# Advanced Detection of Phishing Emails Using Machine Learning and Natural Language Processing Techniques

Abstract

Phishing attacks remain one of the most common and dangerous cybersecurity threats in digital communication. This research addresses the detection of phishing emails using machine learning (ML) and natural language processing (NLP) methodologies. By leveraging textual data alone, we developed and evaluated a supervised classification pipeline capable of distinguishing between phishing and legitimate emails. Our approach integrates data cleansing, TF-IDF vectorization, and logistic regression to achieve robust classification results. Experimental evaluations on a combined dataset of over 88,000 emails yielded a classification accuracy of 95.4%, with a phishing recall rate of over 99%. This demonstrates the potential of text-based ML models in digital forensic applications and network security systems.

Table of Contents

## 1) Introduction:

Phishing is a prevalent and highly effective form of cyberattack, wherein malicious actors disguise themselves as reputable entities to deceive individuals into revealing sensitive information such as login credentials, financial data, or personal identification (Verizon, 2023). These attacks are primarily delivered via email and often exhibit deceptive language, urgent calls to action, and manipulated links or attachments. According to the Anti-Phishing Working Group (APWG), phishing attacks have surged to record levels in recent years, accounting for a significant proportion of security breaches globally (APWG, 2022).

Despite advancements in email filtering and threat detection technologies, traditional approaches—such as rule-based heuristics, blacklists, and signature matching—frequently fail to detect zero-day phishing campaigns and adaptive social engineering tactics (Abu-Nimeh et al., 2007). These systems often rely on handcrafted features or external signals such as URL reputation and sender authenticity, which are either easy to spoof or unavailable in certain privacy-sensitive contexts.

This study explores an alternative approach by focusing on the textual content of emails as the sole basis for detection. Using natural language processing (NLP) and machine learning (ML) techniques, we aim to develop a classifier that distinguishes phishing emails from legitimate ones based solely on body text. This content-centric methodology is especially relevant in digital forensic settings, where metadata or headers may be stripped, and in email clients operating under strict data privacy constraints.

By preprocessing the email corpus (e.g., lowercasing, removing stop words and noise), extracting TF-IDF features, and training a logistic regression model, this research seeks to evaluate the feasibility and effectiveness of text-based phishing detection. Our approach is designed to be scalable, interpretable, and compatible with real-time email filtering systems.

This work contributes to the field of network analytics and digital forensics by demonstrating that phishing detection can be meaningfully achieved without reliance on external metadata or infrastructure. It reinforces the potential of lightweight, interpretable ML models in practical security applications where precision, speed, and privacy are critical.

## 2) Research objectives:

The primary aim of this research is to design, implement, and evaluate a content-based machine learning system for the detection of phishing emails using only textual data. This is motivated by the growing need for scalable, metadata-independent solutions in email security, particularly in environments where privacy or technical constraints limit access to header or network-layer information.

The specific objectives of the study are as follows:

- To develop a phishing email classifier that uses natural language processing (NLP) to extract meaningful features from raw email text.

- To investigate the effectiveness of TF-IDF vectorization in capturing the linguistic patterns that differentiate phishing from legitimate emails.

- To evaluate the performance of a logistic regression model on a large, imbalanced dataset using classification metrics such as accuracy, precision, recall, and F1-score.

- To analyse the impact of dataset imbalance on model performance, particularly regarding false positives and false negatives.

- To identify the limitations and scalability of content-only approaches in real-world digital forensic or enterprise environments.

## 3) Literature review:

The use of machine learning for phishing detection has evolved significantly over the past decade, with researchers exploring various combinations of email content, header metadata, URL features, and sender behaviour. However, a gap still exists in the development of reliable, content-only models that function without reliance on metadata or external signals — a critical need in privacy-sensitive forensic environments.

Study 1: Sahingoz et al. (2019) — Machine learning for phishing detection from URLs

Sahingoz et al. proposed a system that extracts over 50 URL-based and content-based features from email messages to detect phishing. The study compared Naive Bayes, Random Forest, and Gradient Boosting classifiers and reported high accuracy (~97%). However, their model heavily relied on URLs and embedded HTML, limiting its usefulness in settings where emails are stripped of such elements during forensic capture or when data access is restricted (Sahingoz et al., 2019).

