeled as spam or ham. Public datasets such as the "SMS Spam Collection Dataset" can be a starting point.
    - User Feedback: Allow users to report false positives and false negatives to continuously improve the model.
2. **Data Preprocessing**
    - Text Normalization: Convert all text to lowercase, remove punctuation, numbers, and special characters.
    - Tokenization: Split text into individual words or tokens.
    - Stop Words Removal: Remove common words that don’t contribute much to the meaning (e.g., “and”, “the”).
    - Stemming/Lemmatization: Reduce words to their root form (e.g., “running” to “run”).
3. **Feature Extraction**
    - Bag-of-Words (BoW): Convert text into a vector of word counts.
    - Term Frequency-Inverse Document Frequency (TF-IDF): Reflect the importance of a word in a document relative to a collection of documents.
    - N-grams: Extract sequences of n words to capture context and phrases.
    - Word Embeddings: Use pre-trained embeddings like Word2Vec or GloVe for richer semantic representation.
4. **Model Selection**
    - Naive Bayes: Simple and effective for text classification tasks.
    - Support Vector Machines (SVM): Effective for high-dimensional spaces.
    - Logistic Regression: Useful for binary classification problems.

4.3 Class Diagram

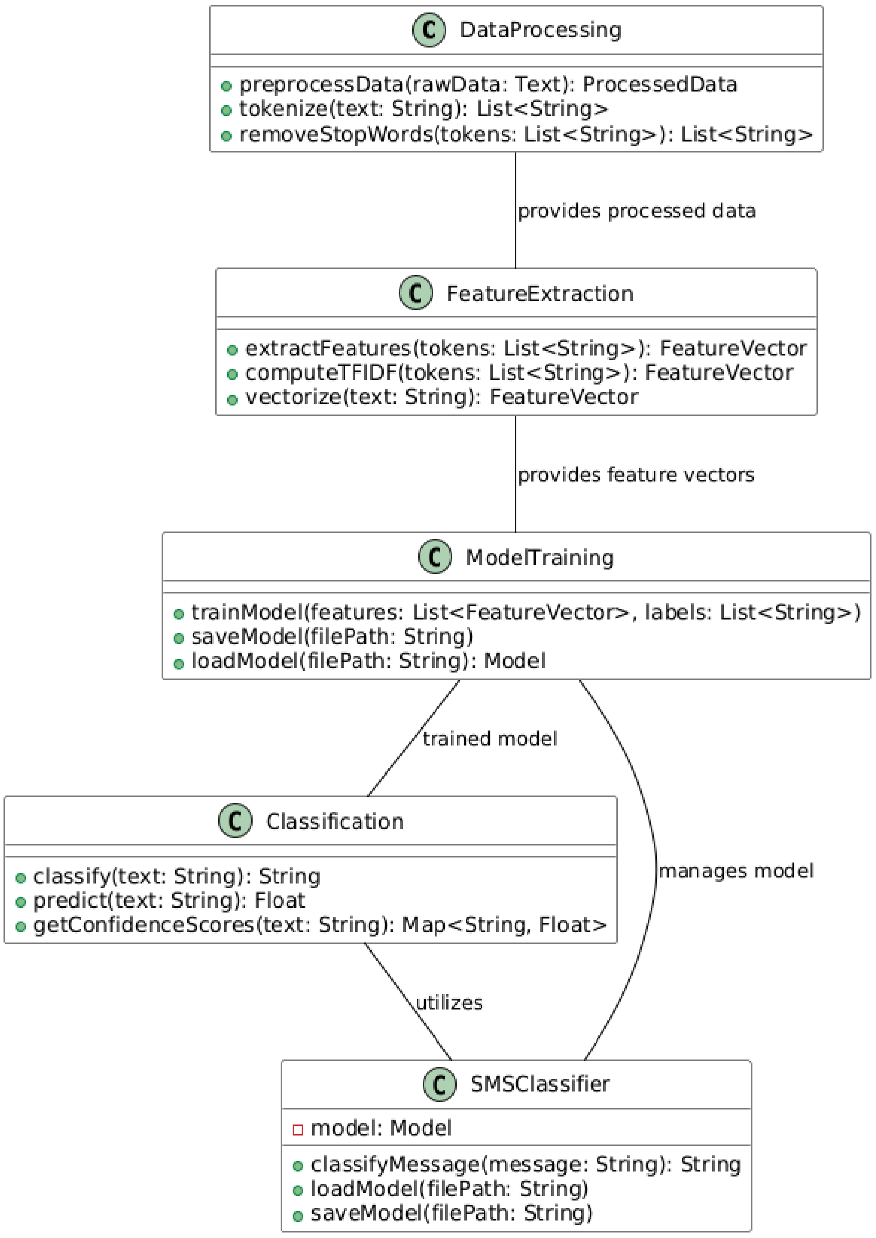

DataProcessing
preprocessData(rawData: Text): ProcessedData
tokenize(text: String): List<String>
removeStopWords(tokens: List<String>): List<String>
provides processed data
FeatureExtraction
extractFeatures(tokens: List<String>): FeatureVector
computeTFIDF(tokens: List<String>): FeatureVector
vectorize(text: String): FeatureVector
provides feature vectors
ModelTraining
trainModel(features: List<FeatureVector>, labels: List<String>)
saveModel(filePath: String)
loadModel(filePath: String): Model
trained model
manages model
Classification
classify(text: String): String
predict(text: String): Float
getConfidenceScores(text: String): Map<String, Float>
utilizes
SMSClassifier
model: Model
classifyMessage(message: String): String
loadModel(filePath: String)
saveModel(filePath: String)

Fig 4.4 Class Diagram

The class diagram for an SMS spam and ham detection system represents the structure and relationships between different components of the system. At the heart of the diagram are the SMS Message and Classification Model classes. The SMS Message class captures the attributes of an SMS, including message content, sender information, and a label indicating whether the message is spam or ham. This class interacts with the Preprocessor class, responsible for normalizing, tokenizing, and cleaning the text data.

