

Université Sultan Moulay Slimane

Faculté Polydisciplinaire Département de Mathématique et Informatique

- Béni Mellal -

Master : **Système de Télécommunication et Réseaux Informatique**

\*\*\*\*\*

Mini Projet

Module : Sécurit des réseaux et du Web

Towards Benchmark Datasets for Machine Learning Based Website Phishing Detection: An Experimental Study

Réaliser par : Amine Nisrine Encadrer par : SADQI Yassine

Ouadi Hajar CHAKIR Oumaima

Ouadi Reda

Année Universitaire : **2024-2025**

TABLE DES MATIÈRES

**[Introduction Général](#_Toc186309692)** [5](#_Toc186309692)

[1.0. Contexte Et Problème 5](#_Toc186309693)

[1.1. Contribution Principale de l'Article 6](#_Toc186309694)

**[Chapitre 1 :](#_Toc186309695)** [Description des attaques de phishing et leurs mécanismes 7](#_Toc186309695)

[1.1. Les attaques de phishing 7](#_Toc186309696)

**[1.1.1. Principe de fonctionnement](#_Toc186309697)** [7](#_Toc186309697)

**[1.1.2. Aspects des attaques de phishing](#_Toc186309698)** [8](#_Toc186309698)

**[1.1.3. Comparaison Attaques Phishing avec d'Autres Attaques](#_Toc186309699)** [9](#_Toc186309699)

**[1.1.4. Exemple d’un lien de phishing](#_Toc186309700)** [11](#_Toc186309700)

[1.2. Mécanismes des attaques de phishing 11](#_Toc186309701)

[1.3. Conséquences des attaques : 12](#_Toc186309703)

[1.4. Conclusion : 12](#_Toc186309704)

[CHAPITRE 2 : Le rôle du machine Learning dans la détection des attaques de phishing : Comparaison des méthodes basées sur la machine Learning avec les méthodes traditionnelles. 13](#_Toc186309705)

[2.1. Introduction 13](#_Toc186309706)

[2.2. Méthodes traditionnelles de détection 13](#_Toc186309707)

**[2.2.1. Listes noires et blanches :](#_Toc186309708)** [13](#_Toc186309708)

**[2.2.2. Approches heuristiques :](#_Toc186309709)** [13](#_Toc186309709)

[2.3. Rôle et avantages du machine Learning 13](#_Toc186309710)

**[2.3.1. Basées sur l'URL :](#_Toc186309711)** [13](#_Toc186309711)

**[2.3.2. Basées sur le contenu :](#_Toc186309712)** [14](#_Toc186309712)

**[2.3.3 Basées sur des services externes :](#_Toc186309713)** [14](#_Toc186309713)

[2.4. Algorithmes utilisés : 14](#_Toc186309714)

**[2.4.1. Random Forest, SVM, Logistic Regression, Decision Tree.](#_Toc186309715)** [14](#_Toc186309715)

**[2.4.2. Random Forest s'est avéré le plus performant avec une précision de 96,61 %.](#_Toc186309716)** [14](#_Toc186309716)

[2.5. Comparaison des performances : 14](#_Toc186309717)

**[2.5.1. Les méthodes traditionnelles sont rapides mais limitées à des attaques connues.](#_Toc186309718)** [14](#_Toc186309718)

**[2.5.2. Le ML est adaptatif et efficace pour détecter des attaques inconnues.](#_Toc186309719)** [14](#_Toc186309719)

[2.6](#_Toc186309720)**[.](#_Toc186309720)** [Exemple de mise en œuvre pratique 14](#_Toc186309720)

[](#_Toc186309721) **[Pour les approches basées sur le contenu :](#_Toc186309721)** [14](#_Toc186309721)

[](#_Toc186309722) **[Pour les approches basées sur l'URL :](#_Toc186309722)** [14](#_Toc186309722)

**[2.6.1. Tableau comparatif des méthodes](#_Toc186309723)** [14](#_Toc186309723)

**[2.6.2. Défis et limitations](#_Toc186309724)** [15](#_Toc186309724)

[2.7. Conclusion 15](#_Toc186309725)

[CHAPITRE 3 : La contribution des auteurs : Décrivez ce que l'article propose comme solution ou innovation pour la détection des attaques de phishing 15](#_Toc186309726)

[3.1. Introduction 16](#_Toc186309727)

[3.2. SCHÉMA DE CONSTRUCTION D'ENSEMBLES DE DONNÉES REPRODUCTIBLES ET EXTENSIBLES 16](#_Toc186309728)

[3.3. Méthodologie de collecte et de préparation de l'ensemble des données pour l'analyse des attaques de phishing 17](#_Toc186309729)

[3.3.1. URL collection 18](#_Toc186309730)

[3.3.3. Génération des ensembles de données 25](#_Toc186309731)

[3.4. METHODOLOGIE DE RECHERCHE 25](#_Toc186309732)

**[3.4.1. Classifier évaluation](#_Toc186309733)** [27](#_Toc186309733)

**[3.4.2. Évaluation des méthodes de sélection des caractéristiques](#_Toc186309734)** [28](#_Toc186309734)

[3.4.2. Expérience V : Analyse de la durée d'exécution de l'extraction de caractéristiques 30](#_Toc186309735)

[3.5. ANALYSE DES RÉSULTATS EXPÉRIMENTAUX 31](#_Toc186309736)

**[3.5.1. Expérience I - Performance des Classificateurs sur Différentes Classes de Caractéristiques](#_Toc186309737)** [31](#_Toc186309737)

**[3.5.2. Expérience II. Performance des modèles combinés formés chacun sur une classe différente de caractéristiques](#_Toc186309738)** [32](#_Toc186309738)

**[3.5.3. Expérience III. Performance par formation sur les fonctionnalités sélectionnées à l'aide de méthodes de classement par filtre](#_Toc186309739)** [33](#_Toc186309739)

**[3.5.4. Expérience IV. Performance par la formation sur les fonctionnalités sélectionnées à l'aide des méthodes d'emballage](#_Toc186309740)** [36](#_Toc186309740)

**[3.5.5. Expérience V. Analyse de l'exécution de l'extraction de caractéristiques](#_Toc186309741)** [37](#_Toc186309741)

[3.6. Rapport d'évaluation et résultats 38](#_Toc186309742)

[3.7. Conclusion 39](#_Toc186309743)

[CHAPITRE 4 : Analyse comparative avec les travaux existants 40](#_Toc186309744)

[4.1. Introduction 40](#_Toc186309745)

[4.2. TRAVAUX CONNEXES 40](#_Toc186309746)

[CONCLUSION GENIRALE 42](#_Toc186309747)

[Références 43](#_Toc186309748)

TABLE DES FIGURES

Figure 1 : Étapes d'une attaque de phishing7

Figure 2 : composant d’un URL8

Figure 3 : Nombre d’emails de phishing au 3e trimestre 20229

Figure 4 : Nombre d’emails de phishing d’Instagram9

Figure 5 : Email de phishing Capital One / Authentify10

Figure 6 : Utilisation d’emails pour le phishing11

Figure 7 : Processus de construction d'un ensemble de données.17

Figure 8 la Classification of Ant phishing features18

Figure 9 Processus d'identification des meilleurs classificateurs et de la meilleure combinaison de classes d'entités.27

Figure 10 Combinaison de modèles formés sur différentes classes de fonctionnalités.28

Figure 11 Processus d'identification des meilleurs filtres et des sous-ensembles de caractéristiques.29

Figure 12 Processus d'identification pour le meilleur algorithme d'encapsulation pour la sélection des fonctionnalités.30

Figure 13 Performance des classificateurs formés sur la classe individuelle des caractéristiques.31

Figure 14 Performance des classificateurs formés sur des classes de fonctionnalités combinées par paires.32

Figure 15 Performance des modèles combinés.33

Figure 16 Précision du modèle de prédiction aléatoire des forêts par rapport à la sélection par étapes des caractéristiques.35

Figure 17 Temps moyen pour l'extraction des caractéristiques37

Figure 18 Captures d'écran du plugin passif développé pour le navigateur Chrome39

Figure 19 Nombre de publications par an portant sur l'utilisation de l'apprentissage automatique pour la détection des sites web d'hameçonnage. 40

LISTE DES TABLEAUX

Table 1 Comparaison entre les méthodes basées sur le machine Learning avec les méthodes traditionnelles.14

Table 2 List of reviews used for features17

Table 3 Caractéristiques basées sur l'URL pour la détection de l'hameçonnage de sites web19

Table 4 Caractéristiques basées sur le contenu pour la détection des sites web d'hameçonnage21

Table 5 Caractéristiques externes pour la détection des sites web d'hameçonnage24

Tableau 6 Les 25 caractéristiques les mieux classées pour chaque filtre utilisé34

Tableau 7 Liste des caractéristiques moins importantes identifiées par les filtres de classement utilisés35

Table 8 Résultats des évaluateurs du wrapper.36

# **Introduction Général**

## 1.0. Contexte Et Problème

L’essor rapide du commerce électronique a révolutionné notre manière de consommer, offrant commodité et accessibilité aux consommateurs du monde entier. Cependant, cette croissance a également entraîné des défis de sécurité importants, principalement dus à la cybercriminalité et aux attaques de phishing. Dans ce contexte numérique en constante évolution, la vérification de la légitimité des pages web devient un enjeu crucial pour protéger les clients et prévenir les attaques sournoises qui exploitent la confiance des utilisateurs.

