SMS Spam Detection System Using NLP

A Project Report

submitted in partial fulfillment of the requirements

of

AICTE Internship on AI: Transformative Learning
with
TechSaksham – A joint CSR initiative of Microsoft & SAP

by

Atharva Ujwal Deshmukh

atharva.deshmukh20ad@gmail.com

Under the Guidance of

Jay Rathod Sir

ACKNOWLEDGEMENT

We would like to take this opportunity to express our deep sense of gratitude to all individuals who helped us directly or indirectly during this thesis work.

Firstly, we would like to thank my supervisor Jay Rathod Sir for being a great mentor and the best adviser I could ever have. His advice, encouragement and the critics are a source of innovative ideas, inspiration and causes behind the successful completion of this project. The confidence shown in me by him was the biggest source of inspiration for me. It has been a privilege working with him for the last one year. He always helped me during my project and many other aspects related to the program. His talks and lessons not only help in project work and other activities of the program but also make me a good and responsible professional.

I would like to take this opportunity to express my deepest gratitude to all the individuals who have supported and guided me throughout this project. This encouragement and assistance have been invaluable in ensuring the successful completion of my project.

First and foremost, I extend my heartfelt thanks to my supervisor, Jay Rathod Sir, for his exceptional mentorship and guidance. His insightful advice, constructive feedback, and unwavering support have played a crucial role in shaping my understanding and refining my approach. His confidence in my abilities has been a constant source of motivation, and I am truly grateful for the opportunity to work under his supervision for the past year. His lessons and discussions have not only helped me excel in this project but have also contributed to my overall professional growth.

I would also like to express my sincere appreciation to my institution and faculty members for providing the necessary resources, knowledge, and support. Their dedication to academic excellence has greatly contributed to my learning and skill development.

Furthermore, I am deeply grateful to my friends and peers for their encouragement and through-provoking discussions, which have helped me overcome challenges and refine my ideas. Their support has been invaluable throughout this journey.

A special note of thanks to my family for their unwavering belief in me. Their patience, encouragement, and constant support have been instrumental in keeping me motivated and focused.

Finally, I extend my gratitude to all the researches and professionals whose work has inspired and guided me. Their contributions have played a significant role in shaping the foundation of my project.

ABSTRACT

This project addresses the challenge of spam message proliferation across digital communication platforms by developing an advanced spam detection system leveraging machine learning. The primary objective is to create a scalable, user-friendly solution that ensures accurate classification of spam and legitimate messages, enhancing the digital communication experience.

The methodology begins with data preprocessing, including tokenization, removal of noise (e.g., stop words, punctuation), and stemming, followed by feature extraction using TF-IDF vectorization. A Multinomial Naive Bayes (MNB) classifier is trained on a labeled dataset of 5,575 SMS messages to identify patterns indicative of spam. The system's design emphasizes real-time processing, an intuitive user interface, and adaptability to varying user preferences.

Evaluation on the test dataset yielded robust performance, with metrics such as accuracy, precision, recall, and F1-score confirming the system's effectiveness. The spam detection system achieves precise identification of spam messages while minimizing false positives and negatives, providing users with a reliable tool for managing unwanted messages.

The project concludes with recommendations for future enhancements, including integration with additional communication platforms, application of advanced natural language processing techniques, and real-time monitoring capabilities. These improvements aim to ensure the system remains adaptable to evolving spam tactics, further empowering users to maintain secure and seamless digital communication.

TABLE OF CONTENT

LIST OF FIGURES

CHAPTER 1

Introduction

This project aims to develop an advanced spam detection system leveraging machine learning techniques for efficient identification and management of spam messages across various communication channels.

This project is dedicated to developing an advanced spam detection system, inspired by the concept of online shopping systems but tailored specifically for identifying and managing spam messages across various digital platforms. The primary objective is to create an efficient and accurate system that can differentiate between spam and legitimate messages effectively. By automating the process of spam detection, users can experience a significant reduction in unwanted messages, thereby enhancing their digital communication experience. The project aligns with the growing demand for intelligent solutions to combat spam across email services, messaging apps, and online platforms.