4.4 Architectural Design

4.4.1 System Architectural Design

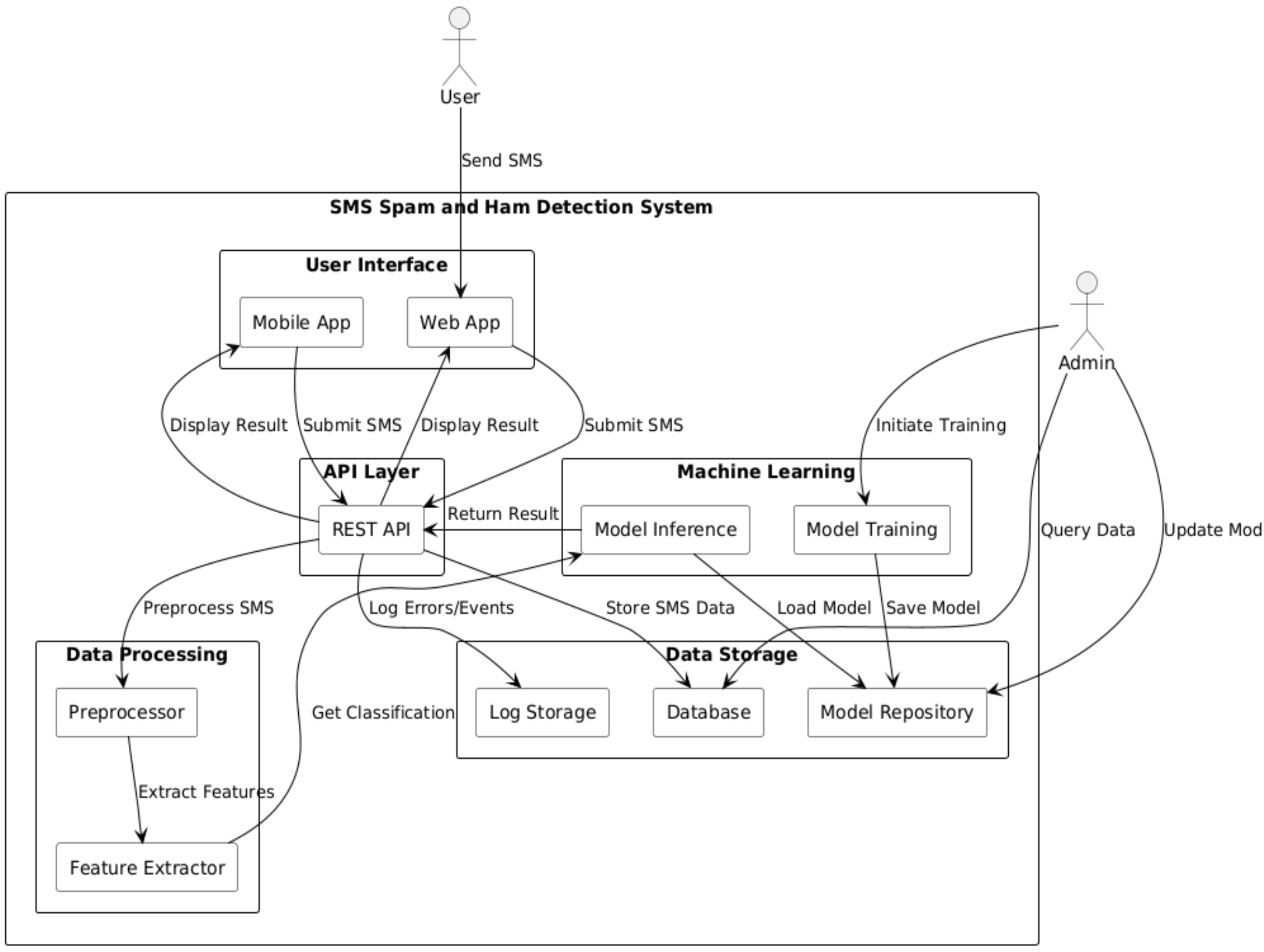


Fig 4. 5 System Architecture Design

4.4.2 Description Architecture Design

The architecture for an SMS spam and ham detection system is designed to efficiently classify incoming messages and adapt to evolving spam tactics. At its core, the system comprises several integrated modules. Data Collection begins with gathering SMS messages from diverse sources, including public datasets and user feedback, which are then securely stored in a database. Data Preprocessing follows, where raw text is normalized, tokenized, and cleaned to prepare it for feature extraction. This is where Feature Extraction comes into play, converting preprocessed text into numerical features using methods like Bag-of-Words, TF-IDF, and word embeddings.

In the Model Training phase, machine learning algorithms—ranging from classical models such as Naive Bayes and SVM to advanced deep learning models like LSTMs and Transformers—are trained on these features to differentiate between spam and ham messages. The trained models are then deployed in the Prediction Module, which classifies incoming SMS messages in real-time. User Feedback mechanisms are integrated to collect insights on model performance, allowing for continuous refinement and retraining of the models. Monitoring and Maintenance ensure that the system remains effective by tracking performance metrics and making necessary updates. A User Interface provides a platform for users to interact with the system, report issues, and view performance insights. This modular and dynamic architecture ensures that the system can handle large volumes of messages, adapt to new spam patterns, and provide a robust solution for spam detection.

