Email and URL Phishing Detection using Machine Learning

# 1. Introduction & Problem Statement

Phishing attacks remain a prevalent cybersecurity threat, targeting users through deceptive emails and malicious URLs. These attacks aim to steal sensitive information such as login credentials, credit card numbers, and personal data. Despite increasing awareness, users still fall victim due to the sophisticated nature of phishing tactics.

This project proposes a machine learning-based solution to automatically detect phishing emails and URLs. By analyzing patterns and characteristics in the data, our model aims to accurately classify whether a given input is malicious or safe, providing a preventive layer in cybersecurity systems.

# 2. Dataset Description

We used publicly available datasets for both phishing emails and URLs:

- Email Phishing Dataset: Contains labeled emails with features such as subject line, content, presence of suspicious words, and metadata (e.g., links, sender info).  
- URL Dataset: Includes over 10,000 URLs labeled as phishing or legitimate. Features extracted include:  
 - URL length  
 - Presence of IP address  
 - Use of HTTPS  
 - Number of dots/subdomains  
 - Suspicious characters (@, //, -, etc.)

Data was cleaned and converted into numerical formats suitable for ML models.

# 3. Preprocessing Techniques

To prepare the data for model training, the following preprocessing steps were performed:

- Data Cleaning: Removed null values and irrelevant fields.  
- Feature Engineering:  
 - For URLs: Extracted length, presence of special characters, domain-related features.  
 - For emails: Tokenized content, removed stopwords, counted suspicious terms.  
- Label Encoding: Converted labels into binary form (1 = phishing, 0 = safe).  
- Scaling: Applied normalization to ensure feature values are within a standard range.  
- Train-Test Split: Typically 80-20 split for training and evaluation.

# 4. Model/Algorithm Used

We experimented with multiple machine learning classifiers. The best-performing ones include:

- Random Forest Classifier: High accuracy and robustness against overfitting.  
- Logistic Regression: Simple and effective for binary classification.  
- XGBoost Classifier (for URL dataset): Provided the best precision-recall balance.

Each model was trained using Scikit-learn, with hyperparameter tuning through GridSearchCV for improved performance.

# 5. Evaluation Metrics

To evaluate the effectiveness of the models, we used:

- Accuracy: Overall correctness of the model.  
- Precision: How many predicted phishing instances were actually phishing.  
- Recall: Ability to detect all phishing instances.  
- F1-Score: Harmonic mean of precision and recall.  
- Confusion Matrix: To visualize true positives, false positives, etc.

Sample Result (Random Forest on URL data):  
- Accuracy: 96.7%  
- Precision: 95.2%  
- Recall: 94.8%  
- F1 Score: 95.0%

# 6. Results

- The Random Forest and XGBoost models achieved over 95% accuracy on phishing detection.  
- Feature importance plots showed that features like "URL length," "presence of IP address," and "use of HTTPS" were crucial.  
- The system was able to detect phishing attempts with minimal false positives.

We built an interactive prototype (in progress) where users can input a URL/email and instantly receive a phishing prediction.

# 7. Future Scope & Improvements

- Deep Learning: Integrate LSTM or BERT-based models for analyzing email content contextually.  
- Real-time Detection: Deploy as a browser plugin or email gateway filter.  
- Continuous Learning: Automatically retrain the model on newly flagged phishing data.  
- User Interface: Improve the demo UI with clear feedback and alerts.  
- Ensemble Models: Combine predictions from multiple models for increased robustness.