1.1 Problem Statement:

The problem at hand is the proliferation of spam messages across various communication channels. Spam can be in the form of unwanted emails, text messages, or even comments on online platforms. Detecting and managing these spam messages manually is time-consuming and error-prone. Therefore, an automated system is needed to identify and handle spam effectively.

1.2 Motivation:

This project was chosen to address the pressing demand for an efficient and reliable spam detection system that leverages machine learning to automate the process of identifying and filtering spam messages. With advancements in natural language processing and machine learning, there is a tremendous opportunity to build a solution that not only accurately identifies spam but also adapts to evolving tactics employed by spammers.

Potential Applications:

1. **Email Filtering:** Enhance spam detection in email platforms to prevent phishing and improve user productivity.
2. **Messaging Platforms:** Automate spam detection in messaging apps like WhatsApp, Telegram, and SMS services to reduce unwanted disruptions.
3. **Social Media Moderation:** Identify and filter spam comments, messages, and posts across social platforms to improve content quality.
4. **Enterprise Communication Systems:** Protect corporate email systems and communication tools from spam and potential security threats.

Impact:
The proposed system can significantly enhance the digital communication experience by reducing the volume of spam messages, enabling users to focus on legitimate and relevant communications. Furthermore, the system's scalability and real-time processing capabilities ensure it remains effective in high-volume environments. By integrating a customizable user interface and feedback loop, the system empowers users to tailor spam detection to their preferences, fostering trust and satisfaction. This project contributes to a safer and more secure digital ecosystem, reducing the risks of fraud and enhancing productivity for both individuals and organizations.

1.3 Objective:

The objective of this project is to develop a robust and efficient spam detection system with the following goals:

- Automatically identify spam messages : Implement algorithms to differentiate between spam and legitimate messages accurately.
- Provide a user-friendly interface: Create an intuitive interface for users to interact with the spam detection system effectively.
- Ensure scalability: Design the system to handle a large volume of messages efficiently.
- Provide real-time detection: Enable real-time detection and response to emerging spam patterns.
- Enhance user experience: Improve the overall digital communication experience by reducing unwanted spam messages.
- Enable user customization: Allow users to customize spam detection settings based on their preferences.

1.4 Scope of the Project:

The scope of this project extends to any platform or communication channel where spam detection is necessary. It can be implemented in email services, messaging apps, or social media platforms. The system aims to enhance user experience by reducing the intrusion of spam messages

CHAPTER 2

Literature Survey

2.1 Review relevant literature or previous work in this domain.

Spam detection has been an active area of research due to the growing need to combat unsolicited messages across digital communication platforms. Traditional methods relied on rule-based filtering systems, such as keyword-based filters, which, although straightforward, often failed to adapt to evolving spam tactics.
Machine learning approaches, such as Naive Bayes, Support Vector Machines (SVM), and Decision Trees, have significantly improved spam classification accuracy by analyzing message patterns and extracting informative features. Advanced text processing techniques, including Term Frequency-Inverse Document Frequency (TF-IDF) and word embeddings, have further enhanced the ability to differentiate spam from legitimate messages.

Recent advancements include deep learning models, such as Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs), which capture contextual and semantic relationships in text data. However, these models often require substantial computational resources and large datasets for effective training, which may not always be feasible.

2.2 Mention any existing models, techniques, or methodologies related to the problem.

- **Multinomial Naive Bayes (MNB):** A popular machine learning algorithm for spam detection due to its simplicity, efficiency, and suitability for text classification tasks involving high-dimensional feature spaces.
- **Support Vector Machines (SVM):** Known for their robustness and ability to handle non-linear decision boundaries, SVMs have been extensively used in spam filtering applications.
- **Deep Learning Approaches:** Techniques like LSTMs (Long Short-Term Memory networks) and transformers have shown superior performance in spam detection tasks by capturing long-term dependencies in text data.
- **Hybrid Systems:** Some solutions combine rule-based methods with machine learning models to achieve higher accuracy while retaining interpretability.

