École Nationale Supérieure d'Informatique et d'Analyse des Systèmes - Rabat

Détéction des adresses URL malveillantes avec Machine Learning

Réalisé par :

Hiba ALAOUI

Aya AZIZ

Professeur:

M. Younes TABII

Table des matières

1 Introduction **2**
1.1 Problématique 2
1.2 Objectif du projet 2
1.3 Intérêt du Machine Learning 2

2 Exploration et Préparation des Données **3**
2.1 Présentation du Dataset 3
2.2 Exploration des Données 3
2.3 Analyse des Caractéristiques 4
2.4 Détection et Suppression des Outliers 4
2.5 Réduction de la Dimensionnalité 4
2.6 Préparation finale 4

3 Modélisation **5**
3.1 Approche Basée sur le Machine Learning Classique 5
3.2 Approche Séquentielle par Deep Learning 5
3.3 Ensemble Learning (Approche Hybride) 5

4 Évaluation des Modèles **6**
4.1 Résultats du modèle Random Forest 6
4.2 Résultats du modèle de Régression Logistique 6
4.3 Résultats des Modèles Séquentiels 7
4.3.1 Modèle CNN 1D 7
4.3.2 Justification de l'Approche Séquentielle 7
4.3.3 Perspectives : LSTM et Entraînement Étendu 7

5 Conclusion et Perspectives d'Amélioration **9**

1 Introduction

La prolifération des cyberattaques, notamment à travers le phishing et la distribution de malwares, représente une menace croissante pour la cybersécurité. Dans ce contexte, la détection précoce et fiable des URLs malveillantes est devenue une priorité stratégique. Les cybercriminels exploitent souvent des techniques d'obfuscation complexes pour dissimuler leurs intentions malicieuses, rendant les méthodes classiques de filtrage insuffisantes.

1.1 Problématique

Les URLs malveillantes peuvent imiter des sites légitimes en adoptant des structures très similaires, ce qui rend leur détection difficile. Une analyse superficielle basée uniquement sur des critères simples (présence de mots-clés, longueur de l'URL, etc.) ne suffit pas à capturer la complexité des techniques utilisées. Il devient donc crucial d'exploiter des approches plus avancées et capables d'apprendre automatiquement des modèles à partir des données.

1.2 Objectif du projet

Ce projet a pour objectif de mettre en place une approche d'apprentissage automatique pour la classification des URLs en deux classes : légitimes ou malveillantes. L'étude explore différentes méthodes classiques et profondes (deep learning) pour améliorer la précision de détection, en s'appuyant sur un dataset riche en caractéristiques extraites des URLs.

1.3 Intérêt du Machine Learning

Le machine learning se révèle particulièrement adapté dans ce contexte pour plusieurs raisons :

- Il permet de traiter un grand volume de données et d'identifier automatiquement les patterns associés aux attaques.
- Les modèles supervisés peuvent apprendre à partir d'exemples étiquetés, et donc reconnaître des structures malveillantes spécifiques.
- Il offre la possibilité d'utiliser aussi bien des caractéristiques tabulaires (statistiques, lexicales, structurelles) que la séquence brute de caractères via des réseaux de neurones.

2 Exploration et Préparation des Données

2.1 Présentation du Dataset

Le dataset utilisé comporte plus de 6,7 millions de lignes et 61 colonnes, chacune représentant une caractéristique spécifique d'une URL. Ces caractéristiques sont regroupées en plusieurs catégories : informations générales, structure de l'URL, analyse du domaine, présence de mots-clés sensibles, mesures statistiques (entropie, distances de Hamming), etc. La colonne `label` représente la classe cible : 1 pour les URLs malveillantes, 0 sinon.

```
data.head()
```

	url	label	source	url_has_login	url_has_client	url_has_server	url_has_admin	url_has_ip	url_isshorted	url_l
0	irs-profilepaymentservice.com/home	1	phishtank	0	0	0	0	0	0	
1	cpuggsukabumi.id	0	majestic_million	0	0	0	0	0	0	
2	members.tripod.com/~don_rc/ring.htm	0	data_clean_test_mendel	0	0	0	0	0	0	
3	optuswebmailadminprovider.weebly.com/	1	phishtank	0	0	0	1	0	0	
4	topoz.com.pl	0	dmoz_harvard	0	0	0	0	0	0	

