ISSN 2277-2685
IJESR/Jan-Mar. 2026/ Vol-16/Issue-1/619-626
Kyama Prasuna Sai *et. al.*, /International Journal of Engineering & Science Research
*Full Length Article*

# Smart Mail Shield: A Spam Detection System Using Python

**J Stella Mary[1], Kyama Prasuna Sai[2], Rayabandi Sirisha[3], Kaminelligaru Swathi[4]**
[1]Associate Professor; Department of Electronics and Communication Engineering Bhoj Reddy Engineering College for Women Hyderabad India
[2,3,4]B.Tech Students; Department of Electronics and Communication Engineering Bhoj Reddy Engineering College for Women Hyderabad India
Mail Id; kswathigoud123@gmail.com[4]
***Accepted 29-03-2026***


***Abstract***
*The widespread use of digital communication platforms has significantly increased the volume of unsolicited and malicious emails. Spam and phishing emails pose serious cybersecurity threats, including identity theft, financial fraud, and unauthorized access to sensitive information. Conventional email filtering techniques often depend on simple keyword-based rules, which are insufficient for identifying sophisticated and constantly evolving phishing strategies.This study presents Smart Mail Shield, an intelligent spam and phishing detection system developed using Python. The proposed solution combines multiple analytical approaches, including heuristic text evaluation, malicious URL identification, domain authenticity verification, and Optical Character Recognition (OCR)-based image analysis. Unlike traditional filtering systems that focus mainly on textual content, the proposed system can also process image-based inputs to detect phishing attempts hidden within embedded images.The system extracts relevant indicators such as suspicious keywords, harmful links, and irregular domain structures to compute a risk score for the analyzed content. A user-friendly web interface built using the Flask framework enables users to submit email text or upload images for automated security assessment. Based on the computed risk score, the system categorizes the input into three levels: Safe, Suspicious, or Phishing.*
***Keywords***
*Spam Detection, Phishing Detection, Email Security, Python, Optical Character Recognition (OCR), Flask Web Application, Heuristic Analysis, Cybersecurity*

## Introduction

In the contemporary digital environment, electronic mail has become one of the most widely used communication channels across multiple sectors, including education, corporate communication, banking services, and social networking. Its accessibility and efficiency make it an essential platform for exchanging information and conducting digital transactions. However, the rapid growth of email usage has also led to an increase in cyber threats, particularly spam and phishing attacks, which exploit vulnerabilities in communication systems and human behavior.Phishing represents a deceptive cyberattack in which malicious actors impersonate legitimate organizations to trick users into revealing confidential information. These fraudulent messages are often designed to resemble official communications from trusted institutions such as financial organizations, social networking platforms, or government services. By manipulating users through carefully crafted messages, attackers attempt to persuade victims to click malicious links, download infected attachments, or disclose sensitive credentials.Over time, phishing strategies have become increasingly sophisticated. Attackers now use techniques such as shortened or masked URLs, domain spoofing, and visual deception to evade traditional detection mechanisms. In some cases, phishing content is embedded within images rather than text, allowing attackers to bypass keyword-based filters. As a consequence, many users unknowingly interact with malicious content, resulting in financial losses, identity theft, and significant data breaches.Traditional spam detection mechanisms generally rely on predefined rules or machine learning models that analyze textual features such as keyword occurrence, frequency patterns, and message structure. Although these methods are effective in identifying simple spam messages, they often struggle to detect modern phishing attempts that involve complex patterns, dynamic links, or image-based content. These limitations demonstrate the need for a more comprehensive approach capable of analysing both textual and visual information within emails.To address these issues, this research proposes **Smart Mail Shield**, an intelligent spam and phishing detection framework developed using Python. The system integrates multiple analytical techniques, including heuristic content analysis, URL extraction, domain verification, and Optical Character Recognition (OCR). By combining these approaches, the system can examine both textual and image-based email content, improving its ability to identify hidden phishing indicators that may evade conventional filters.The primary objective of this

work is to develop a practical and efficient email security solution capable of identifying modern phishing strategies while maintaining low computational complexity. By combining heuristic analysis, link inspection, and OCR-based image processing, the proposed system aims to enhance the reliability of spam detection and contribute to improved cybersecurity practices.