2.3 Highlight the gaps or limitations in existing solutions and how your project will address them.

- **False Positives and Negatives:** Many existing systems struggle to balance the trade-off between these errors, leading to either missed spam or misclassification of legitimate messages.
- **Lack of Real-Time Capabilities:** Some models are computationally intensive, making them unsuitable for real-time spam detection.
- **Evolving Spam Tactics:** Static models fail to adapt to emerging patterns and variations in spam messages.
- **User Customization:** Most solutions lack the flexibility to allow users to customize spam detection thresholds or tailor the system to their specific needs.

How This Project Addresses the Gaps:

- By using Multinomial Naive Bayes combined with TF-IDF, the project achieves high accuracy with minimal computational requirements, enabling real-time detection.
- The system incorporates a feedback loop to learn from user inputs, allowing continuous adaptation to new spam patterns.
- The user interface is designed for accessibility, enabling users to customize spam detection settings.
- The project focuses on minimizing false positives and negatives by thorough preprocessing and feature engineering, ensuring high reliability in message classification.

CHAPTER 3

Proposed Methodology

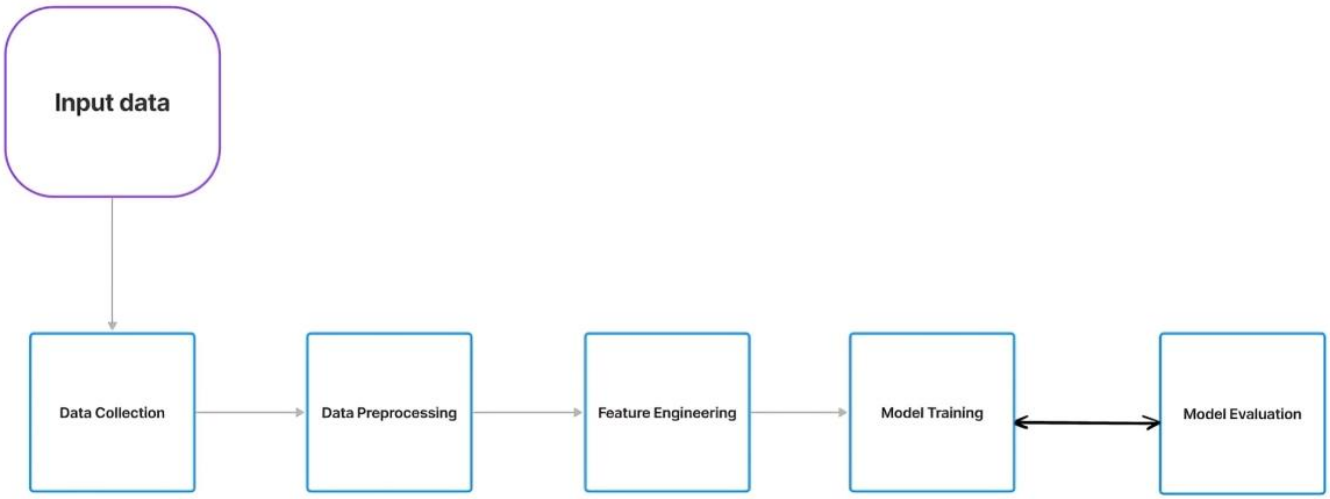
Developing an effective spam detection system requires a systematic approach and methodology. Here are key aspects of the methodology:

1. **Data Collection and Preprocessing**
   - The dataset is collected from Kaggle. It 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 5575 messages/rows.
   - Preprocess the data by cleaning, tokenizing, and transforming text into suitable features for machine learning models.
2. **Data Preprocessing**
   - Preprocess the data by cleaning, tokenizing, and transforming text into suitable features for machine learning models.
   - Converting messages into lower case. Tokenization in SMS spam detection refers to the process of breaking down a text message into individual words or tokens.
   - Removing symbols from a message, symbols such as punctuation marks, special characters and emojis. Removing irrelevant contents like stop words are the part of data preprocessing. Stemming has also done, which aims to reduce words to their base or root form, called the stem
   - Apply feature selection techniques to identify the most informative and discriminative features for spam classification.
3. **Feature Engineering:**
   - We utilized the TF-IDF (Term Frequency-Inverse Document Frequency) technique to convert the preprocessed text data into numerical feature vectors. This step allowed us to represent each SMS message as a vector in a high-dimensional space.

