

**UNIVERSITY INSTITUTE OF COMPUTING**

**PROJECT REPORT**

**Spam Mail Filter**

Program Name: BCA

Subject Name/Code: Data Structures(24CAP-152)

**Submitted by: Submitted to:**

Name**:** Jhuna Kumar Name**:** Parwan Singh

UID: 24BCA10198 Designation: Assistant Professor

Section**:** BCA – 8 “B” Signature:-------------------



**Introduction**

Email is one of the most widely used forms of digital communication in both personal and professional environments. Despite its many benefits, the email system is often abused through the distribution of spam mails—unsolicited and irrelevant messages typically sent in bulk. These emails can range from harmless advertisements to dangerous phishing attacks and malware threats. As the volume and complexity of spam increase, it becomes more important to develop effective and intelligent spam filtering systems.Spam mails not only waste users’ time and storage space but also pose serious cybersecurity risks. Traditional rule-based filters are no longer sufficient to combat the sophisticated techniques used by spammers, such as content obfuscation, fake headers, and domain spoofing. As a result, modern spam filters must adopt more dynamic and adaptive methods that go beyond simple keyword matching or sender blocking.



**Abstract**

With the exponential growth of digital communication, email remains a primary tool for personal and professional exchange. However, this convenience is frequently exploited through spam mails—unsolicited messages often used for advertising, phishing, or distributing malware. Spam not only clutters inboxes but also poses significant security risks and reduces overall productivity. A spam mail filter is a crucial solution to this problem, aiming to detect and block unwanted emails before they reach the user.

This paper presents an overview of the design and implementation of an effective spam mail filtering system using a combination of rule-based and machine learning techniques. The system analyzes various features of an email such as sender address, subject line, content keywords, and metadata to determine its legitimacy.

Advanced filtering algorithms, such as Naive Bayes classifiers, support vector machines (SVM), and neural networks, are employed to improve detection accuracy and adapt to evolving spam tactics. Additionally, blacklisting, whitelisting, and heuristic approaches enhance the filtering process.



**Objective**

The primary objective of this project is to design and develop an effective spam mail filtering system that can accurately identify and block unwanted or harmful emails while allowing legitimate messages to reach the user's inbox.

* ****Detect and Filter Spam Emails**:** Automatically identify spam messages based on their content, sender information, and other metadata using both rule-based and machine learning techniques.
* **Minimize False Positives:** Ensure that legitimate (non-spam) emails are not mistakenly classified as spam to maintain smooth communication.



System Design

The spam mail filtering system is designed to accurately classify incoming emails as either spam or ham (non-spam) using a combination of rule-based logic and machine learning techniques.

1. Replace any references to banking with spam filtering.

2. Create a flowchart that represents the process: receiving email input, checking against spam keywords, and classifying emails as spam or legitimate.

3. Add a simple class diagram for Email Processor and Spam Detector classes.



**Implementation**

The implementation of the spam mail filter involves a step-by-step approach that combines data preprocessing, feature extraction, model training, and deployment. The process is designed to ensure accuracy, efficiency, and adaptability in identifying spam emails.

1. **Modules**: Describe the core modules of the spam filter, including a Spam Detector class that uses keyword matching and hash maps.

2. Include some brief code snippets if possible, to demonstrate the structure of these modules.



Techniques

1. **Initializing Vector with Spam Keywords**:

 The spam Keywords vector is initialized directly within the constructor using an initializer list. This allows us to

manage the list of keywords easily and expand it without changing the program's logic.

2. **Converting Email to Lowercase**:

 To make the keyword search case-insensitive, the transform() function is used to convert all characters in the email

to lowercase. tolower is passed as a function pointer to transform() to perform the conversion, which simplifies

case-insensitive comparisons.

3. **Checking for Spam Keywords with find()**:

 The program uses a for loop to iterate through each keyword in spam Keywords. It checks if the keyword exists

within lower Email using find(). If a keyword is found, it returns true immediately (early exit), classifying the

email as spam. This is efficient because it stops further checks as soon as a match is found.

4. **Display Function with for Loop**:

 Display Keywords() iterates through each keyword in spam Keywords using a range-based for loop to print the

list of keywords. This enhances readability by listing the keywords directly to the user.