Les attaques de phishing, désormais reconnues comme l'une des menaces cybercriminelles les plus répandues et pernicieuses, contournent souvent les défenses traditionnelles en utilisant des tactiques trompeuses plutôt que des méthodes techniques sophistiquées. Les cybercriminels, au lieu de se heurter à des codes complexes ou à des pare-feu robustes, adoptent des approches plus subtiles. Ils recourent à des courriels émotionnels, urgents ou sensibles, incitant les destinataires à divulguer en toute confiance des informations personnelles cruciales par le biais de liens malveillants.

Face à ces défis, l'utilisation répandue de l'apprentissage automatique se profile comme une solution prometteuse. La littérature abonde en études qui explorent les capacités de cette technologie pour détecter et prévenir les attaques de phishing de manière proactive. Cependant, l'efficacité de ces modèles demeure souvent dépendante des jeux de données utilisés, soulevant ainsi la nécessité cruciale de créer des références fiables pour évaluer objectivement les systèmes de détection.

Dans cette perspective, cette étude s'inscrit dans le contexte de la sécurité croissante des transactions en ligne. Nous mettrons en lumière les défis inhérents à la croissance du commerce électronique, soulignant le rôle critique de la vérification de la légitimité des pages web pour renforcer la sécurité des utilisateurs. En outre, nous explorerons la voie prometteuse de l'apprentissage automatique comme une stratégie cruciale pour faire face aux évolutions constantes des tactiques de phishing. Pour atteindre cet objectif, nous baserons notre étude sur un article scientifique clé rédigé par A. Hannousse et S. Yahiouche, publié en 2020. Cette référence académique constitue le fondement de notre démarche, garantissant la validité et la pertinence de notre approche.

## 1.1. Contribution Principale de l'Article

L'article en question aborde de manière significative la problématique de la détection de phishing en mettant l'accent sur la nécessité de créer des jeux de données de référence reproductibles et extensibles. Les auteurs soulignent les défis liés à la dépendance aux jeux de données existants, mettant en lumière l'absence de réplicabilité et le manque de jeux de données appropriés pour une évaluation équitable des systèmes de détection de phishing. Les contributions principales de article sont les suivantes :

* Proposition et expérimentation d'un schéma novateur pour la construction de jeux de données reproductibles et extensibles dédiés à la détection de phishing sur les pages web.
* Proposition d'une classification affinée des caractéristiques (features) pour la détection de phishing.
* Examen des conclusions antérieures concernant les performances des classificateurs et la pertinence des caractéristiques pour la détection en temps réel.
* Expérimentation de différentes méthodes de sélection de caractéristiques.
* Identification du modèle offrant la meilleure précision pour le dataset collecté.

Ces contributions visent à améliorer la fiabilité et l'efficacité des systèmes de détection de phishing, en fournissant des bases solides pour des évaluations objectives et reproductibles

# 

# **Chapitre 1 : Description des attaques de phishing et leurs mécanismes**

## 1.1. Les attaques de phishing

Les attaques de phishing sont parmi les menaces de cybercriminalité les plus courantes et les plus faciles à exécuter. Contrairement aux attaques complexes nécessitant le piratage de codes de chiffrement ou de pare-feu robustes, le phishing repose principalement sur la manipulation psychologique et l'ingénierie sociale.

### **1.1.1. Principe de fonctionnement**

Le phishing est un problème de classification, car il s'agit de distinguer entre des sites web légitimes et des sites frauduleux. Pour détecter les attaques de phishing, nous utilisons l'apprentissage supervisé. Cette technique entraîne des modèles sur des ensembles de données étiquetées, où chaque exemple est marqué comme légitime ou phishing. En analysant des caractéristiques spécifiques des sites web, ces modèles peuvent apprendre à identifier les schémas typiques des attaques de phishing et à les distinguer des sites authentiques.

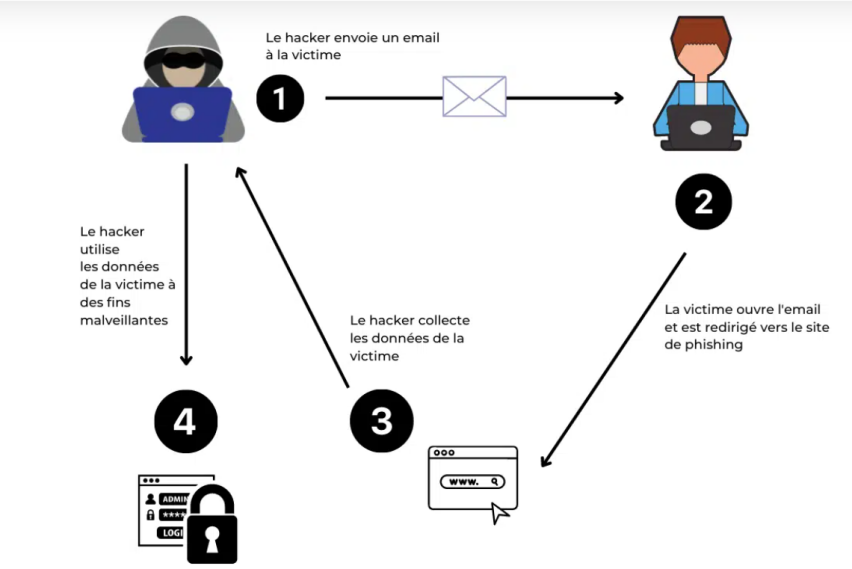


Figure 1 : Étapes d'une attaque de phishing

L'attaquant envoie un email à la victime, se faisant passer pour une entreprise ou une organisation légitime. L'e-mail contient généralement un lien ou une pièce jointe qui, lorsqu'il est cliqué ou ouvert, dirige la victime vers un site Web malveillant. Le site Web malveillant est conçu pour ressembler à un site Web légitime. Il peut utiliser le logo et la mise en page du site Web légitime, ainsi que des messages d'urgence ou alarmants pour inciter la victime à entrer ses informations sensibles. Alors une fois que la victime a entré ses informations sensibles, l'attaquant les utilise pour accéder à ses comptes ou voler son identité (Figure 1).

### **1.1.2. Aspects des attaques de phishing**

Les attaques de phishing sont de plus en plus sophistiquées, et il en existe deux types principaux :

**Phishing par URL** : L'attaque de phishing peut être détectée en analysant les URLs et le contenu des pages Web. Il existe deux approches principales pour détecter ces attaques:

* **URL-based phishing :** Cette méthode se concentre sur l’analyse des URLs des sites Web visités. Des caractéristiques comme la longueur de l'URL, la présence de caractères spéciaux, ou l'utilisation d'adresses IP au lieu de noms de domaine sont des indicateurs importants.

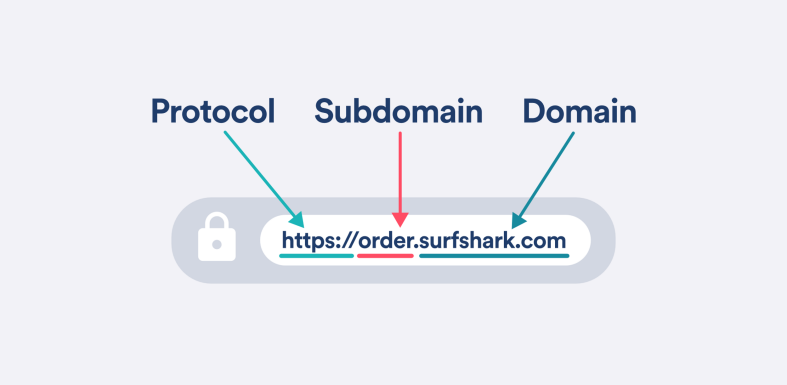


Figure 2 : composant d’un URL

**Phishing par contenu** : Analyse du contenu des pages Web (ex : liens internes/externes, balises HTML).

### **1.1.3. Comparaison Attaques Phishing avec d'Autres Attaques**

Le phishing a été reconnu comme la menace de cybercriminalité la plus facile et la plus répandue. Les pirates n'ont pas besoin de déchirer un code complexe ni de violer un pare-feu dur. Au lieu de cela, ils envoient simplement des courriels émotionnels, critiques ou sensés exhortant les destinataires à présenter leurs informations d'identification personnelles en cliquant sur un lien. Les destinataires sont ensuite redirigés vers de fausses pages Web qui ressemblent beaucoup à ces sites Web authentiques ciblés. Les attaques de phishing diffèrent de plusieurs autres types d'attaques informatiques en raison de leur approche et de leurs objectifs spécifiques. Comparé à d'autres types d'attaques, le phishing se distingue par son utilisation de l'ingénierie sociale pour tromper les victimes. Contrairement aux attaques purement techniques, le phishing repose sur la manipulation psychologique des individus pour obtenir des informations sensibles ou les inciter à prendre des mesures préjudiciables. De plus, le phishing est souvent le point de départ d'attaques plus complexes, telles que les menaces persistantes avancées (APT) et les ransomwares, ce qui en fait une menace majeure pour les organisations.