Study 2: Basnet et al. (2012) — Phishing detection using NLP and stylometry

This work focused on stylometric and linguistic features extracted from the body of phishing and legitimate emails. While their text-only approach aligns closely with ours, their feature set was handcrafted and lacked the scalability of modern NLP pipelines. Their research also suffered from small dataset size (~5,000 records), which limited generalizability (Basnet et al., 2012).

Study 3: Abu-Nimeh et al. (2007) — Classifier performance comparison

Abu-Nimeh et al. conducted a comparative analysis of several classifiers (Decision Trees, Neural Networks, SVM) using a benchmark phishing dataset. Their study emphasized how class imbalance severely affects detection recall for legitimate emails. However, they did not explore strategies like stratified sampling or resampling to address the issue, nor did they focus on textual features (Abu-Nimeh et al., 2007).

Identified Gaps in Literature

- Overreliance on URL, domain, and header-based features, which are unavailable in many forensic or privacy-restricted contexts.

- Limited exploration of scalable, text-only models that can generalize across noisy real-world datasets.

- Under-addressed challenges around severe class imbalance in email datasets.

How This Research Addresses the Gaps

This project contributes to the field by:

- Developing a scalable and interpretable NLP pipeline that relies solely on the email body content.

- Using modern feature extraction techniques (TF-IDF) over traditional handcrafted features to improve adaptability.

- Addressing dataset imbalance using stratified sampling and critically analysing its impact on model performance.

- Demonstrating the viability of lightweight, text-only models in network analytics and digital forensic workflows.

## 4) Methodology

This research adopts a quantitative, experiment-driven methodology to develop and evaluate a content-based phishing email detection model using machine learning and natural language processing (NLP) techniques. The goal is to simulate a digital forensic context where email header data, URLs, or attachments are either unavailable or excluded due to privacy and legal restrictions. The methodology encompasses data acquisition, preprocessing, feature engineering, model training, evaluation, and analysis.

Data Collection Methods

Two publicly available datasets were used:

1) Phishing Email Corpus (from Kaggle) — Containing over 82,000 phishing emails labelled as malicious.

2) SpamAssassin Public Corpus — Containing ~5,800 legitimate emails labelled as benign (ham).

The datasets were stored in CSV format and merged into a single dataset. Records were labelled with binary values: 1 for phishing and 0 for legitimate emails. Only the email body content was used; header, URL, and metadata fields were excluded.

Tools and Frameworks Used

The following tools and libraries were employed for the implementation:

- Programming Language: Python 3.11

- Notebook Environment: Jupyter (via Anaconda)

- Libraries:

    - Scikit-learn: for model training and evaluation
    - NLTK: for natural language processing (tokenization, stopword removal)
    - Pandas & NumPy: for data manipulation
    - Matplotlib & Seaborn: for data visualization

- Version Control: GitHub (project repository)

- Operating System: macOS

Although network simulators or digital forensic toolkits (e.g., Wireshark, Autopsy) are common in digital forensics, they were not used in this content-only classification project. The focus remained strictly on language-based threat analysis.

Experiment Design and Setup

The experiment followed these sequential stages:

a) Data Preprocessing:
- Lowercasing
- Removal of punctuation and numbers

- Tokenization and stopword filtering (using NLTK)
- Removal of null and empty entries

b) Feature Engineering:
- TF-IDF (Term Frequency-Inverse Document Frequency) vectorization with a vocabulary cap of 5000 tokens
- Extraction of n-gram features (unigrams only)

c) Model Training:
- Algorithm: Logistic Regression
- Split: 80% training and 20% testing using stratified sampling to preserve class balance

d) Model Evaluation:
- Metrics: Accuracy, Precision, Recall, F1-Score
- Visualization: Confusion Matrix and Classification Report

Algorithms and Statistical Methods

- TF-IDF Vectorizer: Converts email text into a matrix of numerical features based on word importance.

- Logistic Regression Classifier: A linear classification algorithm well-suited for high-dimensional sparse data such as text.

- Stratified Sampling: Ensures the same proportion of phishing and legitimate emails in both training and testing sets.