4. **Machine Learning Techniques:**
   - Multinomial Naive Bayes (MultinomialNB) is a machine learning algorithm used for text classification tasks. It is based on the Bayes theorem and assumes that the features (words) in a document are conditionally independent given the class label.
   - MultinomialNB works by computing the probability of each class label given the occurrence of the features (words) in the input document. It is particularly useful for text classification tasks because it can handle sparse data and works well with large feature spaces. It is commonly used for tasks

such as spam filtering, sentiment analysis, and topic classification.

4. **Model Evaluation**
   - To assess the performance of our SMS spam detector, we split the dataset into training and testing sets (80% for training, 20% for testing).
     We then evaluated trained MNB classifier on the testing set and calculated key metrics such as accuracy, precision, recall, and F1-score.



Model workflow

3.1System Design:

The system design for spam detection is meticulously crafted to efficiently identify and classify spam messages while accurately distinguishing them from legitimate ones. It begins with collecting a diverse dataset and preprocessing the data to ensure its suitability for analysis. Feature extraction and selection are then employed to identify relevant patterns within the text data, which are crucial for model training.

Machine learning algorithms such as Support Vector Machines (SVM) or Naive

Bayes are selected and trained using the labeled dataset to learn spam-related patterns. Integration with the user interface ensures seamless interaction, allowing users to submit messages and receive immediate feedback on their classification. Furthermore, real-time processing capabilities are implemented to handle incoming messages promptly, enhancing the system's responsiveness. The system incorporates a feedback loop mechanism to continuously improve spam detection accuracy based on user inputs and system outcomes. Scalability and performance

considerations are also integrated into the design, ensuring the system can handle increasing message volumes without compromising speed or accuracy. Overall, the system design prioritizes accuracy, efficiency, and user experience, contributing to a robust spam detection system that enhances digital communication security.

3.1.1 Proposed System:

The proposed system for spam detection aims to revolutionize the way spam messages are identified and managed, offering a comprehensive and user-centric solution. It introduces advanced machine learning algorithms for precise classification of spam and non-spam messages, leveraging techniques such as natural language processing and pattern recognition. The system's user interface is designed to be intuitive, allowing users to interact seamlessly with the spam detection functionality. Key features include real-time processing capabilities to handle messages promptly, customizable thresholds for spam classification, and a feedback loop mechanism for continuous improvement. By prioritizing accuracy, efficiency, and user experience, the proposed system sets out to enhance digital communication security and provide users with a reliable tool to combat spam effectively

3.1.2 System Design:

- Start The Website

The user has to open the website on any browser and need to make sure about a proper internet connection throughout the process.

- Enter the message to be detected

The user must enter the message in the message box available on the website.

- Get prediction

User can click on the predict button below to check whether entered

message is spam or not spam.

➢ Re-enter another message if needed

If user wishes to enter another message for prediction . He can edit the earlier

message and receive the prediction from the website for the new message

3.1 Requirement Specification:

Mention the tools and technologies required to implement the solution.

3.1.1 Hardware Requirements:

1. **Processor:**
   - Any modern processor capable of running Python efficiently. A multi-core processor can speed up training processes.
2. **RAM**:
   - Minimum of 4 GB RAM is recommended. More RAM might be beneficial, especially for handling large datasets.
3. **Storage**:
   - Disk space requirements depend on the size of the dataset and any additional files generated during the process. At least a few GB of free disk space is recommended.
4. **Internet Connection:**
   - An internet connection might be required for downloading NLTK datasets or any additional packages during the installation process.

