1. Domain Background

Phishing is one of the most common attack methods used in cybersecurity. Attackers create fake websites that appear legitimate and trick users into entering sensitive information, such as login credentials or payment details. These phishing sites are typically distributed through deceptive URLs that mimic trusted domains. According to the Anti-Phishing Working Group (APWG), phishing reports reached an all-time high in 2023, affecting users and organisations across every industry [1].

Traditional rule-based systems to detect phishing are often rigid and easy to bypass. Hackers can craft URLs that closely resemble trusted websites to avoid detection. In contrast, machine learning models can analyse patterns in URLs and learn to distinguish between legitimate and malicious ones. This is particularly useful for organisations such as banks, which need to prevent users from interacting with phishing sites in real time.

Machine learning has proven effective in phishing detection due to its ability to identify subtle patterns in URL structure that may not be obvious to users or static filters. Studies such as Sahoo et al. (2017) [2] show how models trained on lexical and structural features of URLs can generalise to new phishing attacks.

This project aims to build a machine learning model that can classify URLs as phishing or legitimate, and deploy it as a real-time prediction service using AWS.

References:
[1] APWG. (2023). Phishing Activity Trends Report. https://apwg.org
[2] Sahoo, D., Liu, C., & Hoi, S. C. (2017). Malicious URL Detection using Machine Learning: A Survey. ACM Computing Surveys, 50(3), 1–33.

2. Problem Statement

The goal is to create a system that takes a raw URL as input and predicts whether it is likely to be a phishing attempt. This will be implemented as a binary classification problem, where the output is either “phishing” or “legitimate”.

The model will be deployed to a SageMaker endpoint, and a Lambda function will be used to process live requests and call the endpoint. This setup simulates how a real-time phishing detection system might work inside a larger security pipeline.

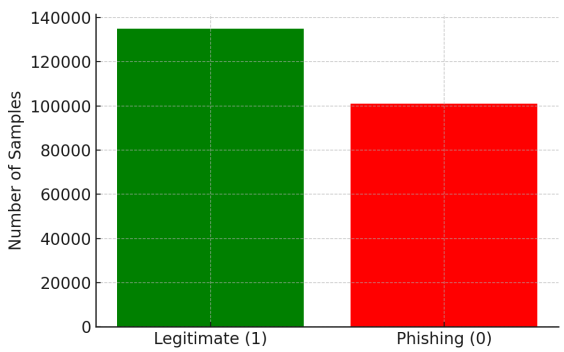
3. Datasets and Inputs

The dataset used for training comes from the UCI Machine Learning Repository: the PhiUSIIL Phishing URL dataset. It contains a total of 235,795 labeled examples, with features derived from the structure and composition of each URL.

The target variable is the label which represents if the URL is legitimate or Phishing and binary:

- 134,850 legitimate URLs (label = 1)
- 100,945 phishing URLs (label = 0)

This distribution shows the dataset is reasonably balanced, with a slight majority of legitimate examples. As such, no



special resampling or rebalancing techniques are expected to be necessary.

For this project, I will focus on a subset of 6 to 8 features that are easy to extract from raw URLs. These include:

- URL length
- Number of dots
- Use of HTTPS
- Presence of IP address
- Presence of “@” symbol
- Number of subdirectories
- Number of digits
- Suspicious keywords in the URL (e.g., “login”, “secure”, “account”)

These features will be extracted during inference to simulate real-world usage. The label is binary: 0 for phishing and 1 for legitimate.

4. Solution Statement

I will build a binary classification model trained on structural URL features. The baseline model will be a logistic regression model. The final model will be either a Random Forest or XGBoost classifier, which are known to perform well on tabular data.

The model will be trained using SageMaker and deployed as a real-time endpoint. A Lambda function will take in a raw URL, extract features, and send them to the SageMaker model for prediction. This allows the system to make live predictions from simple inputs, without external data or complex preprocessing.

5. Benchmark Model

The benchmark model will be a logistic regression classifier. It is simple, interpretable, and often used as a baseline for binary classification tasks.

This model will provide a baseline level of accuracy, precision, recall, and F1 score. Its performance will be compared to more advanced models like Random Forest and XGBoost to show the value of using more flexible learning algorithms.

6. Evaluation Metrics

Because this is a binary classification task, I will evaluate models using:

- F1 score (primary metric)
- Precision
- Recall
- Accuracy (for general context)

F1 score is the most important metric, as it balances false positives and false negatives, which are both critical in a phishing detection scenario.

All models will be tested on the same dataset split, and results will be compared against the baseline to measure improvement.

7. Project Design

Development plan:

1. Load and clean the dataset. Select 6 to 8 strong, usable features.
2. Train a logistic regression model and record baseline performance.
3. Train a final model using Random Forest or XGBoost.
4. Save the final model and prepare a SageMaker-compatible train.py script.
5. Write a custom inference.py script that includes feature extraction.
6. Deploy the model to a SageMaker endpoint.

7. Create a Lambda function that:
- Accepts a raw URL as input
- Extracts features using the same logic as training
- Sends features to the SageMaker endpoint
- Returns the model's prediction

Tools:
- Python
- Pandas / scikit-learn / XGBoost
- AWS SageMaker
- AWS Lambda

Dataset
- UCI PhiUSIIL Dataset: https://archive.ics.uci.edu/dataset/967/phiusiil+phishing+url+dataset