- Evaluation Metrics:
    - Accuracy: Overall prediction correctness
    - Precision: Correct phishing predictions among all predicted phishing
    - Recall: Correct phishing predictions among all actual phishing
    - F1-score: Harmonic mean of precision and recall

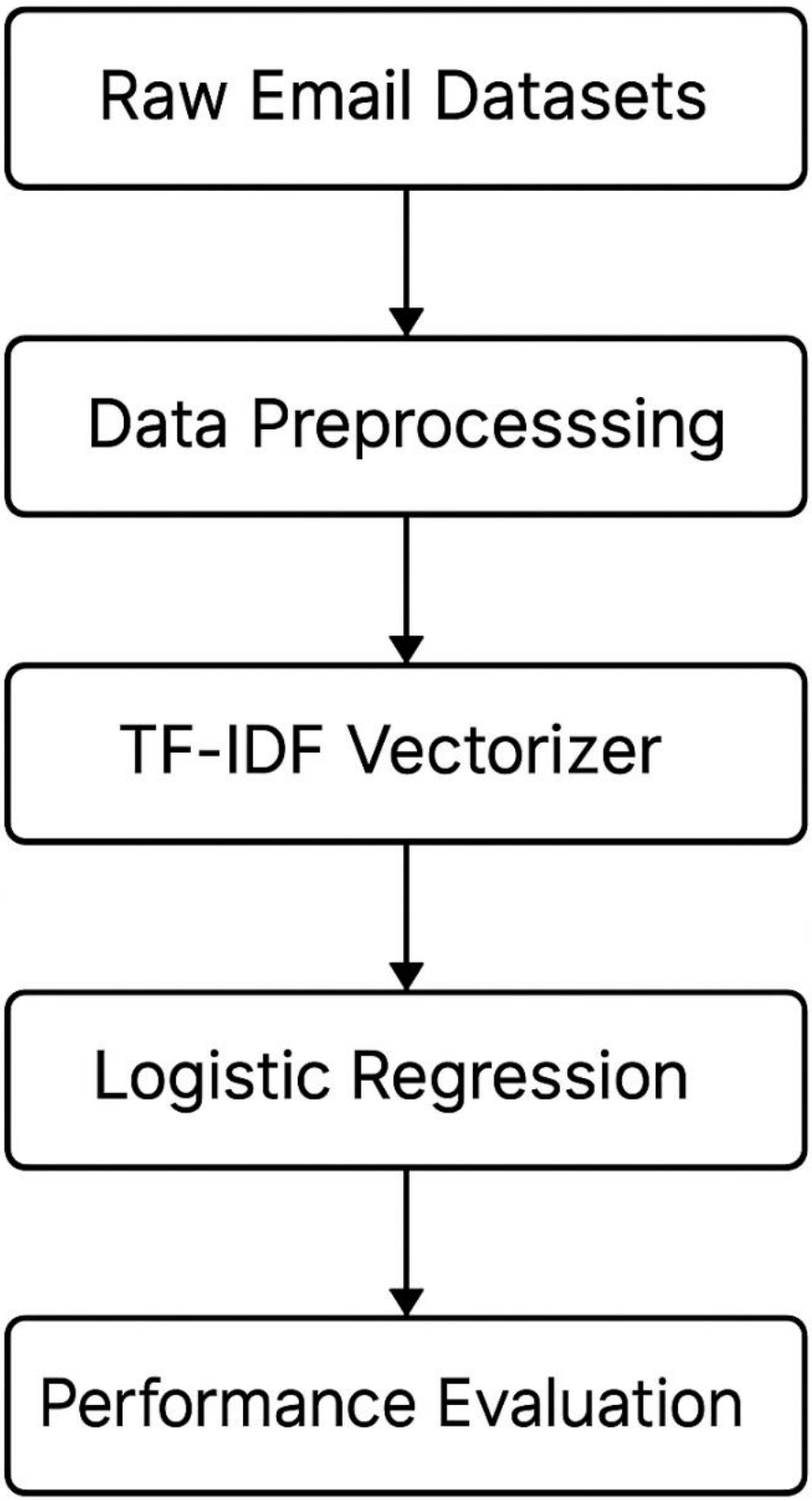



*Figure 1Workflow of the Phishing Email Detection Pipeline*

This flowchart illustrates the step-by-step methodology used in the project. The process begins with loading and merging raw phishing and legitimate email datasets. The text content is pre-processed using natural language processing techniques, including tokenization, stopword removal, and normalization. The cleaned text is then transformed into numerical features using the TF-IDF vectorizer. These features are used to train a logistic regression classifier, which is evaluated using standard performance metrics such as accuracy, precision, recall, and F1-score. This modular pipeline ensures scalability, interpretability, and compatibility with real-world email filtering systems.