5. **User Input with getline()**:

 The getline() function is used to allow the user to input an entire line as the email message, handling spaces and

special characters properly.

6. **Infinite Loop with while (true)**:

 An infinite loop (while (true)) is used to continuously prompt the user for emails until they type "exit". This loop

allows the program to keep running and classifying emails without restarting.

7. **Conditional Statements for Classification**:

 The main loop contains an if-else structure to classify each email as "SPAM" or "NOT SPAM" based on the result

of is Spam()



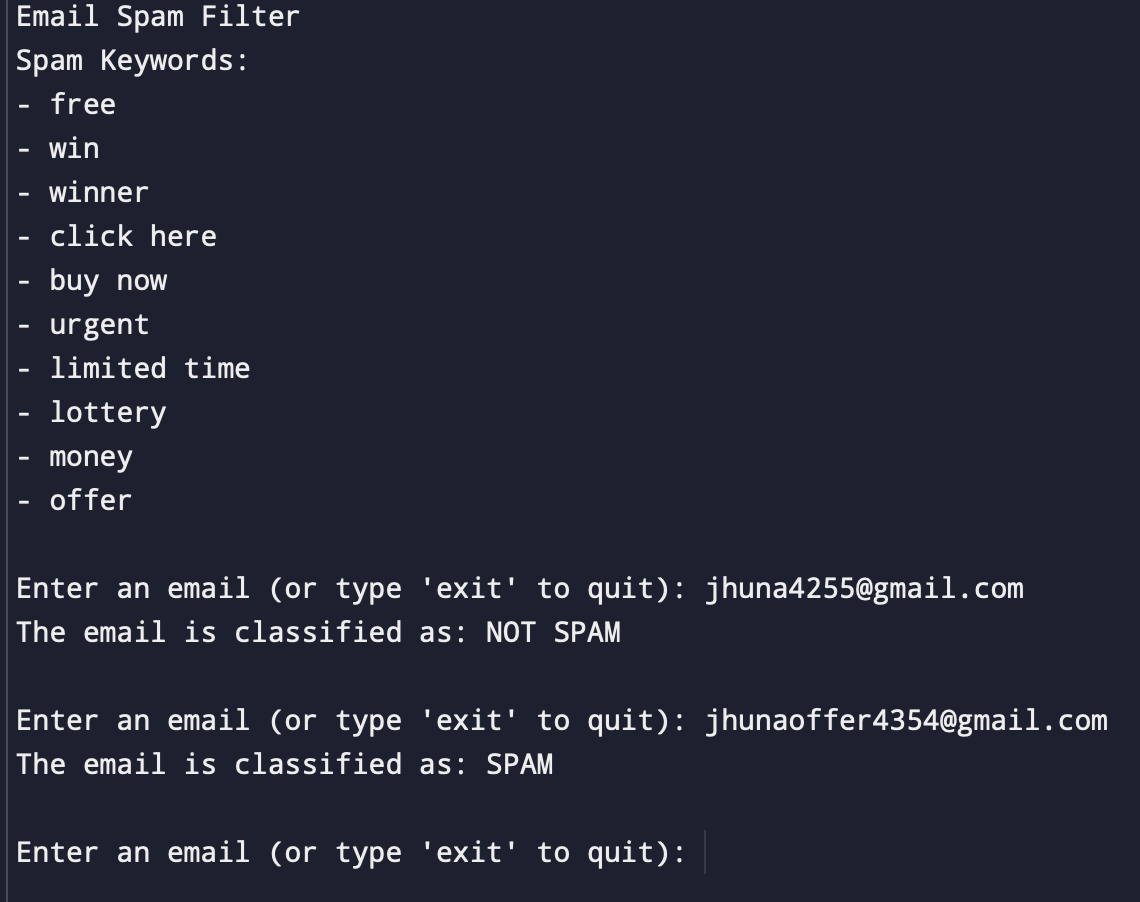
Challenges

While implementing a spam mail filtering system provides clear benefits, it also involves several technical and practical challenges that must be addressed to ensure effectiveness and reliability.