**Literature Survey**

Research on spam and phishing detection has explored a variety of techniques, including machine learning, heuristic analysis, and URL inspection. Several studies have contributed to improving detection accuracy and developing advanced cybersecurity solutions.Jain and Gupta (2019) proposed a phishing detection method based on machine learning algorithms that analyse URL-related features. Their study highlighted how characteristics such as abnormal URL length, the presence of special characters, and misleading domain structures can be used to identify malicious websites. The results demonstrated that URL analysis plays a crucial role in preventing phishing attacks before users interact with harmful content.Kumar and Sharma (2023) introduced a lightweight phishing detection framework that relies on heuristic analysis rather than complex machine learning models. Their approach focuses on identifying suspicious keywords, verifying domain authenticity, and detecting urgent phrases commonly used in phishing emails. This method provides fast and explainable detection with minimal computational overhead.Abdelnabi, Fritz, and Holz (2020) developed VisualPhishNet, a framework that detects phishing websites by analysing visual similarities between fraudulent pages and legitimate websites. The study showed that attackers frequently imitate the appearance of trusted organizations to deceive users, highlighting the importance of visual-based detection techniques.Le, Nguyen, and Vu (2021) investigated phishing emails from the perspective of social engineering patterns. Their research identified common linguistic cues such as urgent warnings, account suspension messages, and prompts encouraging users to verify credentials or click embedded links.Khonji, Iraqi, and Jones (2013) conducted a comprehensive survey of phishing detection methods, examining techniques such as blacklist filtering, heuristic rules, machine learning, and website analysis. Their work emphasized that no single technique can fully address all phishing scenarios and recommended hybrid approaches that combine multiple detection strategies.Similarly, Verma and Dyer (2015) examined the structural and linguistic properties of phishing URLs and demonstrated that statistical analysis of link patterns can significantly improve detection accuracy. Their findings highlight the importance of analyzing URL characteristics as part of a comprehensive phishing detection system.

**Email Spam and Phishing Detection System**

Email continues to play a crucial role in digital communication, supporting activities ranging from personal messaging to financial transactions and academic collaboration. However, the increasing reliance on email platforms has also led to a rise in cyber threats that exploit this communication channel. Among these threats, spam and phishing emails represent some of the most common and damaging forms of cybercrime.Spam messages are unsolicited emails typically sent in large volumes, often containing advertisements, promotional content, or malicious links. Phishing emails, in contrast, are specifically designed to deceive users by impersonating trusted entities. These messages attempt to convince recipients to reveal sensitive information such as passwords, authentication codes, or banking details.To mitigate these threats, various email filtering systems have been developed to analyse message content and identify suspicious patterns. Techniques such as keyword-based filtering, URL inspection, heuristic analysis, and machine learning models are commonly used to improve detection accuracy.However, many existing systems focus primarily on text-based analysis and fail to address modern phishing techniques that incorporate visual deception or dynamic content. Consequently, there is a need for more advanced detection mechanisms that integrate multiple analytical methods.

**Existing Spam Detection System Model**

Traditional email spam detection systems generally follow a rule-based framework that analyses incoming messages through a sequence of processing stages. The system begins by receiving the email input, which may include subject lines, body text, and embedded links.The next stage involves preprocessing the message content to remove unnecessary symbols and convert the text into a standardized format suitable for analysis. After preprocessing, the system performs keyword filtering to detect suspicious words commonly associated with spam or phishing emails.In addition to keyword analysis, the system extracts URLs from the message and evaluates their structure to determine whether they may be malicious. Based on the results of these analyses, the system classifies the email as either spam or legitimate and displays the classification outcome to the user.