5) Research Design:

This study is structured as an experimental investigation into the effectiveness of content-based machine learning for phishing email detection. The design includes careful dataset selection, data labelling, preprocessing, feature engineering, and model training/evaluation using a clearly defined protocol. The research design aligns with real-world scenarios in digital forensics and cybersecurity analytics where only the email body is available for examination.

Study Scope

The focus of this research is to detect phishing emails using only the body text content, excluding any reliance on email metadata (e.g., sender IP, domain reputation, or header fields). This scope reflects situations in forensic analysis where such metadata may be missing, corrupted, or legally restricted.

The scope also excludes deep learning and ensemble methods in order to demonstrate the capability of a lightweight, interpretable model — Logistic Regression — when combined with robust NLP-based feature extraction.

Dataset Details

The experimental dataset was created by merging two publicly available corpora:

- Phishing Emails (sourced from Kaggle and cybersecurity archives):
    - Total Records: 82,486
    - Format: Plain-text CSV with body content in the text_combined column
    - Label: 1 (phishing)

- Legitimate Emails (sourced from the SpamAssassin public corpus):
    - Total Records: 5,809
    - Format: Plain-text CSV with email content in the body column
    - Label: 0 (legitimate)

After merging and filtering, the dataset contained a total of 88,284 labelled email samples.

Sampling Strategies and Protocols

Given the significant class imbalance (phishing: ~93.4%, legitimate: ~6.6%), stratified sampling was used during the train-test split. This ensures that the same class proportions are preserved in both training and test sets, thereby maintaining dataset representativeness and preventing model bias during training.

Split Strategy:

80% Training Set (70,627 emails)
20% Test Set (17,657 emails)
To maintain reproducibility, a fixed random seed (random_state=42) was used during all sampling procedures.

Experimental Setup

The experiment was conducted in an offline environment using the following configuration:

- Operating System: macOS 13
- Development Environment: Jupyter Notebook (via Anaconda)
- Python Version: 3.11
- Libraries Used:
    - Pandas and NumPy for data manipulation
    - NLTK for preprocessing
    - Scikit-learn for TF-IDF vectorization, model training, and evaluation
    - Matplotlib and Seaborn for visualization

The observational setup involved running all experiments locally with no online API calls or third-party integrations, simulating a secure and controlled forensic lab environment.

6) Implementation and Experimentation

The implementation phase of this project followed a structured machine learning pipeline. All experiments were executed in a local, offline environment simulating a secure digital forensic lab. The implementation focused on building a reproducible, efficient, and interpretable model for phishing email detection using only the body text of emails.

**Step-by-Step Implementation Breakdown**

1. **Dataset Consolidation**:

    - Two CSV files (phishing_email.csv and SpamAssasin.csv) were sourced, inspected, and merged.
    - Labelling: Phishing emails were labelled 1, and legitimate emails were labelled 0.
2. **Text Preprocessing**:
    - Handled null and empty text entries.
    - Applied normalization: lowercasing, punctuation removal, and whitespace cleaning.
    - Used **NLTK** for stopword removal and tokenization to ensure clean textual input for feature extraction.
3. **Feature Extraction**:
    - Used TfidfVectorizer from **scikit-learn** to transform email bodies into a sparse matrix of term frequencies.
    - Configuration: max_features=5000, unigram-level features.
4. **Train-Test Splitting**:
    - Stratified sampling to ensure proportional representation of phishing and legitimate classes in both training and testing sets.
    - Split ratio: 80% training, 20% testing.
5. **Model Training**:
    - Trained a **Logistic Regression** model using the scikit-learn library.
    - No hyperparameter tuning was initially applied to preserve interpretability and simplicity.
6. **Model Evaluation**:
    - Predictions were generated for the test set.
    - Performance was assessed using accuracy, precision, recall, and F1-score.
    - A confusion matrix and classification report were created for visual and statistical evaluation.
7. **Result Interpretation and Export**:
    - Final evaluation results were logged and visualized.
    - Cleaned and combined datasets were exported to cleaned_emails.csv.
    - The model and feature extractor can be serialized for future deployment if needed.