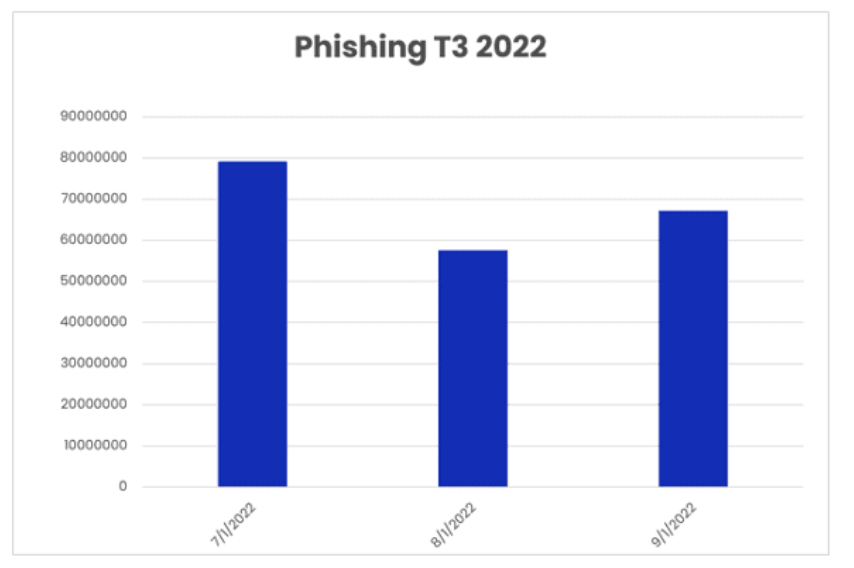


Figure 3 : Nombre d’emails de phishing au 3e trimestre 2022



Figure 4 : Nombre d’emails de phishing d’Instagram

### **1.1.4. Exemple d’un lien de phishing**

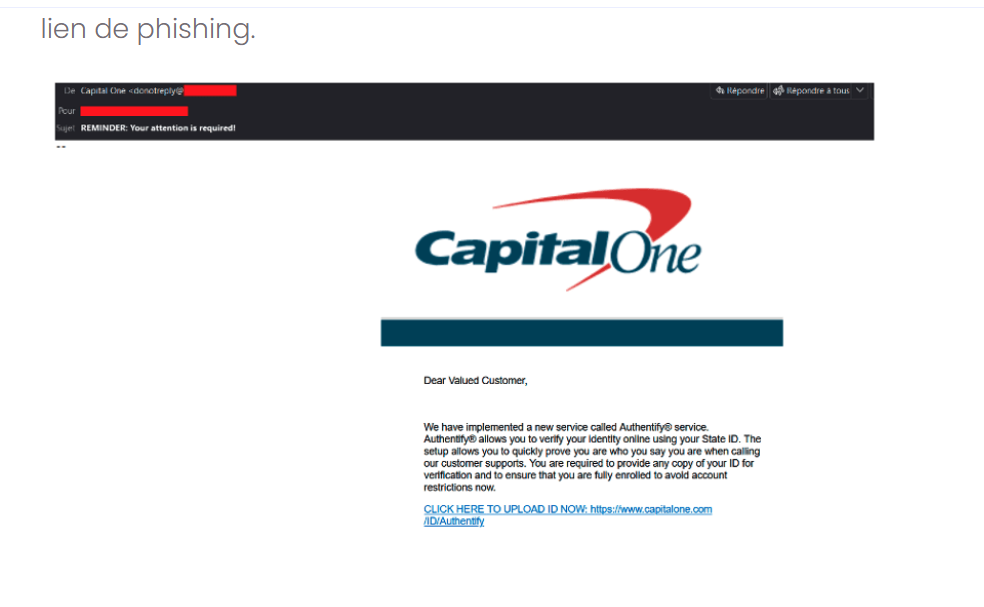


Figure 5 : Email de phishing Capital One / Authentify

Les hackers camouflent le lien à l’aide de texte contenant les noms de Capital One et d’Authentify, mais en réalité, l’URL pointe vers un site Web compromis se faisant passer pour Capital One. Les utilisateurs y sont invités à envoyer une copie du recto et du verso d’une pièce d’identité.

## **1.2. Mécanismes des attaques de phishing**

Le phishing vise à tromper les utilisateurs afin qu'ils divulguent des informations sensibles telles que leurs identifiants, mots de passe ou données bancaires. Cela se fait généralement par l'envoi de courriels ou de messages présentant une apparence légitime mais contenant des liens vers des sites web frauduleux. Ces sites imitent à la perfection des portails officiels pour inspirer confiance.

Dans l’article analysé, les mécanismes à l’œuvre dans les différentes attaques de phishing s’appuient généralement sur la tromperie des utilisateurs et l’exploitation de leur crédulité pour obtenir leurs informations personnelles. Voici les mécanismes principaux :

Imitation de sites légitimes : Le phishing utilise des sites web faux imitant à l’identique, ou presque, l’apparence d’un site légitime pour obtenir des informations sensibles saisies par les utilisateurs (identifiants, mots de passe ou coordonnées bancaires, etc.).

Utilisation d’emails pirate : Le phishing commence souvent par des messages à l’apparence officielle d’une source supposée légitime, et l’invitation au clic sur des liens malveillants menant vers une page web de phishing, elle-même faussement légitime.

### IMG_256

Figure 6 : Utilisation d’emails pour le phishing

SSL pour crédibilité : De plus en plus, les sites de phishing utilisent des certificats SSL (HTTPS), réservés jusqu’alors à des sites jugés légitimes, pour donner un aspect plus crédible, moins suspect au site frauduleux.

Exploitation de la peur ou de l’urgence : Les mails ou sites de phishing exploitent la peur ou l’urgence pour inciter les utilisateurs à agir vite, une notification de problème sur leur compte leur demandant une action immédiate par exemple.

En résumé, il s’agit pour ces attaques d’instrumentaliser la crédibilité des sites Web et l’urgence perçue par l’utilisateur pour lui obtenir ses informations personnelles.

## **1.3. Conséquences des attaques :**

Les victimes subissent des pertes financières, le vol de leurs identités, et parfois des dommages à long terme sur leur réputation numérique. Pour les entreprises, les attaques peuvent entraîner des fuites de données sensibles et des pertes de confiance des clients.

### **1.4. Conclusion :**

Le phishing reste une menace évolutive, adaptée aux nouvelles technologies et tactiques. La recherche mentionnée souligne l'importance de la détection automatisée des sites frauduleux via des méthodes d'apprentissage automatique. En sensibilisant les utilisateurs et en adoptant des solutions avancées, il est possible de réduire l'impact de ces attaques.

# CHAPITRE 2 : Le rôle du machine Learning dans la détection des attaques de phishing : Comparaison des méthodes basées sur la machine Learning avec les méthodes traditionnelles.

## 2.1. Introduction

Le machine Learning (ML) est une branche de l'intelligence artificielle qui permet à un système d'apprendre et de s'améliorer automatiquement à partir de données sans être explicitement programmé. Les algorithmes de ML peuvent analyser des ensembles de données étiquetés pour identifier des modèles, les généraliser et faire des prédictions sur de nouvelles données.

Dans le contexte de la détection des attaques de phishing, le ML transforme cette tâche en un problème de classification, permettant de distinguer les sites web légitimes des sites frauduleux en se basant sur leurs caractéristiques.

## 2.2. Méthodes traditionnelles de détection

### **2.2.1. Listes noires et blanches :**

i. Les listes noires bloquent les URL déjà identifiées comme malveillantes, tandis que les listes blanches autorisent uniquement les URL jugées fiables.

ii. Limites : inefficaces pour les nouvelles attaques, nécessitent des mises à jour fréquentes.