**Working of the Existing System**

The operation of traditional spam detection systems typically follows a sequential processing pipeline. When an email is received, the system first extracts important components such as the subject, body content, and hyperlinks. These elements are then processed to prepare them for analysis.During the preprocessing stage, unnecessary characters, HTML tags, and symbols are removed from the message,

and the text is converted to lowercase to ensure uniformity. This step helps improve the accuracy of keyword matching and reduces irrelevant noise within the dataset.The system then performs keyword filtering by comparing the message content with predefined lists of suspicious terms frequently used in spam emails. If a significant number of these keywords are detected, the email is flagged as potentially malicious.URL analysis is also conducted to evaluate links embedded within the email. Pattern-matching techniques are used to identify unusual characteristics such as excessively long URLs, suspicious domain names, or abnormal symbols. By combining the results of keyword filtering and URL analysis, the system determines whether the email should be classified as spam or legitimate.

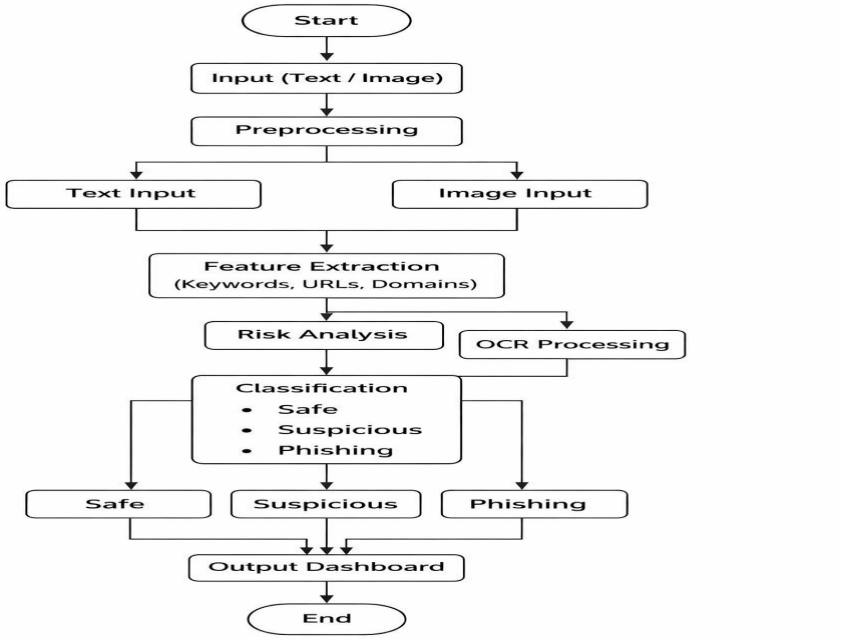
**Limitations of Existing Systems**

Although traditional spam filtering systems provide fast and efficient detection for basic spam messages, they suffer from several limitations. Most systems rely heavily on predefined rules and keyword lists, making them less effective against evolving phishing strategies. Attackers can easily bypass such filters by modifying keywords or embedding malicious content within images.Another limitation is the inability of many systems to analyse non-textual content. Image-based phishing attacks, which hide malicious instructions within graphics or screenshots, often evade detection by standard filters. Furthermore, rule-based systems lack adaptability and require manual updates to respond to emerging threats.These challenges highlight the need for more advanced detection approaches that combine multiple techniques to analyse both textual and visual information within email messages.

**Proposed Smart Mail Shield System Model**

The Smart Mail Shield system is designed as a multi-layer detection framework that processes user input through several analytical stages. The objective of this model is to address the weaknesses of traditional spam detection methods by combining different analysis techniques within a single platform.The system supports two forms of input: text input and image input. Text input may include email messages, URLs, or suspicious text content provided by the user. Image input includes screenshots or graphical messages that may contain phishing information embedded within them. When an image is uploaded, the system applies Optical Character Recognition (OCR) to extract the textual information present in the image.After receiving the input, the system performs preprocessing to clean and standardize the data. This step removes unnecessary characters and converts the content into a consistent format suitable for further analysis. Following preprocessing, the system performs feature extraction to identify relevant components such as suspicious keywords, embedded URLs, and domain names.The extracted features are then evaluated using heuristic-based rules that estimate the potential risk associated with the input. The classification results are stored in a database to maintain records and support future analysis through the dashboard interface. By combining text analysis, image processing, and risk evaluation techniques, the Smart Mail Shield system provides a comprehensive approach to phishing detection.