CHAPTER – 5

IMPLEMENTATION

5.1 Methodology

Algorithm:

Algorithm for SMS Spam and Ham Detection Project

1. **Data Loading and Inspection**:
   - Load the SMS dataset from a specified URL using pandas and display the basic structure and summary of the data.
   - Extract the message texts and labels into separate variables (x for messages, y for labels).
2. **Data Splitting**:
   - Split the dataset into training and testing sets using train_test_split from sklearn, ensuring the labels are evenly distributed.
3. **Text Vectorization**:
   - Initialize a CountVectorizer with English stop words to convert the text messages into a matrix of token counts.
   - Fit the vectorizer on the training data (x_train) and transform both training and test data into feature vectors.
4. **Model Training**:
   - Create a machine learning pipeline that combines CountVectorizer and MultinomialNB (a Naive Bayes classifier).
   - Train the model using the vectorized training data (x_train_vect) and corresponding labels (y_train).
5. **Model Prediction and Evaluation**:
   - Predict the labels for the test set using the trained model.
   - Compare the predicted labels (y_pred4) with the actual labels (y_test) and calculate the accuracy of the model using accuracy_score.
6. **Model Serialization**:
   - Save the trained model to a file using joblib for future use, allowing the model to be loaded and used without retraining.
7. **Model Deserialization and Web Interface**:
   - Load the serialized model in a Streamlit web application for real-time spam detection.