5 rows × 60 columns

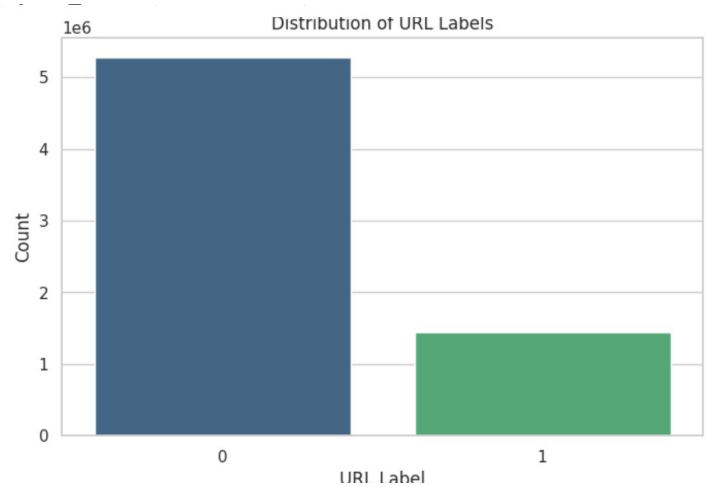
2.2 Exploration des Données

Plusieurs analyses exploratoires ont été menées :

— Le dataset ne contient aucune valeur manquante.

— Il s'agit d'un problème de classification binaire avec un déséquilibre des classes : environ 78% d'URLs légitimes (classe 0) et 22% malveillantes (classe 1).

— Aucune URL n'est dupliquée.



2.3 Analyse des Caractéristiques

Les premières étapes de "feature engineering" ont permis de visualiser et mieux comprendre la distribution des variables. Par exemple, certaines caractéristiques comme `url_length`, `count_dash` ou la présence d'IP ont montré des valeurs très élevées dans les cas d'URLs malveillantes.

2.4 Détection et Suppression des Outliers

Une méthode basée sur l'IQR (Interquartile Range) a été utilisée pour détecter les valeurs extrêmes. Plusieurs caractéristiques ont révélé des outliers significatifs, notamment :

- `url_length` : 501 109 outliers
- `count_dash` : 1 292 516 outliers
- `has_ip` : 32 531 outliers

Ces valeurs aberrantes ont ensuite été supprimées pour éviter qu'elles ne biaisent l'apprentissage.

2.5 Réduction de la Dimensionnalité

Un modèle de forêt aléatoire (*Random Forest*) a été utilisé pour calculer l'importance des variables. Seules les plus pertinentes ont été conservées pour la modélisation, permettant de :

- Réduire le surapprentissage
- Améliorer les performances
- Faciliter l'interprétation

2.6 Préparation finale

Les données nettoyées et filtrées ont été divisées en deux sous-ensembles : 80% pour l'entraînement et 20% pour les tests, avec une stratification pour respecter la distribution des classes.

3 Modélisation

3.1 Approche Basée sur le Machine Learning Classique

Deux modèles supervisés ont été entraînés sur les caractéristiques tabulaires extraites :

— **Régression Logistique** : utilisée comme modèle de base, simple et interprétable. Elle offre de bons résultats sur des données linéairement séparables.

— **Random Forest** : modèle robuste, capable de capturer des relations complexes. Il a été utilisé à la fois pour la classification et pour la sélection des caractéristiques.

Ces modèles se basent exclusivement sur des variables numériques résumant la structure des URLs.

3.2 Approche Séquentielle par Deep Learning

En parallèle, une approche orientée séquence a été développée en utilisant les chaînes de caractères brutes des URLs :

— Les URLs sont converties en minuscules puis tokenisées caractère par caractère.

— Chaque URL est représentée par une séquence d'entiers (via un dictionnaire d'indexation), normalisée à une longueur fixe (60 caractères).

Deux modèles profonds ont été envisagés :

— **CNN 1D (Convolutional Neural Network)** : capture les motifs locaux fréquents dans les URLs, comme des chaînes de caractères typiques de phishing.

— **LSTM (Long Short-Term Memory)** : proposé comme extension future, il permettrait de capter la dépendance à long terme entre caractères, mais n'a pas été implémenté par manque de ressources.

3.3 Ensemble Learning (Approche Hybride)

Enfin, une stratégie de fusion des prédictions a été prévue, malheureusement par contrainte de matériel non mise en place :

— **Voting pondéré** : chaque modèle donne une prédiction pondérée selon sa performance.

— **Stacking** : les prédictions de tous les modèles alimentent un méta-classificateur (comme une régression logistique) qui apprend à les combiner.

Cette combinaison permet de profiter à la fois de la richesse des caractéristiques tabulaires et de la puissance d'analyse des modèles séquentiels.

4 Évaluation des Modèles

4.1 Résultats du modèle Random Forest

Le modèle `RandomForestClassifier` a été entraîné sur les données tabulaires et évalué sur l'ensemble de test. Il a obtenu les performances suivantes :

Classe	**Précision**	**Rappel**	**F1-score**
Classe 0 (légitime)	0.94	0.98	0.96
Classe 1 (malveillante)	0.91	0.76	0.83
Accuracy globale		93.19%	
F1-score (macro)		0.89	
F1-score (pondéré)		0.93	

Ce modèle s'avère performant, notamment pour la détection des URLs légitimes. Le rappel plus faible pour la classe 1 indique que certains cas malveillants échappent à la détection.