**Proposed System Block Diagram and Explanation**




**Block Diagram of Proposed System**

The Smart Mail Shield system follows a structured processing pipeline where input data flows through several stages before generating the final classification result. Each stage performs a specific task that contributes to the overall phishing detection process.The process begins with the input stage, where the system receives data from the user in the form of either text or image. Once the input is received, the system determines the type of input and forwards it to the appropriate processing module.The next stage is the preprocessing stage, where the data is cleaned and standardized. During this process, unnecessary characters, formatting elements, and noise are removed from the input. Text normalization is also performed to ensure consistency during further analysis.If the input is an image, the OCR processing stage is executed. In this stage, Optical Character Recognition extracts textual information from the image so that it can be analysed similarly to standard text input. This capability enables the detection of phishing messages embedded within images.After text extraction, the feature extraction stage identifies important elements from the processed data. These elements include suspicious keywords, URLs, and domain names that may indicate phishing activity.The extracted features are then evaluated during the risk analysis stage. In this step, heuristic rules are applied to identify patterns commonly associated with phishing attacks, such as urgency-based language, suspicious links, or domain inconsistencies.Based on the results of the risk analysis, the classification stage categorizes the input into Safe, Suspicious, or Phishing. Finally, the output stage displays the classification result to the user through the web interface or dashboard and stores the result for future reference.This structured workflow ensures accurate analysis and efficient detection of phishing attempts.

**Modules of Smart Mail Shield System**

The Smart Mail Shield system follows a modular architecture where each component performs a specific task within the phishing detection process. This modular design enhances system flexibility and allows individual modules to be updated or modified without affecting the overall functionality.The Input Handling Module is responsible for receiving user input and initiating the analysis process. The module supports both text and image formats and validates the input before forwarding it for further processing.The Preprocessing Module prepares the input data for analysis by removing unnecessary symbols, converting text into lowercase, and eliminating irrelevant formatting. This step ensures that the data is structured and suitable for feature extraction.The OCR Module enables the system to analyse image-based phishing attacks. When an image is uploaded, the module extracts textual information using Optical Character Recognition and converts it into machine-readable text. This extracted text is then analysed along with normal textual input.The Feature Extraction Module identifies important elements from the input data such as suspicious keywords, URLs, and message patterns. Pattern matching techniques, including regular expressions, are used to detect hyperlinks and structural patterns commonly found in phishing messages.The URL Analysis Module examines detected hyperlinks to identify suspicious characteristics such as excessive length, unusual characters, or the use of IP addresses instead of domain names. These characteristics often indicate malicious intent.The Domain Verification Module evaluates the authenticity of extracted domain names by comparing them with trusted domain lists. This helps detect domain spoofing attempts frequently used in phishing attacks.The Risk Analysis Module evaluates the identified features and calculates a risk score based on predefined heuristic rules. This module acts as the decision-making component of the system.The Classification Module categorizes the analysed input into Safe, Suspicious, or Phishing based on the calculated risk score. This multi-level classification provides more detailed information than traditional binary spam filters.Additional modules such as the Dashboard and History Module, Chatbot Module, and Admin Module improve system usability. The dashboard displays analysis results and historical records, the chatbot assists users with queries, and the admin module allows monitoring and management of system operations..

**Flask Framework**

Flask is a lightweight web framework used to build the web interface of the Smart Mail Shield system. It provides routing mechanisms and request-handling capabilities required for developing a web-based application.In the proposed system, Flask acts as the connection between the user interface and the backend processing modules. It accepts user input, processes the data using Python-based detection algorithms, and returns the classification result to the browser.Because Flask is lightweight and flexible, it is widely used for prototype and academic projects. It allows developers to build functional web applications quickly while maintaining full control over system components.