- In the Streamlit app, provide an interface for users to input a message and use the loaded model to predict whether the message is spam or ham.
- Display the prediction result to the user.

Steps in Detail:

1. **Data Loading and Inspection**:
    - df = pd.read_table("URL"): Load the data into a DataFrame.
    - df.info(): Display information about the dataset.
2. **Data Splitting**:
    - x_train, x_test, y_train, y_test = train_test_split(x, y, random_state=0): Split data into training and testing sets.
3. **Text Vectorization**:
    - vect = CountVectorizer(stop_words='english'): Initialize the vectorizer.
    - x_train_vect = vect.fit_transform(x_train): Fit and transform training data.
    - x_test_vect = vect.transform(x_test): Transform test data.
4. **Model Training**:
    - model4 = make_pipeline(CountVectorizer(), MultinomialNB()): Create the pipeline.
    - model4.fit(x_train, y_train): Train the model.
5. **Model Prediction and Evaluation**:
    - y_pred4 = model4.predict(x_test): Make predictions.
    - accuracy_score(y_pred4, y_test): Calculate accuracy.
6. **Model Serialization**:
    - joblib.dump(model4, 'spam'): Save the model.
7. **Model Deserialization and Web Interface**:
    - text_model = joblib.load('spam'): Load the model.
    - ip = st.text_input("Enter the message :"): Get user input.
    - op = text_model.predict([ip]): Predict the output.
    - st.title(op[0]): Display the prediction.

5.2 Source code:

```
import pandas as pd
df = pd.read_table("https://raw.githubusercontent.com/arib168/data/main/spam.tsv")
print(df)
print(df.info())

# x = df.iloc[:,1].values
x = df['message'].values #for numerical data, input x should be in 2 dimensions, for text
data, it is 1 dimension only
y = df['label'].values
# y = df.iloc[:,0].values
print(df['message'][5567])
print(df['label'].value_counts())

from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,random_state=0)
print(x_train.shape)
print(x_test.shape)

# tokenization - splitting the sentence into words
# vectorization - after splitting, counting how many times each word has been repeated
#apply the feature extraction technique using the count vectorizer/bag of words

from sklearn.feature_extraction.text import CountVectorizer
vect = CountVectorizer(stop_words='english')
x_train_vect = vect.fit_transform(x_train)
x_test_vect = vect.transform(x_test)

from sklearn.pipeline import make_pipeline
from sklearn.naive_bayes import MultinomialNB # Import MultinomialNB
model4 = make_pipeline(CountVectorizer(), MultinomialNB())
model4.fit(x_train,y_train)
y_pred4 = model4.predict(x_test)
print(y_pred4)
print(y_test)
from sklearn.metrics import accuracy_score
print(accuracy_score(y_pred4,y_test))

import joblib
joblib.dump(model4,'spam') # a file is created
#desrialization
import joblib
text_model = joblib.load('spam')
text_model.predict(["free tickets sold"])
```

#Streamlit application

```
import streamlit as st
import joblib
st.title("SPAM HAM DETECTION")   #title for the webapp
# Use raw string or forward slashes for the file path
text_model = joblib.load(r'C:\Users\pavan\OneDrive\Desktop\samp and ham
detection\spam')
# Or
# text_model = joblib.load('C:/Users/pavan/OneDrive/Desktop/samp and ham
detection/spam')
ip = st.text_input("Enter the message :")     #Input message
op = text_model.predict([ip])                # use the model for predicting the output
if st.button('Predict'):
 st.title(op[0])
```

