# Rapport Scientifique : Détection Proactive de Phishing par Machine Learning

## Auteur : Youssef abqari

## Date : Décembre 2025

## Module : Data Science & Machine Learning

## Thématique : Cybersécurité & Détection d'Intrusion

## # https://colab.research.google.com/drive/1tV7bbaGgnS8BmDuOkdmFINBIdr_

## 1. Introduction

### Contexte

La cybercriminalité, et plus particulièrement le phishing (hameçonnage), représente une menace économique et sécuritaire majeure. Les méthodes de défense traditionnelles reposent souvent sur des "listes noires" (blacklists), une approche réactive qui ne permet de bloquer un site qu'après qu'il a été signalé ou a fait des victimes

### Problématique

Comment anticiper la menace en détectant un site de phishing uniquement à partir des caractéristiques techniques de son URL, avant même l'analyse de son contenu ? Il s'agit d'un problème de **classification binaire** supervisée où nous cherchons à distinguer deux classes : * **0 :** URL Légitime * **1 :** URL de Phishing

### Objectifs

L'objectif de ce projet est de construire un modèle prédictif capable de classifier une URL avec une précision (Accuracy) satisfaisante, tout en minimisant les faux négatifs (sites dangereux non détectés). Nous suivrons le cycle de vie complet d'un projet de Machine Learning : du nettoyage des données à l'optimisation des hyperparamètres.

---

## 2. Méthodologie

### 2.1 Le Jeu de Données (Dataset)

Nous avons utilisé le *Phishing Domain Detection Dataset* (Source : Kaggle/MichelleVP). * **Description :** Le jeu de données contient des caractéristiques lexicales (longueur, caractères spéciaux) et techniques extraites des URLs. * **Volume :** Environ 80 000 instances. * **Target :** Variable binaire équilibrée (50% Phishing / 50% Légitime), ce qui facilite l'apprentissage.

### 2.2 Pré-traitement (Preprocessing)

Avant la modélisation, les données ont subi un traitement rigoureux pour garantir la qualité des prédictions :

1. **Nettoyage des doublons :** Suppression des lignes identiques pour éviter le biais de sur-apprentissage.
2. **Gestion de la Target :** Encodage de la variable cible (si textuelle) en format numérique (0/1).
3. **Imputation :** Remplacement des valeurs manquantes par la médiane, une statistique plus robuste aux valeurs aberrantes que la moyenne.
4. **Standardisation (Scaling) :** Application du `StandardScaler` ($\mu = 0, \sigma = 1$). Cette étape est justifiée par l'hétérogénéité des échelles de nos variables (ex: *longueur de l'URL* vs *présence binaire d'une IP*), essentielle pour des algorithmes comme la Régression Logistique.

### 2.3 Choix des Algorithmes

Nous avons comparé trois familles d'algorithmes: * **Régression Logistique :** Modèle linéaire simple, utilisé comme *baseline*. * **Random Forest :** Méthode d'ensemble (Bagging) robuste aux données bruitées et capable de capturer des relations non-linéaires. * **Gradient Boosting :** Méthode de boosting séquentielle, souvent plus précise mais plus coûteuse en calcul.

---

## 3. Résultats & Discussion

### 3.1 Analyse Exploratoire (EDA)

L'analyse des corrélations a permis d'identifier les variables les plus discriminantes:

!

*Figure 1 : Matrice de corrélation des 10 variables les plus influentes.*

**Interprétation :** Nous observons que les variables techniques comme `domain_in_ip` (utilisation d'une IP au lieu d'un domaine) et lexicales comme `length_url` (longueur totale) sont fortement corrélées avec la classe Phishing.

Les pirates cherchent souvent à masquer la destination réelle par des URLs très longues ou des IP brutes.

### 3.2 Performance Comparée des Modèles

Les modèles ont été évalués sur un jeu de test (20% du dataset) après un entraînement sur 80% des données.

| Modèle | Accuracy | ROC-AUC |
|---|---|---|
| **Régression Logistique** | ~0.9318% | 0.9275 |
| **Random Forest** | **~0.9688%** | **0.9667** |
| **Gradient Boosting** | ~0.9532% | 0.9487 |

Nous constatons que les modèles non-linéaires (**Random Forest** et **Gradient Boosting**) surperforment largement la Régression Logistique. Le **Random Forest** a été retenu pour l'optimisation finale en raison de son excellent compromis entre performance et stabilité.

### 3.3 Analyse des Erreurs (Matrice de Confusion)

L'analyse de la matrice de confusion du modèle optimisé montre la répartition suivante :

*Figure 2 : Matrice de confusion du modèle Random Forest optimisé.*

**Discussion :** * Le taux de **Faux Négatifs** (Sites de phishing classés comme sains) est faible. C'est un point crucial en cybersécurité : il est préférable de bloquer par erreur un site légitime (Faux Positif) que de laisser passer une attaque (Faux Négatif).

---

## 4. Conclusion

### Synthèse

Ce projet a permis de démontrer qu'il est possible de détecter le phishing avec une grande précision (supérieure à 90%) en analysant uniquement la structure de l'URL, sans avoir besoin de télécharger le contenu de la page web. [cite_start]Le modèle Random Forest s'est avéré être le plus performant après optimisation des hyperparamètres via GridSearchCV[cite: 44].

### Limites

- Le modèle ne prend pas en compte le contenu sémantique de la page (texte, logos).
- L'utilisation de raccourcisseurs d'URL (bit.ly) masque les caractéristiques lexicales sur lesquelles repose notre modèle.

**Pistes d'amélioration**

Pour aller plus loin, nous pourrions : 1. Intégrer une analyse **NLP (Natural Language Processing)** sur la chaîne de caractères de l'URL. 2. Ajouter des features externes comme l'âge du nom de domaine (WHOIS data), les domaines récents étant plus suspects.