**HTML, CSS and Supporting Libraries**

The frontend interface of the Smart Mail Shield system is developed using HTML and CSS. HTML provides the structural layout of the web application, including text input areas, image upload options, action buttons, and result display sections.CSS is used to enhance the visual presentation of the application by defining layout styles, colors, spacing, and result indicators. These design elements improve the readability and usability of the system.Several supporting Python libraries are integrated into the system. Regular Expressions

(Regex) are used to identify and extract URLs from text content. Werkzeug ensures secure handling of uploaded files. The Python Imaging Library (PIL) and pytesseract enable OCR functionality for analysing image-based phishing content.

Together, these technologies support the efficient operation of the Smart Mail Shield system.

**Optical Character Recognition (OCR)**

Optical Character Recognition is a technology used to convert text contained in images into machine-readable digital format. The process involves analysing character shapes and patterns within an image and transforming them into editable text.OCR is particularly important for detecting modern phishing attacks because attackers often embed malicious instructions or fake login details inside images. Since traditional filters cannot read text within images, such attacks often bypass security systems.In the Smart Mail Shield system, OCR is used to extract text from uploaded screenshots or graphical messages. The extracted text is then analysed using the same techniques applied to regular text input, including keyword detection and URL analysis.The OCR process typically includes image preprocessing, character recognition, and text conversion. Integrating OCR significantly strengthens the system's ability to detect phishing attempts hidden within visual content.

**Results and Discussion**

This section provides a detailed evaluation of the results generated by the Smart Mail Shield system during testing. The system analyses email content by applying text analysis, URL inspection, domain verification, and OCR-based image analysis to classify messages into Safe, Suspicious, or Phishing categories.

presents the main interface of the Smart Mail Shield application. The interface allows users to submit suspicious email text or upload an image for analysis.The design of the interface is intentionally simple and user-friendly to ensure that users can easily analyse both textual and image-based inputs. This improves usability and makes the system suitable for practical applications.shows the analysis of a legitimate email message. In this case, the system classifies the message as **Safe**, with a calculated risk score of 0%.This result indicates that the system successfully recognizes genuine messages and avoids false alarms by confirming the absence of suspicious patterns or malicious indicators.demonstrates the AI explanation feature integrated into the system. This module provides additional information such as detected keywords, domain verification results, and confidence levels associated with the classification.Providing explanatory information enhances transparency and allows users to better understand how the classification decision is generated by the system.

illustrates the detection of a partially suspicious message. In this case, the system categorizes the input as Suspicious because moderate risk indicators were identified during analysis.This intermediate classification is important because it alerts users to potential risks without immediately labelling the message as phishing. It demonstrates the system's ability to distinguish between different threat levels.

presents the detection of a phishing email. The system identifies the message as **Phishing** due to the presence of high-risk indicators such as malicious URLs, urgent language patterns, and suspicious domain characteristics.This result confirms that the system effectively identifies strong phishing signals and accurately flags malicious content.shows the scan history module, where previously analysed inputs are stored together with their classification results.This functionality allows users to review past scans and maintain a record of analysed messages, which improves usability and monitoring.displays the system dashboard, which summarizes key statistics such as total scans performed, safe emails detected, suspicious messages identified, and phishing emails detected.