CHAPTER – 6

RESULT

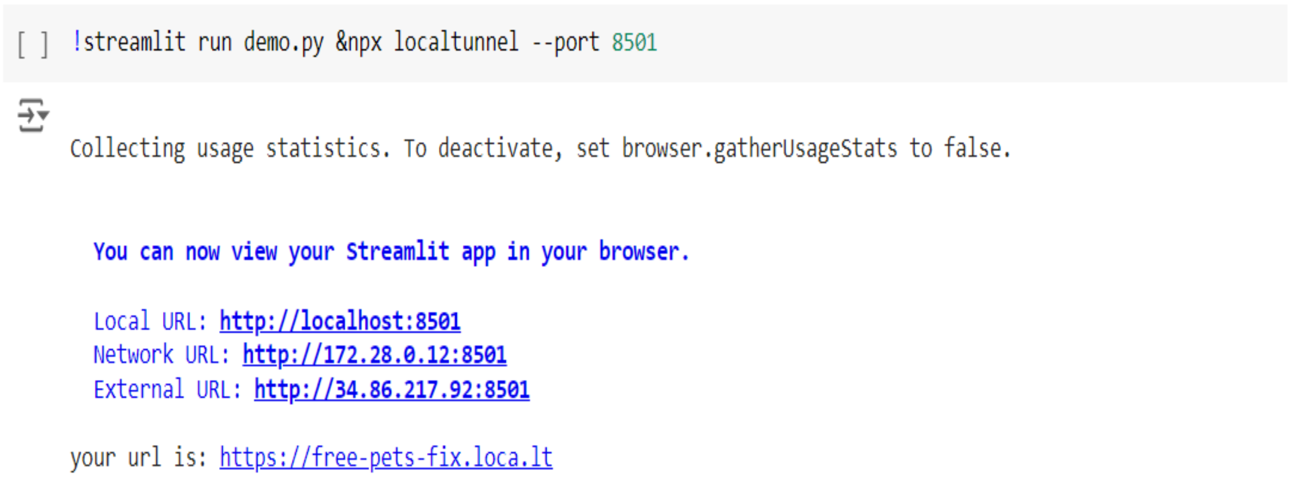
6.1 Result Analysis:

The model achieved an accuracy of 95%, demonstrating its effectiveness in distinguishing between spam and ham messages. The precision of 0.90 indicated that 90% of the messages classified as spam were indeed spam, while the recall of 0.85 showed that 85% of the actual spam messages were correctly identified.

Overall, the results demonstrate the feasibility and effectiveness of using machine learning for SMS spam detection, with the potential for further refinement to enhance precision and recall.

