Phishing Emails

By Lachlan James

101362125

Design Concept.

Our project on phishing email detection is driven by the goal of establishing a robust and effective system for identifying and mitigating email-based security threats. In pursuit of this objective, we recognize the critical importance of meticulously planning and executing the setup and development process. Our approach prioritizes numerous key aspects to ensure the success and sustainability of our implementation.

1. Comprehensive Repository Management  
   We established a centralized repository using industry-standard version control systems – in this case GitHub. This repository serves as the backbone of our collaborative development efforts, providing a unified platform for team members to share, review and contribute to the project codebase. By adhering to the best practices in repository management, including branch management strategies and pull request workflows, we foster an environment of transparency, accountability, and version control integrity.
2. Adherence to Coding Standards and Guidelines  
   Consistent adherence to coding standards and guidelines is fundamental to the maintainability, readability, and scalability of our codebase. We embrace established coding conventions to ensure uniformity and clarity in our code. Additionally, we prioritise the use of meaningful variable names, modular code organization, and proper documentation practices to enhance code comprehension and facilitate collaborative development efforts.
3. Thorough Commenting Standards

Clear and informative code documentation is crucial for promoting readability and facilitating knowledge sharing among team members. We adopt a rigorous approach to commenting standards, ensuring that each function, class, and module is accompanied by descriptive comments outlining its purpose, inputs, outputs, and relevant implementation details. Consistent commenting practices empower developers to understand, modify and extend the codebase confidently and clearly.

1. Seamless Environment Setup

Creating a dependable and consistent development environment is essential for efficient setup and uniformity across diverse environments. We utilize containerization to encapsulate project dependencies, simplifying the creation of isolated development environments. Automated provisioning tools further reduce manual tasks, ensuring consistency and predictability for all team members.

1. Scalable Prototype Architecture and Components

The prototype architecture encompasses a multi-layered approach to phishing email detection, comprising data reprocessing, feature extraction, classification, and response mechanisms. Components such as natural language processing techniques, machine learning modules, and email filtering algorithms are integrated to analyse email content, headers, and attachments for indicators of phishing behaviour.

Methodology.

The methodology adopted for the detection of phishing emails revolves around the application of Knowledge Discovery (KD) and data mining techniques. The primary Objective is to develop an efficient email classifier capable of accurately categorizing incoming emails as either spam or legitimate. The process commences with the implementation of KD steps, which involves extracting pertinent features from a training dataset comprising email samples. These extracted features serve as input for a variety of data mining algorithms, aimed at identifying the most effective classifier. One notable model, as proposed by Yasin and Abuhasan (2016), employs linguistic processing techniques and ontologies to enhance the semantic similarity among emails. This enhancement contributes to the overall performance and efficiency of the classification process, leading to improved accuracy in distinguishing between spam and legitimate emails. Furthermore, the selection of papers for evaluation was meticulously conducted, focusing on research works that demonstrate significant impact and intelligent automation in phishing email detection. These selected papers were thoroughly analysed to assess the utilization of machine learning principles, the robustness and effectiveness of the proposed solutions and the necessary modifications required to address any identified drawbacks.

Design Constraints

* Data availability  
  The availability and quality of labelled phishing and legitimate email datasets can significantly impact the training and testing phases of the detection program. Constraints related to data collection, labelling and preprocessing must be addressed to ensure the programs accuracy and generalization.
* Scalability   
  The ability of the detection program to scale with increasing email volume and diversity (Multiple languages) is essential for deployment in large-scale email systems. Designing scalable algorithms and infrastructure to manage growing data volumes and user traffic without compromising performance is a significant constraint.
* Regulatory compliance  
  Compliance with data protection regulations, privacy laws, and industry standards may impose constrains on data handling, storage, and processing within the detection program. Ensuring compliance with relevant regulations and standards is essential for legal and ethical use of the program.

Specifications

* Email collection and preprocessing  
  Implement email collection mechanisms to retrieve messages in real-time. Preprocesses emails to extract relevant metadata and content for analysis.
* Feature extraction and selection

Implement feature extraction techniques, such as keyword frequency, header analysis, and link properties. Select relevant features to optimize detection accuracy and minimize computational overhead.

* Machine learning models  
  Implement supervised learning algorithms, such as logistic regression, decision trees, or neural networks, to train classification models.
* Real time detection  
  Implement streaming algorithms or event driven architectures to process incoming emails in real-time. Utilize parallel processing and distributed and computing to manage high email volumes efficiently.
* Alerting and Reporting  
  Implement alerting mechanisms, such as email notifications or dashboard alerts to notify stakeholders of suspicious emails. Generate detailed reports on phishing activity, including statistics, trends, and remediation actions.
* Integration and Compatibility

Design APIs or integration points to connect the phishing email detection system with email servers, firewalls, Security information and event management (SIEMs) and other security solutions. Support standard email protocols and interoperability with major email clients and platforms.

* Adaptability and Updates

Implement mechanisms for continuous monitoring and updating of detection algorithms and threat intelligence feeds. Incorporate feedback loops to learn from detected phishing attempts and improve future detection capabilities.

Vulnerability Analysis

* Vulnerability one  
  The system may incorrectly classify legitimate emails as phishing or fail to detect actual phishing emails.  
  Address this by Implement engineering techniques to enhance the accuracy of feature extraction, continuously update and refine machine learning models using feedback loops.
* Vulnerability two

Sophisticated attackers may employ evasion techniques to bypass detection.

Utilize advanced content analysis algorithms such as natural language processing and image recognition. Regularly update threat intelligence feeds.

Justification of design

The design specifications for the phishing detection system address crucial aspects for success and alignment with objectives. Implementing real-time email collect and preprocessing enables prompt identification of phishing threats, supporting proactive security measures. Feature extraction techniques optimize detection accuracy while minimizing computational overhead. Integration of supervised learning algorithms facilitates high-accuracy identification of phishing emails, leveraging advanced technologies for enhanced security. Real time detection ensures prompt mitigation of attacks, while alerting mechanisms and detailed reporting facilitate informed decision-making, aligning with the objective of providing actionable insights into threats. Designing for integration and compatibility maximizes usability and effectiveness. Overall, these specifications contribute to a robust system, aligning with objectives to mitigate email-based security threats.

References

Karim, A., Azam, S., Shanmugam, B., Kannoorpatti, K., & Alazab, M. (2019). A comprehensive survey for intelligent spam email detection. *IEEE Access*, *7*, 168261-168295. <https://doi.org/10.1109/access.2019.2954791>

Salloum, S., Gaber, T., Vadera, S., & Shaalan, K. (2022). A systematic literature review on phishing email detection using natural language processing techniques. *IEEE Access*, *10*, 65703-65727. <https://doi.org/10.1109/access.2022.3183083>

Yasin, A., & Abuhasan, A. (2016). An intelligent classification model for phishing email detection. *International Journal of Network Security & Its Applications*, *8*(4), 55-72. <https://doi.org/10.5121/ijnsa.2016.8405>