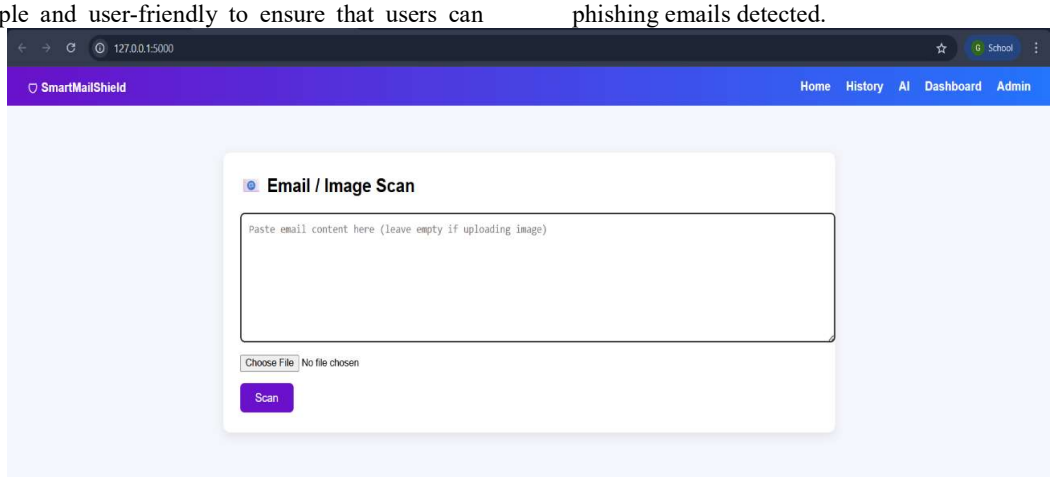



**Fig 1 Smart Mail Shield Email / Image Scan**

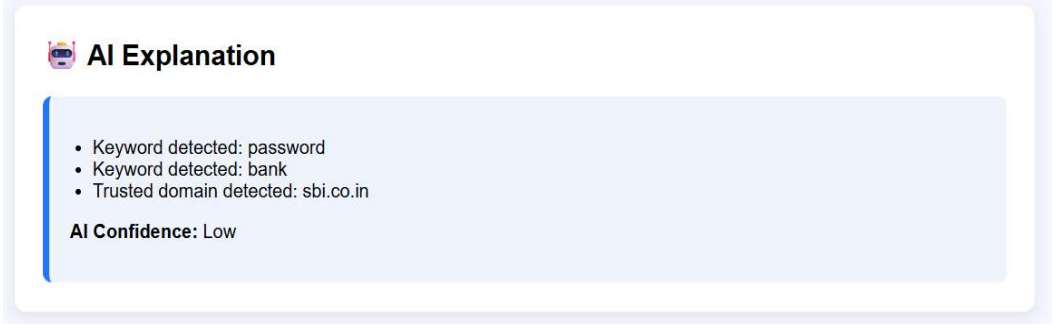



**Fig 3 Smart AI Detection**

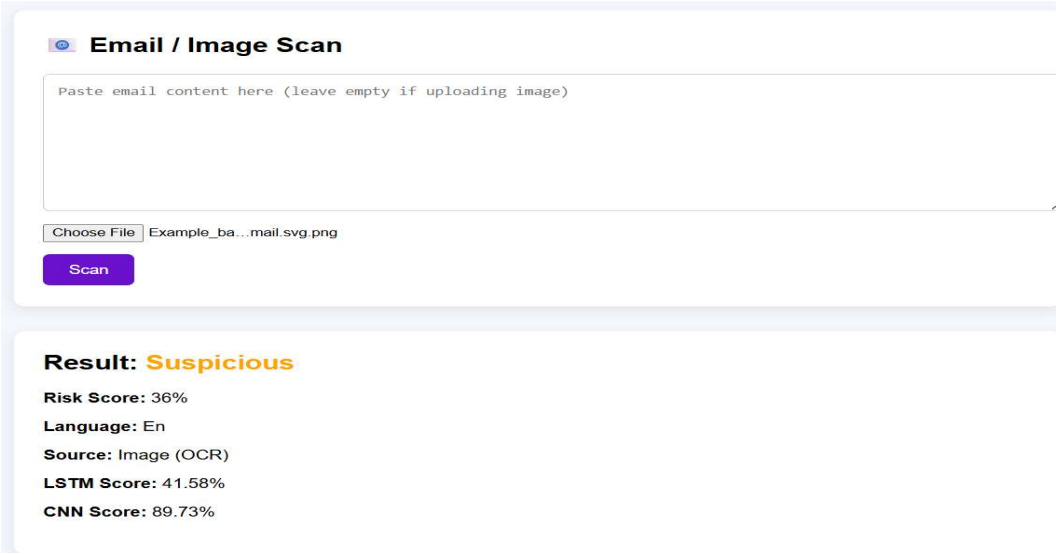



**Fig 4 Suspicious Email Detection**

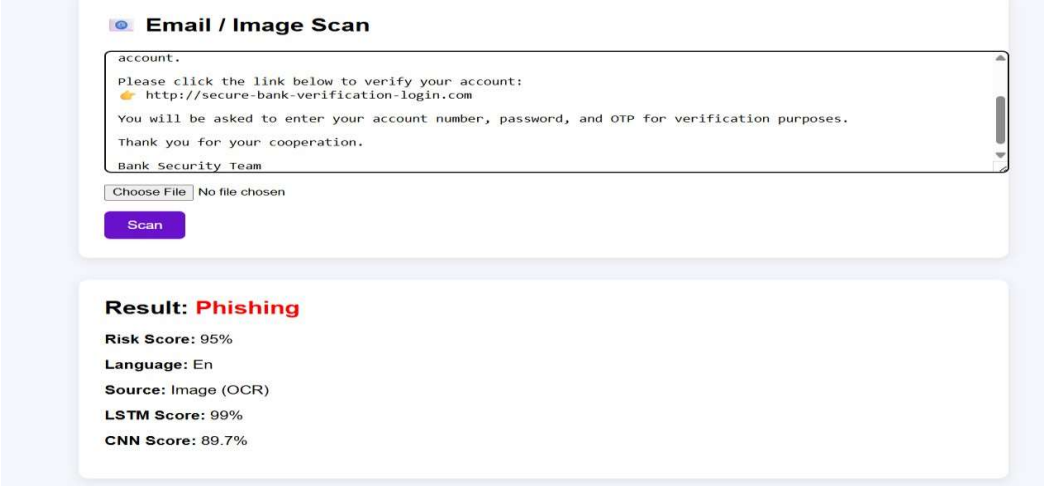



**Fig 75 Phishing Email Detection**

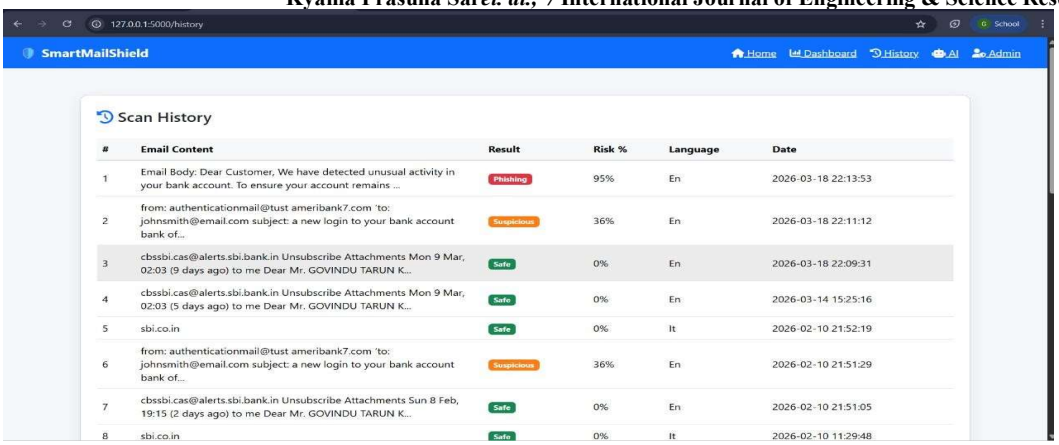



**Fig 6 Smart Mail Shield Scan History**


**Fig 7 Security Dashboard**

The dashboard provides a visual overview of system activity and helps users monitor detection performance more effectively.The experimental results indicate that the Smart Mail Shield system performs efficiently in identifying spam and phishing threats. By combining keyword analysis, URL verification, domain validation, and OCR-based image inspection, the system achieves improved detection capability.The inclusion of OCR technology significantly enhances the system's ability to identify phishing attacks embedded within images, which traditional text-based detection systems often fail to detect. Furthermore, the heuristic risk analysis approach ensures accurate classification, while the three-level categorization scheme improves the interpretability of the results.