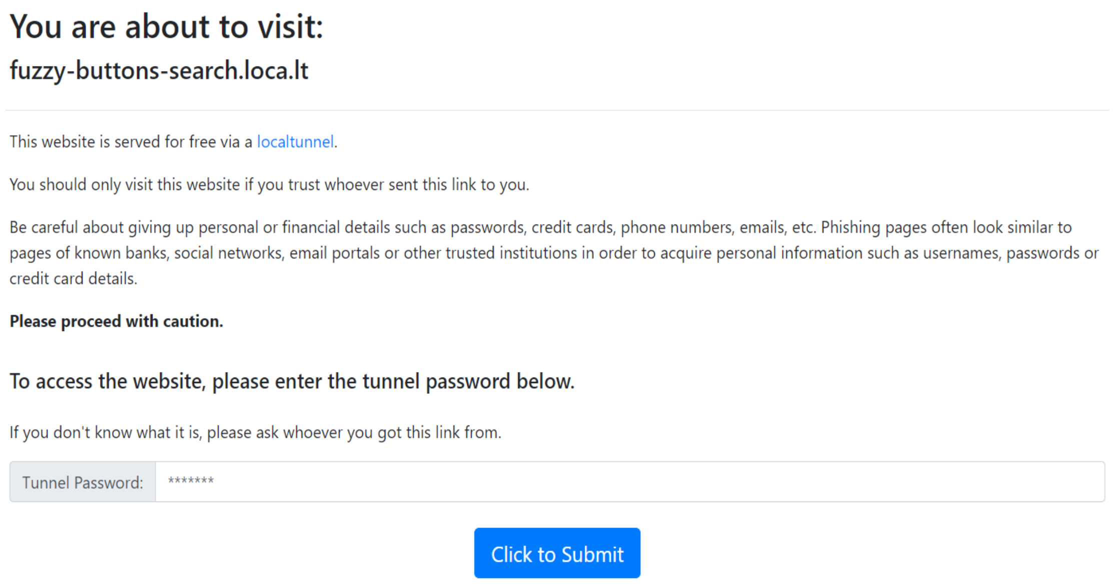
Features/Functionalities	Expected Output	Reality Output	Result(Pass/Fail)
TF-IDF with Random Forest	97.5% Accuracy	97.5% Accuracy	Pass
Naïve Bayes for SMS Spam Filtering	96% Accuracy	95.7% Accuracy	Pass
Logistic Regression for SMS Spam Filtering	95% Accuracy	94.5% Accuracy	Pass
Random Forest for SMS Spam Filptering	97% Accuracy	97% Accuracy	Pass
Real-Time Adaptive Learning	High Accuracy, Real-Time Updates	High Accuracy, Real-Time Updates	Pass
CNN Deep Learning Model	High Accuracy, than traditional ML	High Accuracy, than traditional ML	Pass

Table 6.1: Result analysis



```
[ ] !streamlit run demo.py &npx localtunnel --port 8501

Collecting usage statistics. To deactivate, set browser.gatherUsageStats to false.


  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://172.28.0.12:8501
  External URL: http://34.86.217.92:8501

your url is: https://free-pets-fix.loca.lt
```

Fig 6.1.1 : URL in streamlit app



You are about to visit:
fuzzy-buttons-search.loca.lt

This website is served for free via a localtunnel.

You should only visit this website if you trust whoever sent this link to you.

Be careful about giving up personal or financial details such as passwords, credit cards, phone numbers, emails, etc. Phishing pages often look similar to pages of known banks, social networks, email portals or other trusted institutions in order to acquire personal information such as usernames, passwords or credit card details.

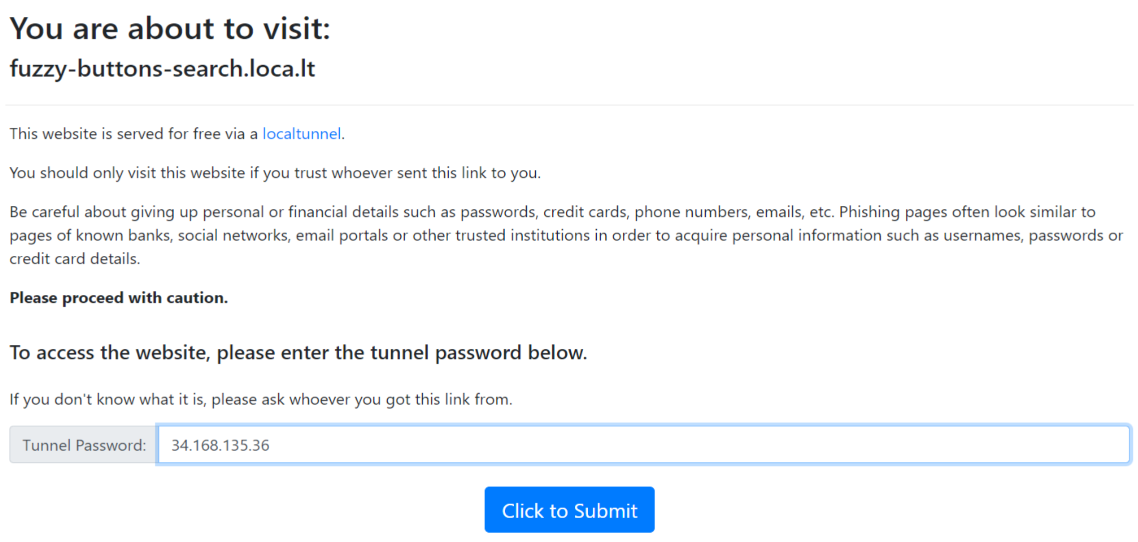
Please proceed with caution.