1. **Evolving Nature of Spam:**  
    Spammers constantly change their tactics to bypass filters by using obfuscated text, image-based spam, or random word insertions. This makes it difficult for static or rule-based filters to keep up without frequent updates.

2. **High False Positives/Negatives:**  
One of the biggest challenges is balancing the accuracy of the model. A high false positive rate (marking genuine emails as spam) can frustrate users, while a high false negative rate (failing to detect actual spam) reduces the system’s usefulness.







**Acknowledgment**

I would like to express my sincere gratitude to all those who supported and guided me throughout the development of this project.

First and foremost, I extend my heartfelt thanks to my project supervisor, [Supervisor’s Name], for their valuable insights, encouragement, and continuous support. Their expert guidance played a crucial role in shaping the direction and outcome of this work.

I am also thankful to [Institution/Department Name] for providing the resources and learning environment necessary for the successful completion of this project.

Special thanks to my friends and classmates for their helpful discussions, suggestions, and feedback during various stages of the project.

Finally, I would like to thank my family for their constant motivation, patience, and emotional support, which helped me stay focused and dedicated throughout this journey.



**Future Enhancement**

While the current spam mail filtering system provides a reliable and efficient solution for detecting unsolicited emails, there are several areas where future improvements can be made to enhance its performance, adaptability, and user experience:

1.Integration of Deep Learning Models:

Incorporating advanced deep learning techniques such as Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs) can improve the system’s ability to understand complex patterns and context in email content, leading to higher accuracy.



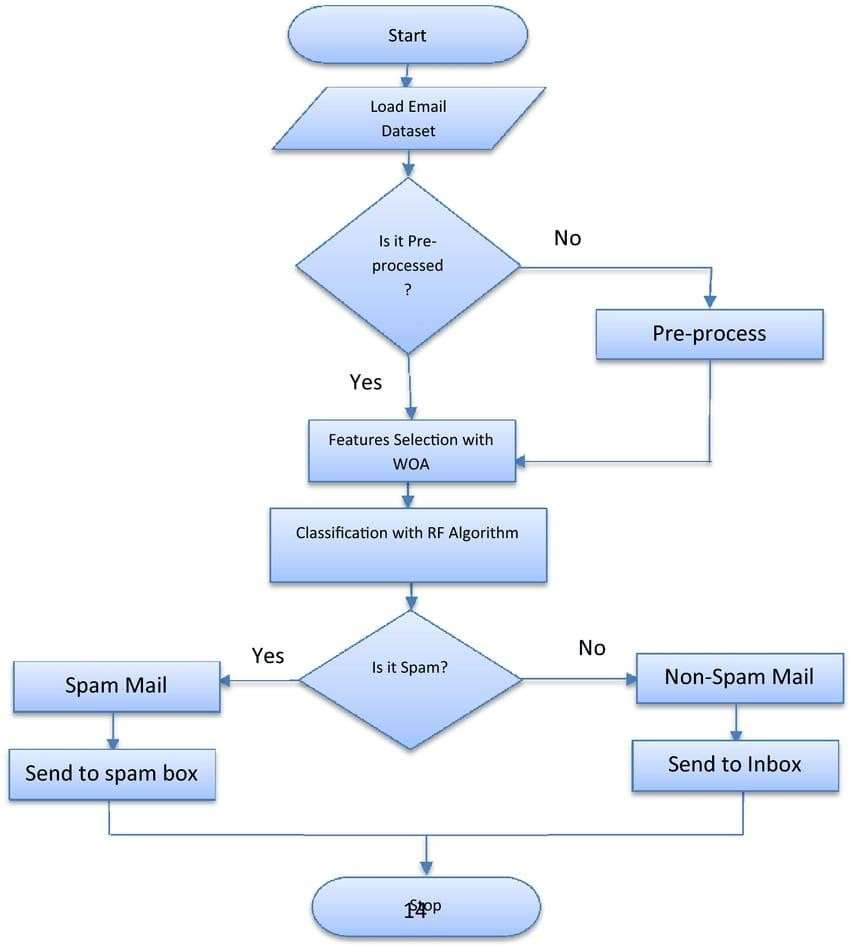
Problem Definition

With the rapid growth of email communication, spam emails have become a major concern for individuals, organizations, and service providers. Spam refers to unsolicited, irrelevant, or harmful emails often sent in bulk, which can lead to wasted time, reduced productivity, and serious security threats such as phishing, scams, and malware attacks.