### **2.2.2. Approches heuristiques :**

1. Utilisent des règles prédéfinies basées sur des caractéristiques comme la longueur des URL ou la présence de caractères spéciaux.

ii. Limites : rigides et incapables de s'adapter à l'évolution des tactiques de phishing.

## 2.3. Rôle et avantages du machine Learning

### **2.3.1. Basées sur l'URL :**

1. Les listes noires bloquent les URL déjà identifiées comme malveillantes, tandis que les listes blanches autorisent uniquement les URL jugées fiables.
2. Limites : inefficaces pour les nouvelles attaques, nécessitent des mises à jour fréquentes.

### **2.3.2. Basées sur le contenu :**

i. Analyse des balises HTML, des scripts suspects et des redirections multiples.

### **2.3.3 Basées sur des services externes :**

1. Informations issues de WHOIS, popularité du domaine, historique du site.

## 2.4. Algorithmes utilisés :

### **2.4.1. Random Forest, SVM, Logistic Regression, Decision Tree.**

### **2.4.2. Random Forest s'est avéré le plus performant avec une précision de 96,61 %.**

## 2.5. Comparaison des performances :

### **2.5.1. Les méthodes traditionnelles sont rapides mais limitées à des attaques connues.**

### **2.5.2. Le ML est adaptatif et efficace pour détecter des attaques inconnues.**

## 2.6**.** Exemple de mise en œuvre pratique

Récemment, Das et al. ont mené un examen systématique des études publiées entre 2010 et 2017 qui ont utilisé des techniques de machine learning pour détecter les attaques de phishing. Ils ont révélé que :

### **Pour les approches basées sur le contenu :**

Les algorithmes SVM et Logistic Regression étaient les plus utilisés.

### **Pour les approches basées sur l'URL :**

Décision Tree et Random Forest dominaient en termes de performance.

Par ailleurs, des jeux de données comme le référentiel UCI ont été employés pour entraîner les modèles, bien que ces ensembles souffrent parfois de limitations comme l'absence d'URL spécifiques ou la nature éphémère des sites de phishing.

.

### **2.6.1. Tableau comparatif des méthodes**

|  |  |  |
| --- | --- | --- |
| **Critère** | **Méthodes traditionnelles** | **Machine Learning** |
| **Adaptabilité** | Limitée aux attaques connues | Capable de détecter des attaques inconnues |
| **Maintenance** | Mises à jour fréquentes nécessaires | Nécessite un entraînement initial des modèles |
| **Précision** | Moyenne | Élevée (jusqu'à 96,61 %) |
| **Complexité de mise en œuvre** | Faible | Élevée (extraction et sélection des données) |
| **Temps de détection** | Rapide | Variable selon les modèles utilisés |

Table 1 Comparaison entre les méthodes basées sur la machine Learning avec les méthodes traditionnelles.

### **2.6.2. Défis et limitations**

#### **2.6.2.1**.** Qualité des données :**

Les modèles de ML nécessitent des jeux de données représentatifs et diversifiés pour un entraînement efficace. Cependant, la collecte de données fiables reste un défi, notamment en raison de la nature éphémère des sites de phishing.

#### **2.6.2.2**.** Consommation de ressources :**

Les techniques de ML, en particulier les modèles complexes comme Random Forest, peuvent nécessiter des ressources matérielles importantes pour le traitement et l'entraînement.

#### **2.6.2.3 Biais des modèles :**

Les modèles peuvent être biaisés si les ensembles de données utilisés pour l'entraînement ne reflètent pas fidèlement les caractéristiques des attaques actuelles.

## 2.7. Conclusion

Le machine Learning révolutionne la détection des attaques de phishing en offrant des modèles adaptatifs capables de généraliser des modèles d'attaque et de les détecter en temps réel. Les méthodes ML complètent les approches traditionnelles et permettent une détection proactive, réduisant ainsi les impacts des attaques

# 

# CHAPITRE 3 : La contribution des auteurs : Décrivez ce que l'article propose comme solution ou innovation pour la détection des attaques de phishing

## 3.1. Introduction

La détection des attaques par hameçonnage est un défi majeur en matière de cybersécurité. Les méthodes de détection traditionnelles, qui reposent souvent sur des listes négatives ou des caractéristiques déjà définies, sont limitées par leur incapacité à s'adapter aux nouvelles techniques d'attaque. Dans leur article, les auteurs proposent une solution innovante basée sur l'apprentissage automatique pour remédier à ces limitations. Cette méthode vise à améliorer la précision et la généralisation des systèmes de détection afin de réduire efficacement les risques liés à l'hameçonnage.

## 3.2. SCHÉMA DE CONSTRUCTION D'ENSEMBLES DE DONNÉES REPRODUCTIBLES ET EXTENSIBLES

Les auteurs soulignent que les ensembles de données existants pour la détection des sites d'hameçonnage sont déficients, en raison de la courte durée de vie des sites d'hameçonnage et de l'absence de liens actifs, ce qui empêche une analyse basée sur le contenu. En outre, les ensembles de données existants manquent souvent d'informations sur les URL utilisés, ce qui limite la répétabilité et l'extensibilité des recherches.

Les chercheurs proposent des lignes directrices pratiques pour la création d'ensembles de données fiables et évolutifs pouvant servir de référence dans ce domaine :

* G1 : Collecte d'URLs provenant de sources diverses

Il est conseillé de commencer par collecter des URLs provenant de différentes sources. Il a été observé que de nombreuses études collectent des sites légitimes à partir d'Alexa. Cependant, Alexa ne propose que des domaines classés parmi les meilleurs sans mentionner les sous-domaines ni les chemins. Par conséquent, pour assurer l'homogénéité de l'ensemble de données, on ne peut pas utiliser directement les listes d'Alexa, en particulier lorsque les caractéristiques concernent les sous-domaines et les chemins. Pour résoudre ce problème, il est proposé d'utiliser la liste des principaux domaines fournie par Alexa comme point de départ pour explorer plus d'URLs génératives. Pour diversifier les URLs, il est aussi possible d'utiliser des URLs issues des catégories Alexa.

* G2 : Prétraitement des URLs collectées

Les URLs collectées doivent être prétraitées. Cette étape de prétraitement inclut : (1) la suppression des URLs en double, (2) éviter d'utiliser trop d'URLs provenant du même domaine, car ces URLs risquent de partager des caractéristiques similaires. Cela permet d'obtenir des échantillons plus représentatifs.

* G3 : Suivi des URLs utilisées lors de l'extraction des caractéristiques

Lors de l'extraction des caractéristiques, il est essentiel de garder une trace des URLs utilisées, soit en les utilisant comme index pour l'ensemble de données, soit en les sauvegardant dans un fichier séparé et en les mettant à disposition des utilisateurs. Cela permet de reproduire les caractéristiques basées sur les URLs et d'étendre l'ensemble de données principal avec de nouvelles caractéristiques.

* G4 : Génération de l'arborescence DOM (Document Object Model)

Pour surmonter la courte durée de vie des pages d’hameçonnage(phishing), il est possible d'utiliser des outils disponibles pour générer l'arbre du modèle d'objet de document (DOM) des pages et de les sauvegarder dans l'ensemble de données ou dans un ensemble de données séparé indexé par URLs. Cela permet d'expérimenter et d'étendre l'ensemble de données principal avec plus de caractéristiques basées sur le contenu, même avec des liens morts.

* G5 : Mélange et équilibrage des ensembles de données

Les ensembles de données finaux doivent être mélangés et équilibrés afin d'éviter le problème signalé [1] concernant l'impact des ensembles de données non équilibrés sur les performances des classificateurs.

* G6 : Datation des ensembles de données collectées

Les ensembles de données collectées doivent être actualisés. De cette manière, les systèmes peuvent être testés sur des ensembles de données collectées sur différentes périodes raisonnables afin de vérifier l'efficacité des modèles et la pertinence des caractéristiques adoptées. Cela permet également de suivre l'évolution des tactiques d'hameçonnage dans le temps.

## 3.3. Méthodologie de collecte et de préparation de l'ensemble des données pour l'analyse des attaques de phishing

Dans cette section, Hannousse et Yahiouche utilisent le schéma proposé dans la section 3.2. Pour collecter un ensemble de données qui sera utilisé ultérieurement pour les expériences menées dans cette étude. Ils ont ainsi collecté un jeu de données de taille raisonnable, composé de 11430 URLs de phishing et légitimes. La Figure illustre le processus global adopté pour la construction du jeu de données.