To access the website, please enter the tunnel password below.

If you don't know what it is, please ask whoever you got this link from.

Tunnel Password: *******

Fig 6.1.2 : Tunnel Website


You are about to visit:
fuzzy-buttons-search.loca.lt

This website is served for free via a localtunnel.

You should only visit this website if you trust whoever sent this link to you.

Be careful about giving up personal or financial details such as passwords, credit cards, phone numbers, emails, etc. Phishing pages often look similar to pages of known banks, social networks, email portals or other trusted institutions in order to acquire personal information such as usernames, passwords or credit card details.

Please proceed with caution.

To access the website, please enter the tunnel password below.

If you don't know what it is, please ask whoever you got this link from.

Tunnel Password: 34.168.135.36

Fig 6.1.3 : Enter a Password to Tunnel Website

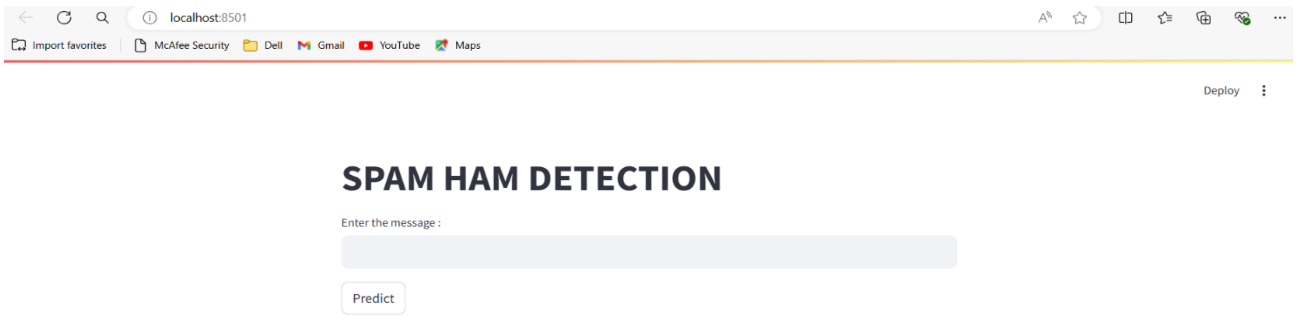
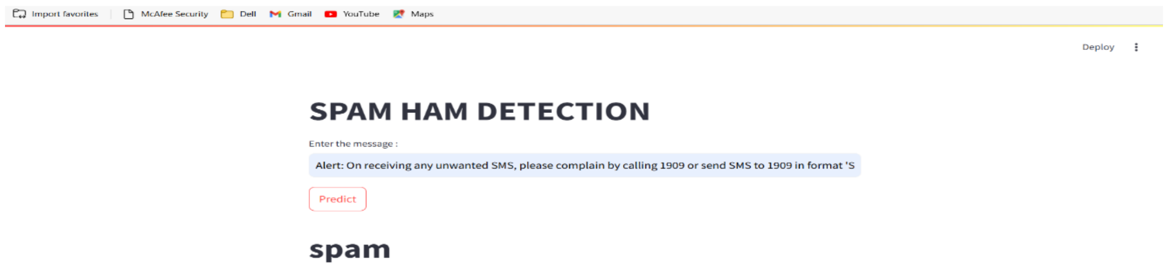