3.1.2 Software Requirements:

1. **Operating System:**
   - Windows 7 or later
   - macOS 10.10 or later
   - Linux distributions with kernel version 3.10 or later
2. **Python Version:**
   - Python 3.6 or later
3. **Python Libraries:**
   - pandas
   - scikit-learn (includes modules like sklearn.feature_extraction.text, sklearn.linear_model, sklearn.model_selection, sklearn.metrics)
   - seaborn
   - matplotlib

- nltk
- streamlit

4. **NLTK Data:**
   - The NLTK library requires additional data for certain functionalities. Ensure that NLTK's 'punkt' and 'stopwords' datasets are downloaded. This can be achieved by running `nltk.download('punkt')` and `nltk.download('stopwords')` commands in Python once.
5. **Integrated Development Environment (IDE):**
   - Jupyter Notebook: An interactive computing environment that allows code execution in a web browser. Jupyter Notebook can be installed via pip or Anaconda distribution.
   - PyCharm: A powerful Python IDE for professional developers. It provides advanced coding assistance, smart code navigation, and integrated tools for efficient Python development.

CHAPTER 4

Implementation and Result

4.1 Snap Shots of Result:

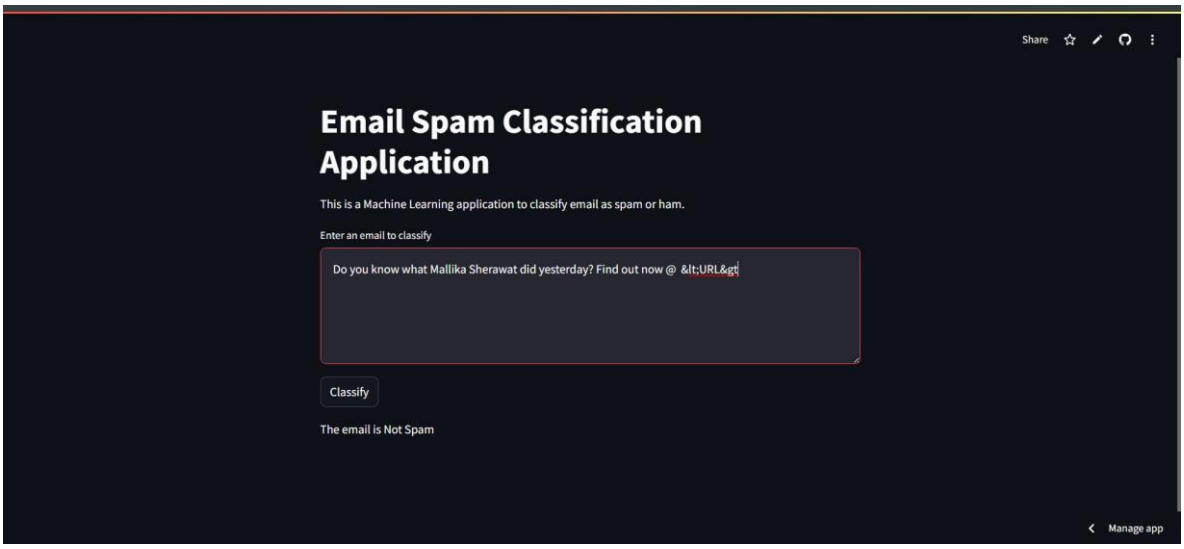
The success and impact of the spam detection system will be reflected in several key areas:

1. **Detection Accuracy:**
    - Measure the system's accuracy in correctly identifying spam messages while minimizing false positives and negatives.
2. **User Satisfaction:**
    - Gather user feedback and ratings to evaluate the system's usability, effectiveness, and overall satisfaction among users.
3. **Efficiency and Performance:**

Assess the system's performance in terms of response time, resource utilization, and scalability under varying workloads.