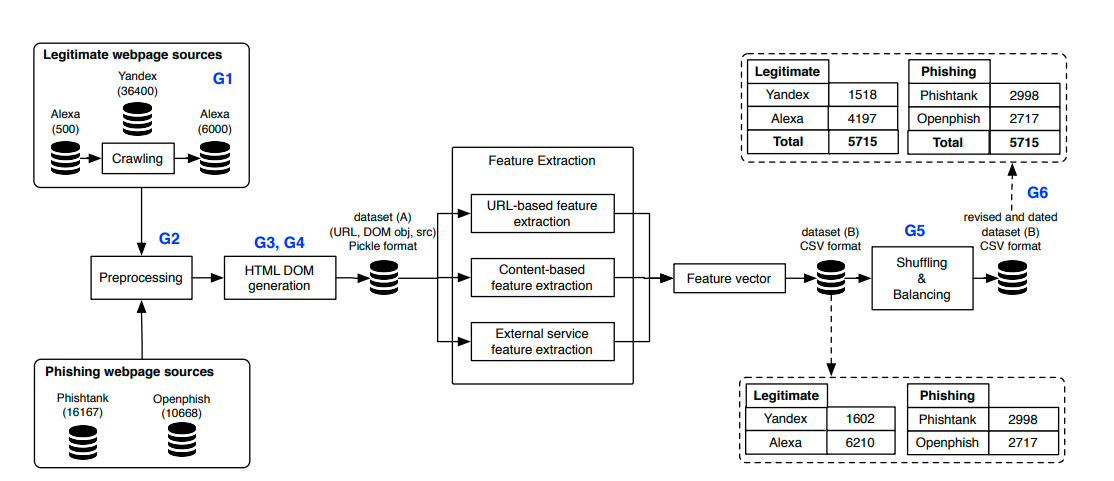


Figure 7 : Processus de construction d'un ensemble de données.

Cette section décrit le processus de collecte des URL pour l'étude. Il s'agit de collecter des URL légitimes à partir d'Alexa et de Yandex [2], ainsi que des URLs de phishing à partir de Phishtank et d'Openphish. Les URLs dupliquées et mortes sont supprimées, et un maximum de 12 URLs par domaine est conservé. Les URLs approuvées sont stockées avec leurs informations de source. De plus, des arbres de modèles d'objets du document HTML (DOM) sont générés pour toutes les URLs approuvées et stockés dans un ensemble de données distinct.

### 3.3.1. URL collection

L'étude est d'expérimenter un nombre maximal de caractéristiques populaires. Pour ce faire, Hannousse et Yahiouche ont examiné des revues récentes analysant la fréquence d'utilisation des caractéristiques pour la détection de sites de phishing. La liste des revues examinées est donnée dans le tableau 2. L'examen a montré que plusieurs caractéristiques sont données avec des noms et/ou des mesures déférentes (par exemple, ratio ou nombre absolu). Pour atténuer ces problèmes, les auteurs ont exécuté de manière itérative le processus en deux étapes suivantes, ce qui leur a permis d'obtenir 87 caractéristiques : (1) identifier et supprimer les caractéristiques en double, et (2) adopter une mesure unique et couramment utilisée pour chaque caractéristique.

|  |  |  |
| --- | --- | --- |
| Reviews | Year | Features |
| Korkmaz et al.[3] | 2020 | 120 |
| Das et al.[4] | 2020 | 88 |
| El-Assal et al.[1] | 2020 | 83 |
| Althobaiti et al.[5] | 2019 | 40 |

Table 2 List of reviews used for features

Hannousse et Yahiouche ont décrit aussi le diagramme de caractéristiques présenté à la Figure 8,

Qui illustre les catégories de caractéristiques couramment utilisées par les systèmes anti phishing existants

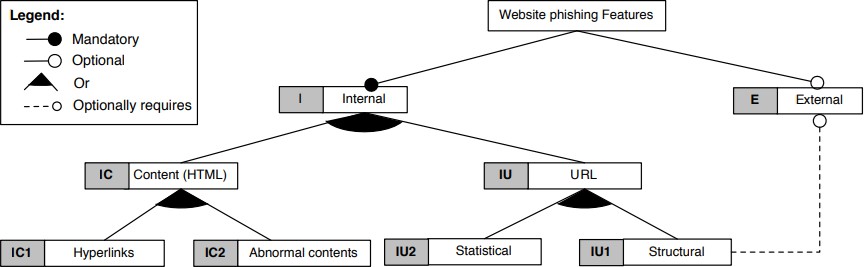


Figure 8 la Classification of Ant phishing features

Ces catégories de caractéristiques sont les suivantes :

* Caractéristiques internes (I) : Caractéristiques extraites directement des URL (IU) ou du contenu HTML des pages Web correspondantes (IC).
* Caractéristiques externes (E) : Caractéristiques obtenues par interrogation de services externes avec des URL ou des domaines d'URL.

Hannousse et Yahiouche modélisent cette classification dans la Figure 8 en utilisant les relations suivantes :

* Relation obligatoire avec les caractéristiques internes
* Relation facultative avec les caractéristiques externes
* Relation Or entre les caractéristiques internes IC et IU, ce qui signifie que les systèmes anti phishing hybrides utilisent une collection de caractéristiques des deux catégories.

Hannousse et Yahiouche, décrivent ensuite le processus d'extraction des caractéristiques, qui est entièrement automatisé à l'aide d'un script Python. Ce script prend l'URL d'une page Web en entrée et génère son vecteur de caractéristiques correspondant de taille 87x1. Pour garder une trace de l'URL source et la rendre disponible pour une utilisation ultérieure, les auteurs utilisent l'URL d'entrée comme index pour l’ensemble de données généré.

#### 3.3.2.1. Fonctionnalités basées sur l'URL

Les caractéristiques basées sur les URLs sont obtenues en analysant simplement le texte des URLs. Ces caractéristiques peuvent également être divisées en caractéristiques structurelles et statistiques.

* Caractéristiques structurelles (IU1) : Ces caractéristiques sont concernées par la présence, la position et la nature des éléments de base de l'URL (protocole, domaine, sous-domaines, chemin, port et domaine de premier niveau). Des exemples de ces caractéristiques sont la présence de ports, l'utilisation du protocole "https" et la position du domaine de premier niveau (TLD). Certaines

Caractéristiques structurelles peuvent nécessiter l'utilisation de services externes pour leur estimation.