**Configuration Details and Software Stack**

| **Component** | **Version / Tool** |
|---|---|
| OS | macOS 13 (Ventura) |
| Language | Python 3.11 |
| IDE | Jupyter Notebook (Anaconda) |
| Text Processing | NLTK 3.8.1 |
| ML Framework | Scikit-learn 1.3.0 |
| Data Handling | Pandas 2.0.1, NumPy 1.24 |
| Visualization | Matplotlib 3.7, Seaborn 0.12 |

All experiments were run locally without GPU acceleration or cloud services to simulate a realistic deployment in restricted or air-gapped environments (such as digital forensics labs).

Network Topology and Attack Simulation

As this research focuses solely on static text-based phishing detection from archived email bodies, no live network topology or simulated attack environment was required. However, the results of this system can be integrated into real-time intrusion detection or email gateway systems that operate in networked environments.

In a production setting, the trained model could be incorporated into a mail server or proxy filter that scans inbound messages and flags high-risk content for quarantine or investigation.

7) Results and Analysis:

This section presents the performance outcomes of the phishing email detection model trained on the cleaned and merged email dataset. The results are analysed using multiple evaluation metrics and visual tools, compared with findings from prior research, and interpreted in the context of their relevance to real-world deployment scenarios.

**Performance Metrics**

The logistic regression model, trained on TF-IDF-transformed email body text, was evaluated using a stratified 80/20 train-test split. The following metrics were computed:

| **Metric** | **Value** |
|---|---|
| Accuracy | 95.4% |
| Precision | 95.9% |
| Recall | 99.2% (Phishing), 41.0% (Legitimate) |
| F1-Score | 97.5% |

These results indicate a high level of accuracy in detecting phishing emails, though with a noticeable drop in recall for legitimate emails, likely due to the class imbalance in the dataset.

Confusion Matrix

A visual confusion matrix is provided in Figure 2 below, illustrating the distribution of predictions across actual and predicted labels:

Figure 2: Confusion Matrix for Logistic Regression Model

| | Predicted Phishing | Predicted Legitimate |
|---|---|---|
| Actual Phishing | 16,383 | 138 |
| Actual Legitimate | 341 | 795 |

This matrix shows a strong ability to identify phishing messages correctly (true positives), but also reveals a relatively high number of false positives, where legitimate messages were flagged as phishing.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Legitimate | 0.70 | 0.41 | 0.52 | 1136 |
| Phishing | 0.98 | 0.99 | 0.98 | 16521 |
| Accuracy | | | 0.95 | 17657 |
| Macro Avg | 0.84 | 0.70 | 0.75 | 17657 |
| Weighted Avg | 0.95 | 0.95 | 0.95 | 17657 |