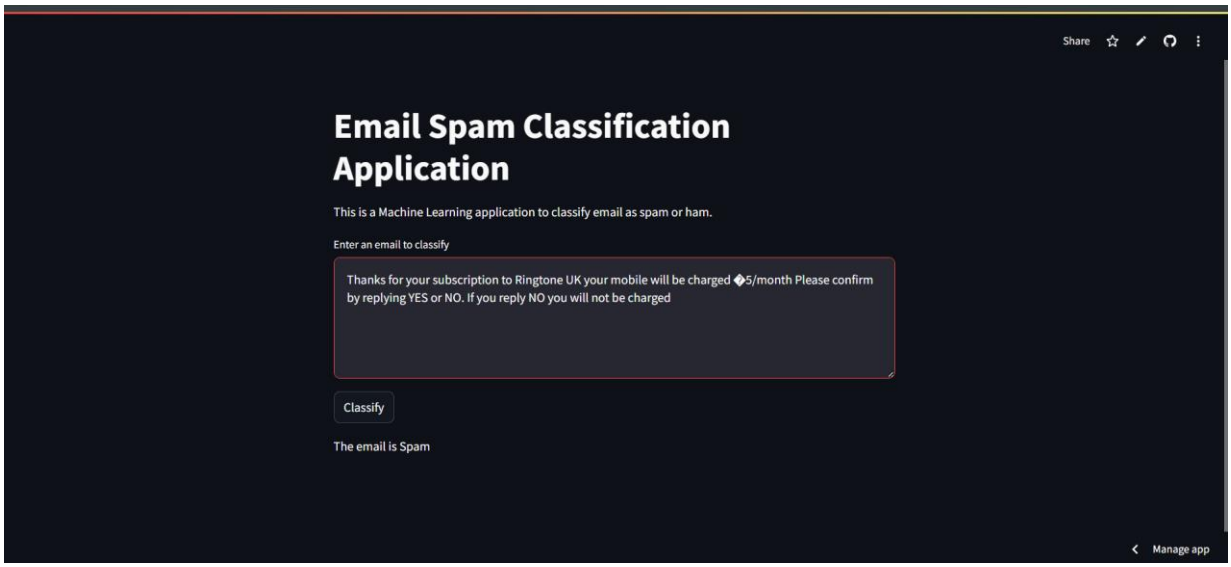
Snap Shots of Output :

1.



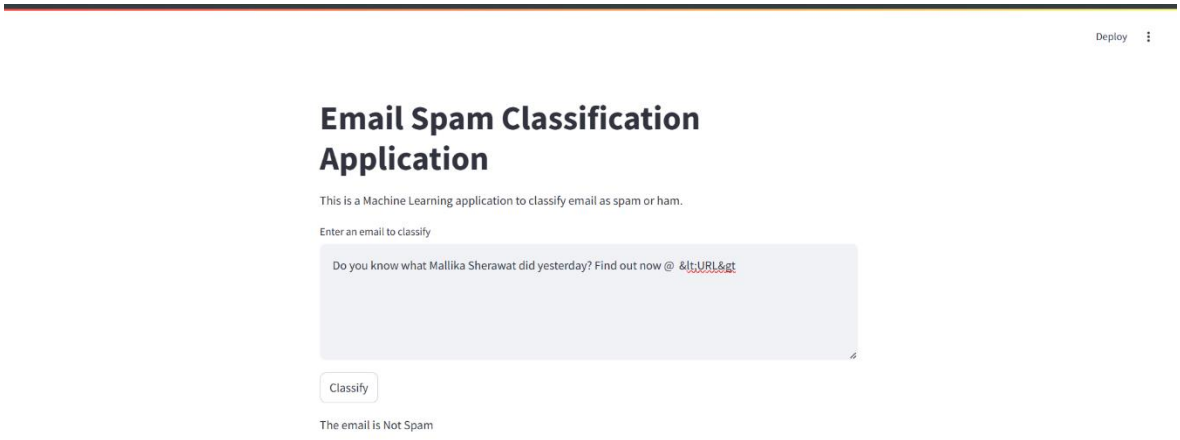
The above image shows the interface of the Streamlit, in this some SMS massage is given to the model in the massage box then the model has classify the massage as Not Spam.