* Caractéristiques statistiques (IU2) : Ces caractéristiques sont concernées par le nombre ou la distribution des éléments de base de l'URL, des mots spécifiques ou des caractères dans le texte des URL. Des exemples de ces caractéristiques sont le nombre de points et de sous-domaines et la longueur des mots. L'ensemble des caractéristiques expérimentées de cette classe est donné dans le tableau 2 ainsi que leur type de valeurs, leurs sous-classes et les intuitions derrière leur adoption. Dans le tableau, IU1\* fait référence aux caractéristiques structurelles qui peuvent nécessiter des services externes pour leur estimation.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Index | Feature | Type | Class | Intuition |
| f1-2 | URL parts  lengths | Int | IU2 | Long URLs are used to hide real domains and  subdomains. We consider: full URL length (f1) and hostname length (f2). |
| f3 | IP | 0/1 | IU1 | IP addresses are used in hostnames to hide the identity of websites. IPs can also be used wi-thout dots or hexadecimal encoded. Presence of IPs in any format in hostnames is consider-ed as phishing indicator. |
| f4-20 | Spécial Caractères | Int | IU2 | Special characters are used to deceive novice  users of real domains and subdomains. We consider the number of occurrences of the fol- lowing characters: ‘.’(f4), '-' (f5), '()' (f6), ‘?'  (f7), '(&)' (f8), '|' (f9), '=' (f10)), '(\_)' (f11),  '~' (f12),'%' (f13), '/' (f14), '\*' (f15), ‘:' (f16),  ',' (f17), ‘;' (f18), '($)' (f19), '%20' or space (f20). |
| f21-24 | (#) common  terms | Int | IU2 | Common terms in URLs such as 'www' (f21),  '.com' (f22), 'http' (f23) and '//' (f24) are used only once in legitimate URLs where it is ob- served that they are used more than once in phishing URLs. |
| f25 | HTTPS to-  ken | 0/1 | IU1 | Most phishing websites do not provide any  security facilities compared with legitimate ones. Thus, the use of HTTPS is a legitimacy indicator. |
| f26-27 | Ratio of di-  gits | Float | IU2 | High number of digits in URLs is considered  as a phishing indicator. We consider ratio of digits in full URLs (f26) and hostnames (f27). |
| f28 | Punycode | 0/1 | IU1 | Punycode is used in domain names to replace  some ASCIIs with Unicode characters. URLs will then look legitimate where they refer to di erent websites. URLs with punycodes are considered phishing. |
| f29 | Port | 0/1 | IU1 | Port numbers are rarely used in legitimate  URLs. Therefore, URLs with port indicator are considered phishing. |
| f30-31 | TLD posi-  tion | 0/1 | IU1 | In well-formed URLs, top-level domains  (TLDs) appear only before the path. When TLDs appear in the path (f30) or in the sub- domain part (f31), the URL is considered phi- shing. |
| f32 | Abnormal  subdomains | 0/1 | IU1 | Phishing URLs may use the following pat-  tern 'w[w] ?[0-9]\*' instead of 'www' to deceive users. Thus, URLs with subdomains matching such pattern are considered phishing. |
| f33 | (#)subdomains | Int | IU2 | Phishing URLs use more number of subdo-  mains compared with legitimate ones. Thus, the number of subdomains is a phishing fea- ture. |
| f34 | Pre x Su x | 0/1 | IU1 | Pre xes and su xes separated by "-" are used  in domain names to make users feel that they are dealing with legitimate pages. Thus, when "-" is found in domain names, the URL is considered phishing. |
| f35 | Random do-  mains | 0/1 | IU1 | Phishing URLs use words formed from ran-  dom characters. Therefore, domain names are checked for randomness. |
| f36 | Shortening  service | 0/1 | IU1 \* | URL shortening service is used to indicate  short URLs that serve as a redirect to other long and complex URLs. This ser- vice can be used in phishing to hide the name of real hosts. Therefore, the use of shortening service is considered as a phi- shing indicator. The list of shortners are extracted from URLTeam tracker available at https. [://www.archiveteam.org/index.](http://www.archiveteam.org/index) php ?title=URLTeamURL\_shorteners. |
| f37 | Path exten-  sion | 0/1 | IU1 \* | Malicious scripts can be added to legitimate  pages. Some le extensions used in URL paths may lunch such kind of attacks. Presence of the following malicious path extensions is considered : 'txt', 'exe', 'js'. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| f38-39 | Redirections | Int | IU2 | URL redirection is a technique used to open  pages with di erent URLs than those initially selected by users. This is useful to prevent ac- cess to broken links when web pages are mo- ved. URLs can be redirected to pages with the same domain (i.e. internal redirection) or to pages from di erent domains (i.e., external redirections). However, redirection can also be used for hostile purposes. The number of redi- rections (f38) and external redirections (f39) are considered as phishing indicators. |
| f40-50 | NLP fea-  tures | Int | IU2 | Natural language processing and word-raw  features are also used in phishing detection. We consider number of words (f40), char re- peat (f41), shortest words in URLs (f42), host- names (f43), and paths (f44), longest words in URLs (f45), hostnames (f46), and paths (f47), average length of words in URLs (f48), host- names (f49), and paths (f50). |
| f51 | Phish hints | Int | IU2 | Phishing URLs use sensitive words to gain  trust on visited web pages. The number of such words in URLs is considered as phishing indicator. |
| f52-54 | Brand do-  mains | 0/1 | IU1 \* | Phishing URLs use brand domain names in  di erent URL parts. The presence of brand names in the domain part is considered as a legitimacy indicator (f52) where their pre- sence in subdomains (f53) or paths (f54) is considered as a phishing indicator. The list of brand domain names are collected from https :  [//www.101domain.com/.](http://www.101domain.com/) |
| f55 | Suspicious  TLD | 0/1 | IU1 \* | TLDs are checked for suspiciousness.  List of suspicious TLDs, used in this study, are collected from : Spamhaus.org (https [://www.spamhaus.org)](http://www.spamhaus.org/) and Blue coast system Inc. (https [://www.broadcom.com/).](http://www.broadcom.com/)) |
| f56 | Statistical  report | 0/1 | IU1 \* | URL domains are checked if their IP addresses  match one of top phishing domains. The list is collected from previous works. |

Table 3 Caractéristiques basées sur l'URL pour la détection de l'hameçonnage de sites web

#### 3.3.2.2. Fonctionnalités basées sur le contenu

Les caractéristiques basées sur le contenu sont extraites en chargeant les pages Web des URL et en analysant leur contenu HTML. Elles peuvent être divisées en caractéristiques basées sur les hyperliens et caractéristiques basées sur le contenu anormal.

* Caractéristiques basées sur les hyperliens (IC1) : Ces caractéristiques sont concernées par le nombre, l'état et la nature des hyperliens (c'est-à-dire internes/externes) utilisés dans les balises HTML.
* Caractéristiques basées sur le contenu anormal (IC2) : Ces caractéristiques sont concernées par l'identification de contenus suspects ou de scripts implémentant des comportements suspects. Des exemples de contenus suspects sont l'utilisation de liens vides et de noms de domaine différents dans la balise titre des pages Web. Des exemples de comportements suspects sont l'envoi de contenus de formulaires à des adresses e-mail et la 21 désactivation des clics droits. Hannousse et Yahiouche, fournissent une liste des caractéristiques basées sur le contenu expérimenté dans leur étude, ainsi que leur type de valeurs, leurs classes et les intuitions derrière leur adoption.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Index | Feature | Type | Class | Intuition |
| f57 | (#)hyperlinks | Int | IC1 | Legitimate websites are supposed to consist of  bigger number of pages compared with phi- shing ones. Therefore, the number of links in URL web page contents is considered for dis- tinguishing phishing websites. |
| f58-59 | Ratio inter-  nal/external hyperlinks | Float | IC1 | Legitimate pages usually use hyperlinks with  the same base domain of the website while phishing pages use more external hyperlinks pointing to target websites. The ratio of inter- nal (f58) and external (f59) hyperlinks of web pages are considered as phishing indicators. |
| f60 | Ratio null  hyperlinks | Float | IC1 | To mimic target pages, the same hyperlinks of  legitimate web pages appear in phishing web pages but with empty links. Therefore the ra- tio of null hyperlinks in tags is used as a phi- shing indicator. |
| f61 | # External  CSS | Int | IC1 | Legitimate websites use an internal style or  more than one CSS le. Instead, phishing web- sites use only a sole external CSS le that contains links to CSS les of target websites. Consequently, the number of external CSS  les is considered as a phishing indicator. |
| f62-63 | #Internal/  External redirections | Int | IC1 | Links in phishing web pages may redirect to  other legitimate or fake pages. The ratio of internal (f62) and external (63) redirections are proposed for distinguishing phishing web pages. |
| f64-65 | Ratio In-  ternal/ External errors | Float | IC1 | Fake hyperlinks are usually present in phi-  shing web pages. Therefore, all hyperlinks of web pages are checked and the ratio of internal (f64) and external(f65) hyperlinks connection errors are counted. |
| f66 | Login forms | 0/1 | IC2 | Login forms are another means commonly  used for stealing information of web users. Lo- gin forms with external action links or empty actions are considered phishing. Empty action formats considered in this study are : "", "# ", " # nothing", "# doesnotexist","# null", "# void", "# whatever", "# content", "ja-  vascript : :void(0)", "javascript : :void(0) ;", "javascript : : ;", "javascript". |
| f67 | External fa-  vicon | 0/1 | IC1 | To mimic legitimate websites, phishers use  the same favicon icon of the target website in the address bar of navigators. Therefore, websites using external favicons are conside- red phishing. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| f68 | Links in tags | Float | IC1 | In legitimate websites it is expected that  <Link> tags use links pointing to web pages of the same domain as the URL. Therefore, the ratio of internal links in <Link> tags is considered for phishing detection. |
| f69 | Submit to  Email | 0/1 | IC2 | Phishers submit user inputs in web forms into  speci c email addresses. Form actions contai- ning 'mailto :' or 'mail()' are considered phi- shing. |
| f70-71 | Ratio inter-  nal/external media | Int | IC1 | Legitimate websites mostly use media  (images, audio, video) stored in the same domain. Phishing websites use more external media, usually stored in the target website domain, to save the storage space. Ratios of internal (f70) and external(f71) media le links are counted and used for distinguishing legitimate form phishing websites. |
| f72 | SFH | 0/1 | IC2 | Normally, actions should be taken upon  submitted information on web page forms. Therefore, Forms with an empty string or 'about :blank' are considered parts of phishing web pages. |
| f73 | Invisible  iframe | 0/1 | IC2 | Frame tags are used to incorporate additio-  nal web pages to those actually shown. Phi- shing websites may use <iframe> with invi- sible border so that users may think that addi- tional pages are part of current websites while they are actually from di erent domains. The- refore, the use of invisible <iframe> tags is considered as a phishing indicator. |
| f74 | Pop-up win-  dow | 0/1 | IC2 | Pop-up windows are used by legitimate web-  sites to alert users with warnings but rarely used to submit user information. The presence of pop-up windows with text elds is conside- red as a phishing indicator. |
| f75 | Safe anchor | Int | IC2 | The <a> tag is used to enable linking from  one page to another. Tags with one of the follo- wing links '# ', 'javascript', 'mailto' are consi- dered unsafe. Thus, we consider the number of unsafe anchors. |
| f76-77 | Right-click | 0/1 | IC2 | Scripts can be used to disable the right-  click function. However, this can also be used by phishers to unable viewing the source code of web pages. Therefore, the presence of onmouseover' attribute (f76) and use of 'event.button==2' as an action to 'onmouseo- ver' attribute (f77) is considered as a phishing indicator. |
| f78 | Empty title | 0/1 | IC2 | Most legitimate websites describe the title of  web pages in the <title> tag. The absence of web page title is considered as phishing indi- cator. |
| f79 | Domain in  title | 0/1 | IC2 | Legitimate websites often use the domain  name as part of the title of web pages. Phi- shing websites use legitimate domains in titles to deceive users. Therefore, the presence of the domain of URL as part of the web page title is considered as legitimacy indicator. |
| f80 | Domain  within copy- right | 0/1 | IC2 | Legitimate websites indicate their domain  name within the copyright logo. Phishing web- sites do not use their actual domain. Presence of the domain of URLs within the copyright logo is a legitimacy indicator. |