## Conclusion

The Smart Mail Shield system presents an effective approach for detecting spam and phishing emails using a combination of modern detection techniques. With the growing dependence on digital communication, protecting users from phishing attacks, fraud, and information theft has become increasingly important.Traditional spam filtering mechanisms often rely on simple keyword matching techniques, which are insufficient for detecting advanced phishing strategies. The proposed system addresses this limitation by integrating multiple detection techniques, including heuristic text analysis, URL inspection, domain validation, and OCR-based image processing.The system categorizes analysed inputs into three risk levels: **Safe, Suspicious, and Phishing**, which provides users with a clearer understanding of potential threats. The incorporation of OCR technology allows the system to analyse image-based phishing messages, which represents a significant improvement over traditional text-only spam filters.Furthermore, the system includes additional functionalities such as an interactive web interface, scan history storage, and a visualization dashboard. These features enhance user interaction and provide useful insights into detection outcomes.The experimental evaluation demonstrates that the Smart Mail Shield system performs efficiently in identifying spam and phishing threats with satisfactory accuracy and real-time response. Overall, the proposed framework provides a reliable and scalable solution for improving email security in modern communication environments.

**Future Scope**

Although the Smart Mail Shield system provides an effective phishing detection solution, several improvements can be implemented in future work to enhance its performance and scalability.One potential enhancement is the integration of machine learning and deep learning models. These techniques can learn patterns from large datasets and detect complex phishing strategies that may not be captured by rule-based methods.Another improvement involves enhancing the OCR module and image analysis capabilities. Advanced image processing techniques could improve the extraction of text from low-quality or distorted images and enable detection of visually deceptive phishing pages.Future development may also incorporate real-time threat intelligence services to perform advanced URL and domain reputation analysis. This integration would allow the system to detect newly generated malicious domains more effectively.The system can also be expanded with a more advanced administrative dashboard that provides detailed analytics, trend analysis, and security reports. Enhanced historical tracking could help identify recurring patterns and suspicious activities.

Additional improvements may include support for multi-language phishing detection, allowing the system to analyse phishing messages written in different languages.The system could also be extended into a mobile application or browser extension, enabling users to analyse suspicious content directly while browsing or using messaging platforms.

**References**

1. [1] N. Jain and B. B. Gupta, “Machine Learning Techniques for Detecting Phishing Websites Using URL-Based Features,” *IEEE Access*, vol. 7, pp. 42993–43004, 2019.
2. [2] S. Kumar and A. Sharma, “A Lightweight Heuristic Method for Detecting Phishing Emails Using Content-Based Indicators,” *IEEE Access*, vol. 11, pp. 24567–24578, 2023.
3. [3] H. Abdelnabi, M. Fritz, and T. Holz, “VisualPhishNet: Detecting Zero-Day Phishing Websites Through Visual Similarity Analysis,” in *Proceedings of the IEEE Symposium on Security and Privacy*, 2020, pp. 1234–1248.
4. [4] Q. Le, T. Nguyen, and P. Vu, “Detection of Phishing Attacks Using Suspicious Content Indicators and Social Engineering Patterns,” *International Journal of Computer Applications*, vol. 174, no. 12, pp. 1–7, 2021.
5. [5] M. Khonji, Y. Iraqi, and A. Jones, “A Comprehensive Survey of Phishing Detection Techniques and Countermeasures,” *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 2091–2121, 2013.
6. [6] R. Verma and K. Dyer, “Understanding Phishing URL Characteristics Using Statistical Learning Classifiers,” in *Proceedings of the ACM Conference on Data and Application Security and Privacy (CODASPY)*, 2015, pp. 111–122.