4.2 Résultats du modèle de Régression Logistique

Le modèle `LogisticRegression` a été utilisé pour comparaison. Ses résultats sont résumés dans le tableau suivant :

Classe	**Précision**	**Rappel**	**F1-score**
Classe 0 (légitime)	0.89	0.96	0.92
Classe 1 (malveillante)	0.79	0.55	0.65
Accuracy globale		87.16%	
F1-score (macro)		0.78	
F1-score (pondéré)		0.86	

La régression logistique est moins performante que la forêt aléatoire, en particulier pour détecter les URLs malveillantes (rappel = 55%).

4.3 Résultats des Modèles Séquentiels

4.3.1 Modèle CNN 1D

Le modèle CNN 1D, dont l'architecture est décrite précédemment, a été entraîné sur un sous-ensemble du dataset en raison de limitations matérielles. L'entraînement a été réalisé sur 10 époques avec un split de validation interne de 10%. L'optimiseur utilisé est `adam` avec la fonction de perte `binary_crossentropy`.

Malgré l'absence de GPU et la contrainte de temps, le modèle a obtenu des performances très satisfaisantes :

— **Loss sur le test (CNN 1D)** : `0.1268`

— **Accuracy sur le test (CNN 1D)** : `0.9613` (96.13%)

Ces résultats démontrent que le CNN 1D est capable d'extraire efficacement des motifs textuels significatifs depuis les séquences de caractères des URLs, et ce malgré une architecture relativement simple.

4.3.2 Justification de l'Approche Séquentielle

L'analyse séquentielle repose sur l'idée que la structure d'une URL — notamment l'ordre des caractères — contient des motifs récurrents associés à des comportements malveillants. Deux architectures principales ont été explorées ou envisagées :

— **CNN 1D** : particulièrement efficace pour détecter des motifs locaux (comme les séquences `.exe`, `.zip`, etc.) indépendamment de leur position, grâce aux filtres convolutifs.

— **LSTM** : ce type de réseau récurrent est capable de modéliser les dépendances à long terme dans la séquence, prenant en compte le contexte précédent et suivant chaque caractère. Cela permettrait d'analyser des structures plus complexes telles que les chemins d'accès profonds ou les chaînes de requêtes dynamiques.

Les couches d'`Embedding` facilitent cet apprentissage en produisant des représentations vectorielles denses et informatives à partir des caractères.

4.3.3 Perspectives : LSTM et Entraînement Étendu

L'intégration d'un modèle LSTM complet n'a pas pu être réalisée dans ce projet en raison de contraintes techniques (absence de GPU, temps d'entraînement important). Néanmoins, son utilisation reste une perspective prometteuse pour améliorer la robustesse et la capacité de généralisation du système.

Une extension future pourrait également inclure :

— Un entraînement sur le dataset complet.

— L'utilisation de modèles bidirectionnels (BiLSTM).

— L'intégration d'un mécanisme d'attention.

5 Conclusion et Perspectives d'Amélioration

Ce projet a permis d'explorer différentes approches pour la détection automatique d'URLs malveillantes, en combinant des techniques classiques de machine learning avec des méthodes plus avancées de deep learning.

Les modèles entraînés sur les caractéristiques tabulaires ont donné de bons résultats, notamment le modèle Random Forest qui a atteint une précision globale de 93.19%. Toutefois, son rappel pour la classe des URLs malveillantes reste limité (76%).

L'introduction d'un modèle CNN 1D a permis d'atteindre une précision plus élevée (96.13%), en exploitant directement la structure séquentielle des URLs. Cette approche a montré que les réseaux de neurones sont capables d'apprendre automatiquement des motifs textuels discriminants.

Perspectives d'Amélioration

Plusieurs axes d'amélioration peuvent être envisagés :

- **Utilisation de GPU** pour entraîner des modèles plus complexes (ex. : LSTM, modèles bi-directionnels).
- **Optimisation des hyperparamètres** via des méthodes de recherche systématique ou bayésienne.
- **Fusion plus poussée des modèles hybrides**, en testant d'autres stratégies d'ensemble (Boosting, modèles empilés profonds).
- **Analyse de la robustesse** des modèles face à des attaques adversariales ou des URLs générées automatiquement.
- **Extension à d'autres langues ou alphabets**, en rendant la tokenisation multilingue.

En conclusion, le machine learning, et en particulier le deep learning, offre des outils puissants pour automatiser la détection de menaces en ligne. Ce travail constitue une base solide pour des systèmes de cybersécurité plus intelligents et adaptatifs.

Remerciements

Nous tenons à exprimer notre profonde gratitude à **Monsieur Younes Tabii**, notre cher professeur, pour **ses efforts continus en cours magistraux et en projets pratiques**, qui ont grandement contribué à la réussite de ce projet.