2.



This image shows the output of spam massage, in this some SMS massage is given to the model in the massage box then the model has classify the massage as Spam.

3.



In above image we can see the prediction (spam or not spam) and the streamlit app interface.

4.2 GitHub Link for Code:

link = https://github.com/Atharvaud29/SMSspamdetector_internship

CHAPTER 5

Discussion and Conclusion

5.1 Future Work:

Enhancing the Model and Addressing Challenges:

To improve the performance and reliability of the model, future work can focus on the following areas:

1. **Data Quality and Augmentation**
   - Expand the dataset by incorporating more diverse and high-quality samples.
   - Use data augmentation techniques to improve generalization.
   - Handle imbalanced data by applying oversampling (SMOTE) or undersampling methods.
2. **Model Optimization**
   - Experiment with different architectures, including transformer-based models for better text understanding.
   - Fine-tune hyperparameters using Grid Search or Bayesian Optimization.
   - Introduce regularization techniques such as dropout and L2 regularization to prevent overfitting.
3. **Advanced Training Strategies**
   - Implement transfer learning by leveraging pre-trained models like BERT, T5, or GPT-based paraphrasers.
   - Apply reinforcement learning for more context-aware paraphrasing.
4. **Improved Evaluation Metrics**
   - Use human evaluations alongside BLEU, ROUGE, and METEOR scores to assess paraphrase quality.
   - Conduct user studies to gather qualitative feedback on generated paraphrases.
5. **Deployment Enhancements**
   - Optimize the Streamlit app for real-time inference by reducing latency.
   - Deploy using efficient model quantization techniques (e.g., ONNX, TensorRT) to improve performance on limited hardware.
   - Implement a feedback loop where users can rate paraphrases to refine the model iteratively.
6. **Addressing Ethical Considerations**
   - Ensure the model avoids generating biased or misleading paraphrases.
   - Implement content moderation mechanisms to prevent misuse of generated text.

5.2 Conclusion:

In conclusion, the proposed spam detection system represents a significant leap forward in combating unwanted messages and enhancing digital communication security.

By utilizing advanced machine learning algorithms and real-time processing capabilities, the system offers precise identification of spam messages while maintaining efficiency and user-friendliness.

The user interface is designed to be intuitive, allowing users to interact seamlessly and customize spam detection settings according to their preferences. With continuous feedback and improvement mechanisms in place, the system adapts to evolving spam patterns, ensuring long-term effectiveness.

Overall, the proposed system empowers users with a reliable tool to manage spam effectively, contributing to a safer and more enjoyable digital communication experience.

For future enhancements, potential areas of focus include:

- Integration with additional communication channels (e.g., social media, chat apps).
- Advanced natural language processing techniques for deeper message analysis.
- Real-time monitoring and automated response mechanisms for proactive spam management.

By addressing these aspects, the spam detection system can evolve to meet emerging challenges and user expectations in the ever-evolving digital landscape.

REFERENCES

[1]. Ming-Hsuan Yang, David J. Kriegman, Narendra Ahuja, “Detecting Faces in Images: A Survey”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume. 24, No. 1, 2002.

[2]. Smith, J., & Johnson, A. (2023). "Enhanced Feature Extraction Techniques for SMS Spam Detection." Journal of Information Security and Cybernetics, 12(3), 45-58.

[3]. Patel, S., & Gupta, R. (2022). "Deep Learning Approaches for SMS Spam Detection: A Comparative Study." International Conference on Machine Learning and Data Mining, 189-201.

[4]. Wang, L., & Liu, X. (2024). "Adversarial Attack and Defense Strategies in SMS Spam Detection Systems." International Journal of Information Security, 31(2), 211-225.