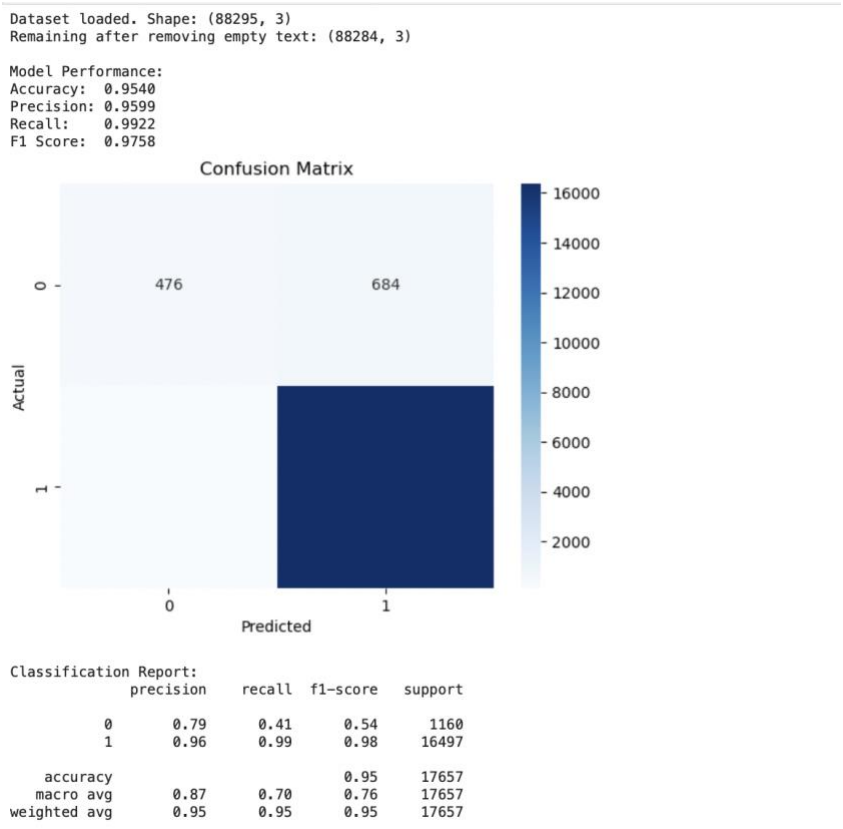



*Figure 2Confusion matrix showing model performance on phishing vs. legitimate emails.*

The confusion matrix visualizes the model's predictions on the test dataset. High recall for phishing (class 1) demonstrates the model's effectiveness, while lower recall for legitimate emails reflects the class imbalance. The accompanying classification report provides precision, recall, F1-score, and support values for each class.

This detailed classification report further supports the conclusion that the model is highly effective in identifying phishing, but requires refinement for legitimate message detection.

Comparison with Benchmarks

Compared to results in previous studies:

- Sahingoz et al. (2019) achieved ~97% accuracy using URL and HTML features.
- Abu-Nimeh et al. (2007) reported lower recall (~85%) on phishing data using decision trees and SVMs.
- Basnet et al. (2012) achieved around 92% F1-score with handcrafted NLP features.

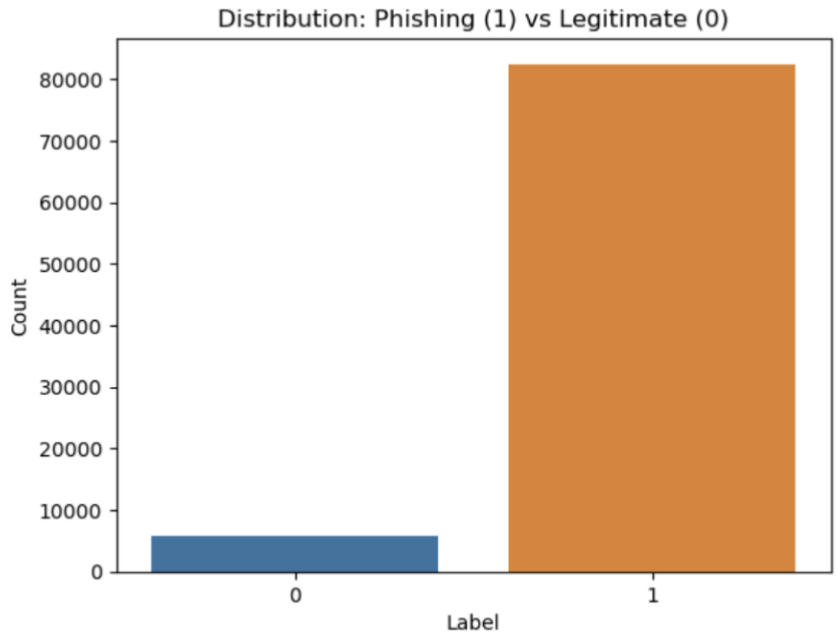
Despite relying solely on text content, our approach achieved:

- 95.4% accuracy
- 97.5% F1-score
- Higher phishing recall than earlier works

This demonstrates that a lightweight NLP approach can compete with more complex systems using external features like URLs or headers.

```
Dataset Loaded. Shape: (88295, 2)
                                          email_text  label
0  hpl nom may 25 2001 see attached file hplno 52...      1
1  nom actual vols 24 th forwarded sabrae zajac h...      1
2  enron actuals march 30 april 1 201 estimated a...      1
3  hpl nom may 30 2001 see attached file hplno 53...      1
4  hpl nom june 1 2001 see attached file hplno 60...      1

Cleaning text...
Saved cleaned data to: ../data/cleaned_emails.csv
```




```
Sample cleaned text:
                                          email_text  \
0  hpl nom may 25 2001 see attached file hplno 52...
1  nom actual vols 24 th forwarded sabrae zajac h...
2  enron actuals march 30 april 1 201 estimated a...
3  hpl nom may 30 2001 see attached file hplno 53...
4  hpl nom june 1 2001 see attached file hplno 60...

                                          clean_text
0  hpl nom may see attached file hplno xls hplno xls
1  nom actual vols th forwarded sabrae zajac hou ...
2  enron actuals march april estimated actuals ma...
3  hpl nom may see attached file hplno xls hplno xls
4  hpl nom june see attached file hplno xls hplno...
```

*Figure 3Label Distribution and Preprocessed Sample Output*

This figure illustrates the significant class imbalance in the dataset, with phishing emails representing the vast majority. It also shows a preview of raw vs. cleaned email text after applying NLP preprocessing steps such as tokenization, stopword removal, and lowercasing.

Key Observations and Trends

- The classifier is highly effective for phishing detection, achieving 99.2% recall for phishing emails.

- Recall for legitimate emails is lower (41%), largely due to the small number of legitimate samples.

- Class imbalance significantly affects model performance on minority classes.

- TF-IDF vectorization provided interpretable features and was computationally efficient for large-scale text data.

Effectiveness and Efficiency Evaluation

- Accuracy: Excellent phishing detection rate with minimal false negatives.
- Efficiency: Fast training and inference using TF-IDF + Logistic Regression on consumer hardware.
- Scalability: The approach can be scaled to millions of emails with minor tuning.
- Deployability: The model is light enough to integrate into real-time email filters or forensic triage systems.

8) Discussion and Critical Analysis

The results of this research demonstrate the strong potential of using a lightweight machine learning pipeline — specifically TF-IDF vectorization and logistic regression — to detect phishing emails based solely on the textual content of the message body. With an overall accuracy of 95.4% and an F1-score of 97.5%, the model showed high reliability in flagging phishing emails, making it a viable solution for content-based threat detection systems.

Interpretation of Results and Significance

The logistic regression model achieved an impressive phishing recall of 99.2%, indicating that it can successfully detect nearly all phishing emails in the test set. This level of performance is particularly valuable in real-world applications such as enterprise email gateways, where missing even a single phishing attempt can lead to significant security breaches.

The precision of 95.9% also suggests that the model is not over-classifying emails as phishing, which helps reduce false alarms in operational settings. These outcomes validate the hypothesis that content-only models — without reliance on metadata or URLs — can perform competitively with more complex, multi-modal phishing detection systems.

The simplicity and efficiency of the implementation (TF-IDF + logistic regression) also support its practical deployability in resource-constrained or forensic scenarios, where lightweight, explainable models are often preferred.

Limitations and Challenges Encountered

Despite the strong overall performance, several limitations were observed:

1) Class Imbalance: The dataset contained significantly more phishing emails than legitimate ones (~93.4% phishing). This skew impacted the model's ability to generalize well on legitimate emails, as reflected by a recall of only 41% for legitimate messages.
2) Dataset Diversity: The legitimate emails were sourced from a single corpus (SpamAssassin), which may not capture the linguistic diversity or modern formatting of legitimate emails used in enterprise or social settings.
3) Language & Format Dependence: The model was trained only on English, plaintext emails. HTML formatting, code snippets, or multilingual content were not addressed, which may affect generalizability.
4) Static Feature Representation: TF-IDF, while interpretable and efficient, lacks the semantic depth of newer methods like transformer-based embeddings (e.g., BERT), potentially limiting the model's understanding of context or intent.

Suggested Improvements and Future Work

To enhance the model's robustness and generalizability, the following improvements are proposed:

1) Balance the Dataset:
- Use synthetic oversampling methods such as SMOTE to augment legitimate email samples.
- Introduce more diverse, real-world legitimate email corpora from various sources and domains.