The core problem is the need for a more intelligent, adaptive, and accurate spam filtering system that can effectively distinguish between spam and non-spam (ham) emails, while minimizing false classifications. The system must be capable of learning from patterns, analyzing email content, and adapting to new threats in real time.

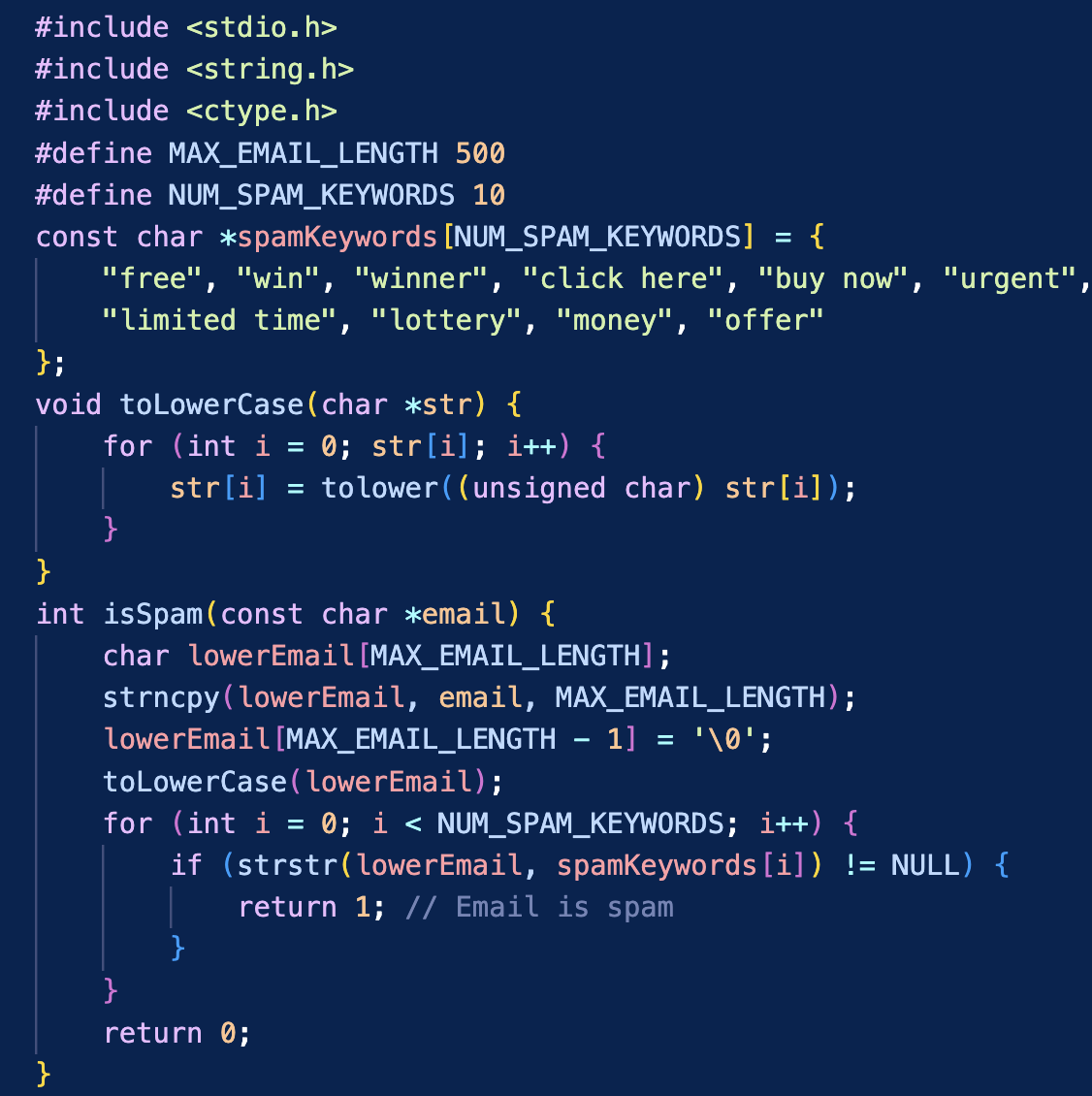


Flowchart

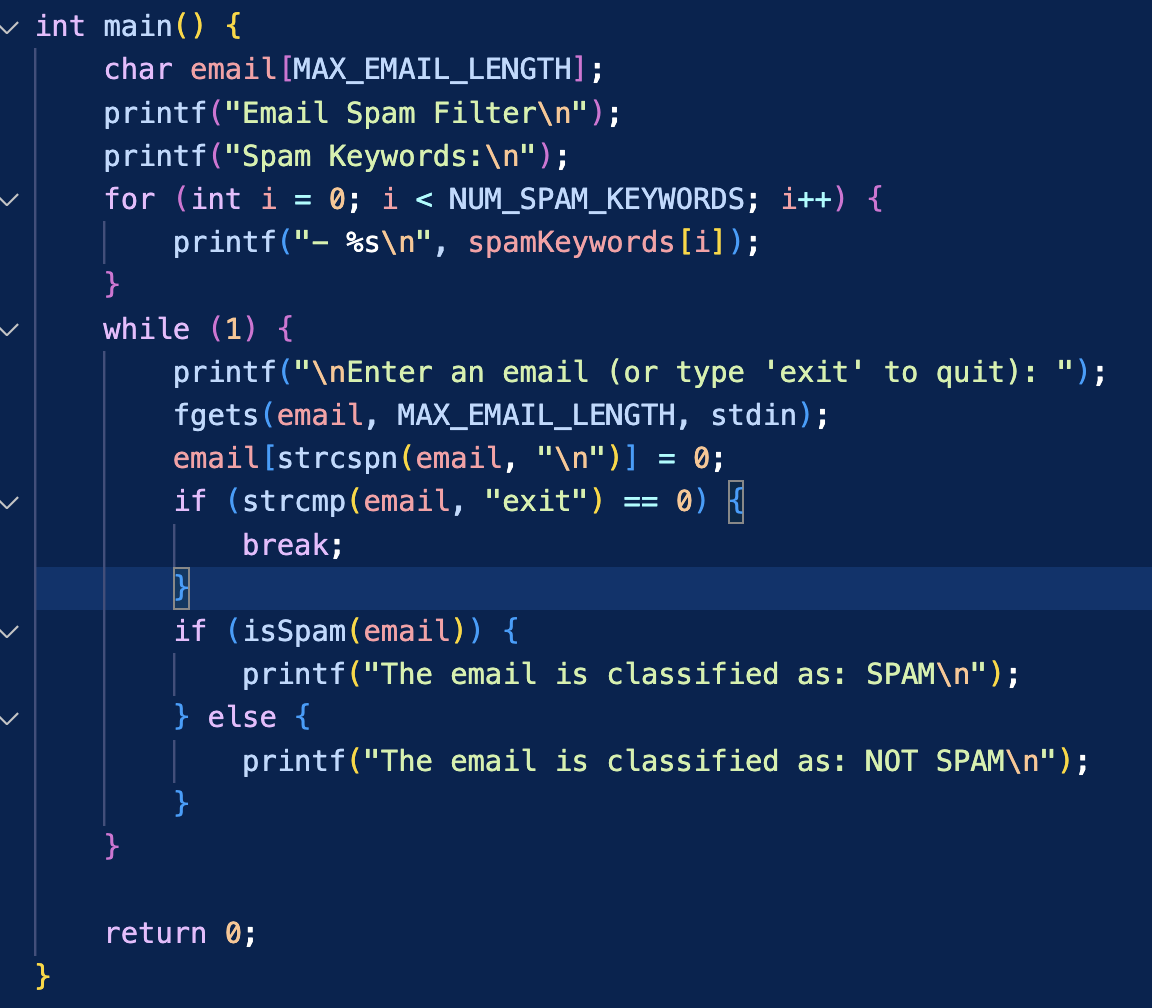




Code









output