Table 4 Caractéristiques basées sur le contenu pour la détection des sites web d'hameçonnage

#### 3.3.2.3. Caractéristiques externes (External-based features)

La dernière étape de la section décrit les caractéristiques externes, qui sont obtenues en interrogeant des services tiers de référence et des moteurs de recherche. Des exemples de ces services tiers sont WHOIS, Alexa, Open PageRank et Google. La liste des caractéristiques externes expérimentées par les auteurs dans leur étude est donnée dans le tableau 5 :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Index | Feature | Type | Class | Intuition |
| f81 | WHOIS re-  gistered do- main | 0/1 | E | Domains of phishing websites do not match  any WHOIS database record contrary to most legitimate domains. Therefore, URLs with do- mains not registered in WHOIS are considered phishing. |
| f82 | Domain  registration length | Int | E | Phishing websites live for a short period of  time, while legitimate websites are regularly paid for several years in advance. Instead of proposing a speci c threshold as proposed in [20, 18, 27], we use the number of years the do- main renewal amount was paid as a phishing indicator |
| f83 | Domain age | Int | E | Since phishing websites are short lived, the age  of URL domains is considered as a phishing indicator. |
| f84 | Web tra c | Int | E | Phishing websites generally have less number  of visitors compared with legitimate websites. Alexa is used to identify the web tra c of |
| f85 | DNS record | 0/1 | E | Domain Name Server (DNS) is mandatory to  retrieve the IP address of URLs for access. Therefore, URL domains must be registered within the DNS. A missing DNS recored is a phishing indicator. |
| f86 | Google index | 0/1 | E | Phishing websites live for short times and are  often accessible through direct links sent to users in emails, they do not need to be indexed by Google. Web pages not indexed by Google are supposed phishing. |
| f87 | Page rank | Int | E | Phishing web pages are not very popular,  hence, they supposed to have low page ranks compared with legitimate web pages. We use Open PageRank to get the value of this feature. |

Table 5 Caractéristiques externes pour la détection des sites web d'hameçonnage

* WHOIS9 : Ce service fournit des informations sur les noms de domaine, tels que le propriétaire, la date d'enregistrement et la date d'expiration.
* Alexa : Ce service fournit des statistiques de trafic sur le site Web, telles que le nombre de visiteurs et de pages vues.
* Openpagerank10 : Ce service fournit un classement du site Web en fonction du nombre de back links qu'il a. Google11 : Il s'agit d'utiliser les requêtes et les résultats des moteurs de recherche Google pour extraire des informations sur le site Web.
* Les auteurs fournissent des intuitions derrière l'adoption de chaque caractéristique. Ces intuitions sont basées sur la compréhension de la façon dont les pirates utilisent ces caractéristiques pour créer des sites de phishing. Par exemple, les pirates utilisent souvent des domaines nouveaux ou des domaines avec un faible classement pour créer des sites de phishing.

### 3.3.3. Génération des ensembles de données

Cette section décrit le processus de génération du jeu de données utilisé par les auteurs pour entraîner et évaluer leurs modèles de détection de phishing. Le processus se déroule en trois étapes à savoir :

Extraction des caractéristiques : Les auteurs extraient les caractéristiques de chaque URL en utilisant les informations stockées dans un ensemble de données indexé par le champ URL (ensemble de données A dans la figure 7). L'ensemble de données A est stocké au format pickle, où chaque ligne contient une URL, sa source (Yandex, Alexa, Phishtank, Openphish) et son arbre DOM généré.

Stockage des vecteurs de caractéristiques : Les vecteurs de caractéristiques générés sont stockés dans un ensemble de données séparé indexé par les URLs au format CSV.

Équilibrage et préparation du jeu de données : Ensuite, l'ensemble de données B est équilibré avec 50 pourcents pour chaque classe (légitime, phishing). Enfin, les lignes sont mélangées et le jeu de données est prêt à être utilisé.

## 3.4. METHODOLOGIE DE RECHERCHE

Hannousse et Yahiouche expliquent ici la méthodologie et les objectifs des expériences qu'ils ont menées.

Ont mené cinq expériences sur l'ensemble de données collectées.

Les deux premières expériences visaient à identifier le ou les meilleurs classificateurs pour la détection de phishing sur les sites web.

Les deux expériences suivantes se concentraient sur le rôle des méthodes de sélection de caractéristiques dans l'amélioration des performances des classificateurs.

Enfin, la dernière expérience visait à identifier les caractéristiques ou classes de caractéristiques adaptées à la détection instantanée de phishing.

Les quatre premières expériences ont été menées dans la plateforme Weka sans aucune modification du développement central de l'outil.

Pour l'évaluation des Performances Hannousse et Yahiouche ils utilisent deux métriques principales pour évaluer les performances.

La première métrique est l'Exactitude (Accuracy), qui représente le ratio d'échantillons correctement prédits par rapport au nombre total d'échantillons.

Cette métrique est adaptée aux ensembles de données équilibrés. La précision d'un modèle est calculée à l'aide de la formule suivante :

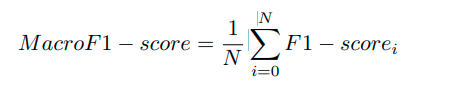
*TP* + *TN*

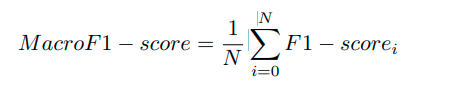
*Accuracy* =

*TP* + *TN* + *Fp*+*FN*

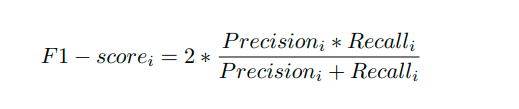
True Positive (TP) désigne le nombre de prédictions correctes de pages web de phishing, tandis que True Négative (TN) désigne le nombre de prédictions correctes de pages web légitimes. De même, False Positive (FP) désigne le nombre de prédictions incorrectes de pages web de phishing et False Négative (FN) désigne le nombre de prédictions incorrectes de pages web légitimes.

* La deuxième métrique est le Macro F1-score, qui représente la moyenne des F1-scores par classe. Le Macro F1-score est obtenu en faisant la moyenne des F1-scores calculés pour chaque classe i. Le score F1 macro est calculé selon la formule suivante :

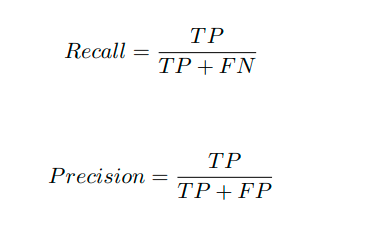




Le F1-score pour chaque classe représente le meilleur compromis entre la précision et le rappel de la classe. Il est calculé à l'aide de la formule suivante :



Le rappel et la précision sont calculés en termes de TP, FP et FN comme suit :



### **3.4.1. Classifier évaluation**

Cette section explique qu'ils se concentrent sur l'évaluation des classificateurs pour la détection de phishing sur les sites web. Ils mentionnent avoir basé leur choix de classificateurs sur le classement élaboré par Das et al. [6], lequel sont établi en fonction de la fréquence d'utilisation dans des études antérieures. Les cinq meilleurs classificateurs identifiés dans l'étude de Das et al. [6] sont testés et comparés sur l'ensemble de données collecté. Cette évaluation prend en compte deux aspects : (1) les différentes classes de caractéristiques et (2) l'effet de la combinaison des meilleurs modèles pour chaque classe.