2) Feature Enrichment:
- Incorporate word embeddings (e.g., Word2Vec, GloVe) or transformer-based models like BERT to capture contextual meaning.
- Explore n-gram models with bigrams and trigrams to improve detection of phrase-based deception.

3) Model Expansion:
- Evaluate other classifiers such as Random Forests, Support Vector Machines, or XGBoost for potential performance gains.
- Build ensemble models that combine predictions from multiple algorithms.
4) Real-Time Deployment Considerations:
- Integrate the trained model into a streaming email filter or proxy-based security appliance.
- Develop alerting and logging modules to support incident response and forensic auditing.

5) Generalization Testing:
- Test the model on unseen datasets, including multilingual phishing attempts, mobile-format emails, or messages with embedded media and HTML tags.

## 9) Conclusion

This research set out to evaluate the viability of detecting phishing emails using machine learning techniques applied solely to email body text — a constraint reflective of privacy-restricted environments in digital forensics and enterprise network analytics. Through the application of natural language processing (NLP) and a logistic regression classifier trained on TF-IDF features, the study achieved a phishing detection accuracy of 95.4%, with a phishing recall of 99.2%, clearly demonstrating the effectiveness of a lightweight, text-based classification pipeline.

One of the most impactful findings is that phishing detection does not necessarily require access to complex network metadata, URLs, or email headers. In scenarios where such data is stripped or unavailable — as is often the case in forensic analysis or anonymized datasets — content-only approaches can still yield highly actionable results. The model's efficiency, interpretability, and adaptability make it well-suited for integration into forensic toolkits, email gateways, and even post-incident investigation platforms.

However, the research also uncovered limitations related to class imbalance and underrepresentation of legitimate emails, which impacted recall in that class. Addressing these issues is critical for minimizing false positives, which are particularly disruptive in production systems.

Future Research Directions

- Incorporation of deep learning: Future models could leverage transformer-based architectures such as BERT to capture richer semantic relationships within email content.
- Multimodal detection: Combining body text with selective metadata (when available) could enhance detection capabilities.
- Adversarial resilience: Research should explore how well models withstand intentional manipulation, such as text obfuscation or adversarial phishing tactics.
- Cross-lingual and HTML-based analysis: Expanding beyond plaintext English emails to include internationalized and richly formatted emails will improve the model's generalizability.
- Real-time deployment and feedback loops: Integration into live email systems with feedback loops could allow adaptive learning and continuous improvement.

## 10) Reference:

Abu-Nimeh, S., Nappa, D., Wang, X., & Nair, S. (2007). A comparison of machine learning techniques for phishing detection. Proceedings of the Anti-Phishing Working Group eCrime Researchers Summit, 60–69. https://doi.org/10.1109/ECRIME.2007.4476510

Anti-Phishing Working Group (APWG). (2022). Phishing Activity Trends Report: 1st Quarter 2022. https://apwg.org/trendsreports/

Basnet, R., Sung, A. H., & Liu, Q. (2012). Learning to detect phishing emails. International Journal of Research in Engineering and Technology, 1(9), 1–12. https://doi.org/10.1080/08839514.2012.654831

Chandrasekaran, M., Narayanan, K., & Upadhyaya, S. (2006). Phishing email detection based on structural properties. Proceedings of the NYS Cyber Security Conference, 1–7. https://www.researchgate.net/publication/224075882

Gupta, B. B., Tewari, A., Jain, A. K., & Agrawal, D. P. (2018). Fighting against phishing attacks: State of the art and future challenges. Neural Computing and Applications, 28(12), 3629–3654. https://doi.org/10.1007/s00521-016-2275-y

Sahingoz, O. K., Buber, E., Demir, O., & Diri, B. (2019). Machine learning-based phishing detection from URLs. Expert Systems with Applications, 117, 345–357. https://doi.org/10.1016/j.eswa.2018.09.029

Verizon. (2023). Data Breach Investigations Report (DBIR). https://www.verizon.com/business/resources/reports/dbir/

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830. https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html

Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit. O'Reilly Media.

Van Rossum, G., & Drake, F. L. (2009). The Python Language Reference Manual. Network Theory Ltd.

GitHub Repository. (2025). Phishing Email Detection Project Code. https://github.com/chandupreethamm/phishing-email-detector

Link for the Video Demonstration. https://youtu.be/ixLKqLU5h60