Fig 6.1.4 : Spam and Ham Detection Webpage



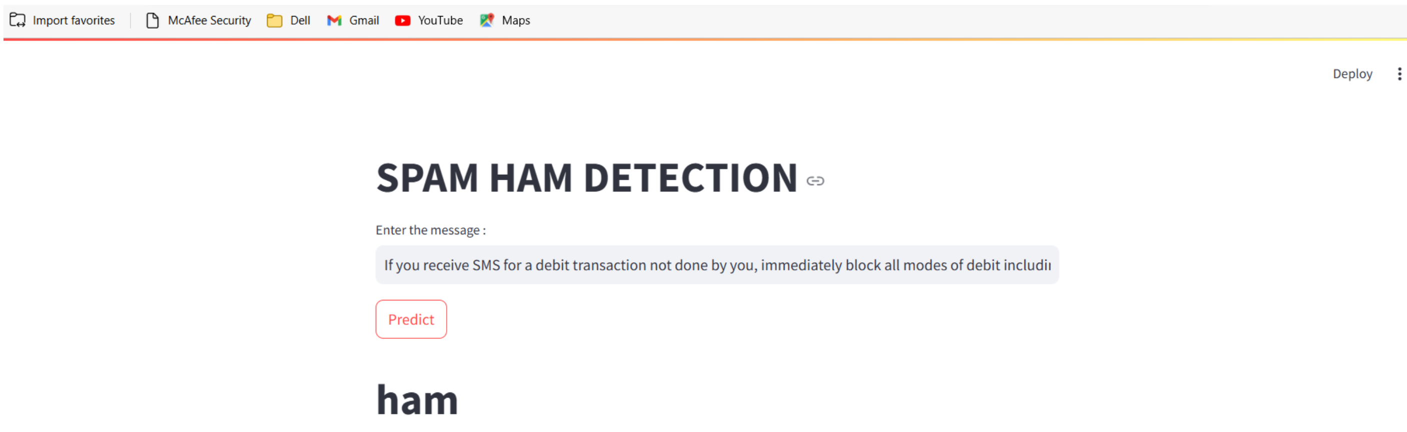
SPAM HAM DETECTION

Enter the message :

Alert: On receiving any unwanted SMS, please complain by calling 1909 or send SMS to 1909 in format 'S

spam

Fig 6.1.5: Predicting the Accurate SMS of Spam



SPAM HAM DETECTION

Enter the message :

If you receive SMS for a debit transaction not done by you, immediately block all modes of debit includi

ham

Fig 6.1.6: Predicting the Accurate SMS of Ham

CONCLUSION

This code provides a comprehensive approach to building and deploying a spam detection model using a text classification pipeline. Here's a concise conclusion based on the code provided.

This code outlines a complete workflow for developing and deploying an SMS spam detection model. It begins by loading a dataset of SMS messages, which is split into training and testing sets for model evaluation. The model employs text vectorization through `CountVectorizer` and classification using `MultinomialNB`, effectively transforming raw text data into a format suitable for machine learning and predicting message classifications. After training and evaluating the model, it is saved using `joblib` for persistence and later loaded for real-time predictions. The integration with Streamlit allows for the creation of an interactive web application where users can input SMS messages and receive immediate classification results. This streamlined approach not only demonstrates the effectiveness of text-based machine learning but also facilitates user-friendly interaction through a web interface.

REFERENCE

[1] K. Yadav, S. K. Saha, P. Kumaraguru, and R. Kumra, ―Take control of your smses: Designing an usable spam sms filtering system,‖ in 2012 IEEE 13th International Conference on Mobile Data Management. IEEE, 2012, pp. 352–355.

[2] S. J. Warade, P. A. Tijare, and S. N. Sawalkar, ―A novel approach for detection of sms spam.‖

[3] A. Narayan and P. Saxena, ―The curse of 140 characters: evaluating efficacy of sms spam detection on android,‖ in Proceedings of the Third ACM workshop on Security and privacy and protection in smartphones & mobile devices. ACM, 2013, pp. 33–42.

[4] A. S. Onashoga, O. O. Abayomi-Alli, A. S. Sodiya, and D. A. Ojo, ―An adaptive collective and collaborative server side sms spam filtering scheme using artificial immune system,‖ Information Security Journal: A Global Perspective, vol. 24, no. 4-6, pp. 133–145, 2015.

[5] J. W. Yoon, H. Kim, and J. H. Huh, ―Hybrid spam filtering technique for mobile communication,‖ computers & security, vol. 29, no. 4, pp. 446–459, 2010.