#### 3.4.1.1. Expérience I

Son objectif est d'identifier les meilleurs classificateurs pour chaque classe individuelle de caractéristiques, ainsi que la meilleure combinaison de classes de caractéristiques. Pour ce faire, les auteurs ont appliqué plusieurs classificateurs individuellement à différentes classes de caractéristiques. La figure représente le processus d'identification des meilleurs classificateurs et de la meilleure combinaison de classes d'entités pour une tâche de classification. Le processus est divisé en quatre étapes principales. Les performances de chaque classificateur ont été enregistrées et comparées aux performances des autres classificateurs. Les classificateurs ayant les meilleures performances pour chaque classe de caractéristiques ont été sélectionnés comme candidats pour l'expérience 2.

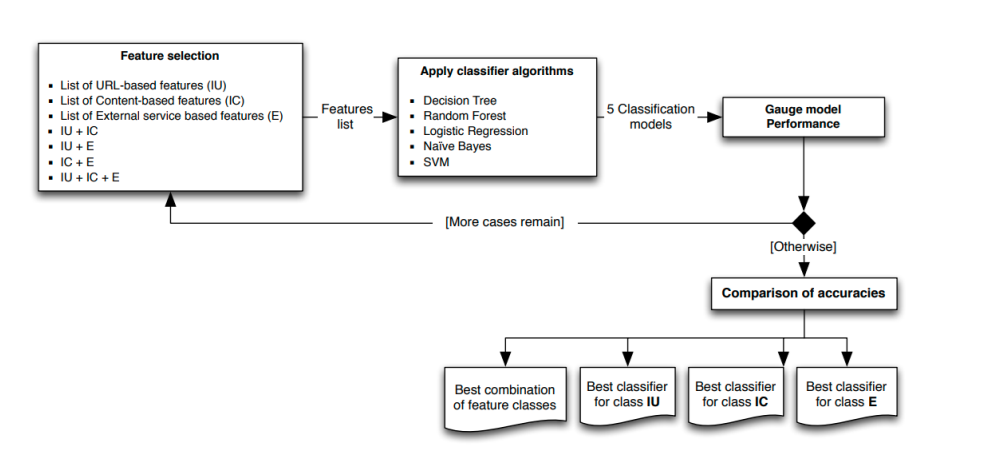


Figure 9 Processus d'identification des meilleurs classificateurs et de la meilleure combinaison de classes d'entités.

#### 3.4.1.2. Expérience II

Ici, les meilleurs classificateurs de chaque classe individuelle, identifiés dans l'expérience I, sont utilisés pour évaluer les améliorations potentielles de performances résultant de leurs combinaisons. Quatre types de combinaisons sont examinés : Combinaison AND, Combinaison OR, Empilement et Vote majoritaire. La figure illustre le processus global adopté pour cette expérience. Cette étape vise à explorer comment la combinaison de différents classificateurs peut potentiellement améliorer la détection de phishing sur les sites web, élargissant ainsi les perspectives après l'identification des meilleurs classificateurs individuels dans l'expérience I.

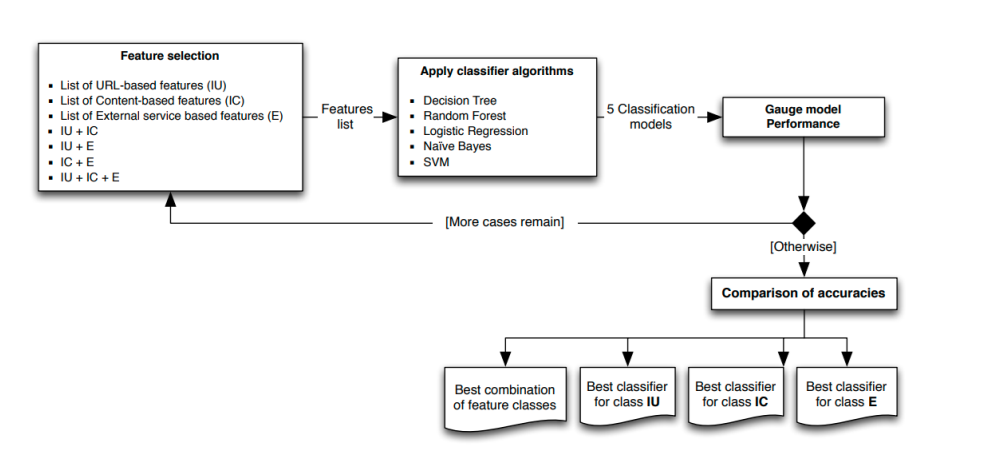


Figure 10 Combinaison de modèles formés sur différentes classes de fonctionnalités.

### **3.4.2. Évaluation des méthodes de sélection des caractéristiques**

Les expériences de cette partie concernent la sélection des meilleurs sous-ensembles de caractéristiques pour améliorer la précision des modèles.

Pour ce faire, nous examinons la capacité des techniques existantes à sélectionner les caractéristiques discriminantes, à savoir les méthodes de filtrage et les méthodes d'enveloppement.

Les méthodes de filtrage et d'enveloppement. L'objectif des expériences est de vérifier la validité de l'hypothèse largement acceptée selon laquelle

Les caractéristiques les mieux classées offrent une plus grande précision de classification.

#### 3.4.2.1. Expérience III

Cette section évalue comment différentes méthodes de filtre peuvent être utilisées pour sélectionner les meilleures caractéristiques an d'améliorer la précision des modèles de détection de phishing. Les filtres sont évalués en utilisant des classificateurs déjà déterminés comme référence dans l'expérience II. L'objectif de cette partie est d'évaluer la capacité des méthodes existantes à sélectionner des sous-ensembles de caractéristiques qui améliorent la précision des modèles. Deux types de méthodes de sélection de caractéristiques sont évalués : les méthodes de filtre (filtre Methods) et les méthodes d'emballage (wrapper Methods). Les méthodes de filtre sélectionnent des caractéristiques indépendamment de tout algorithme d'apprentissage automatique. Elles choisissent des caractéristiques en fonction de scores associés à celles-ci basés sur leur corrélation avec les valeurs de l'attribut de classe. Quatre filtres de classement sont testés : chi-square, Pearson corrélation, information gain and relief [6].

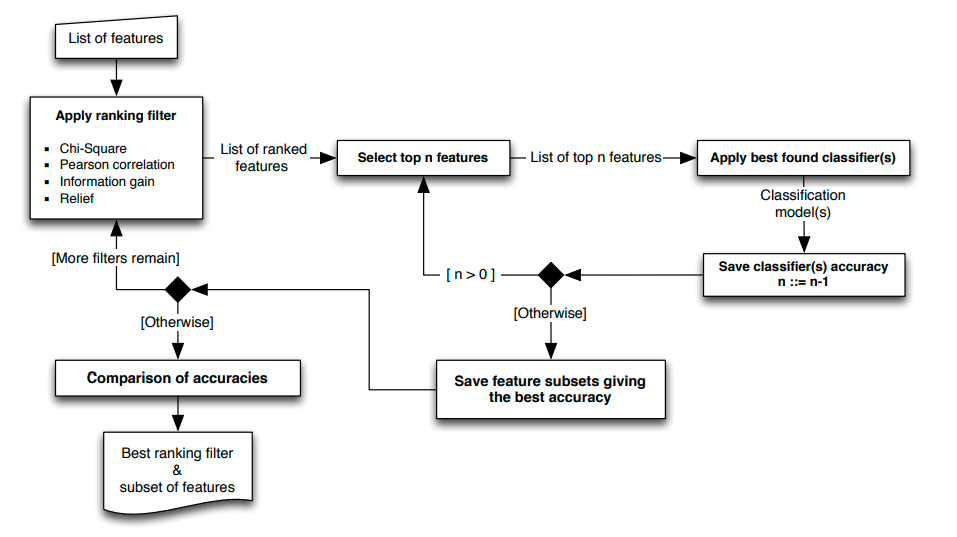


Figure 11 Processus d'identification des meilleurs filtres et des sous-ensembles de caractéristiques.

#### 3.4.2.1. Expérience IV

L'objectif de cette expérience est d'évaluer l'efficacité des méthodes d'emballage pour la sélection de caractéristiques. Les méthodes d'emballage sélectionnent des caractéristiques en utilisant un algorithme d'apprentissage automatique spécifique. Dans cette expérience, deux évaluateurs, ClassifierSubsetEval et WrapperSubsetEval, sont examinés.

* ClassifierSubsetEval : Cet évaluateur commence par alimenter le classificateur sélectionné avec un sous ensemble de caractéristiques. En fonction de la précision obtenue et des déductions tirées des calculs précédents, l'algorithme décide d'ajouter ou de supprimer des caractéristiques du sous-ensemble jusqu'à atteindre la meilleure précision.
* WrapperSubsetEval : Il suit une approche similaire à ClassifierSubsetEval mais utilise un processus d'évaluation différent.
* La recherche Best-First est adoptée comme méthode de recherche, car elle a été identifiée comme la méthode la plus performante dans une référence spécifiée ([7]). La figure 12 décrit le processus adopté pour cette expérience.

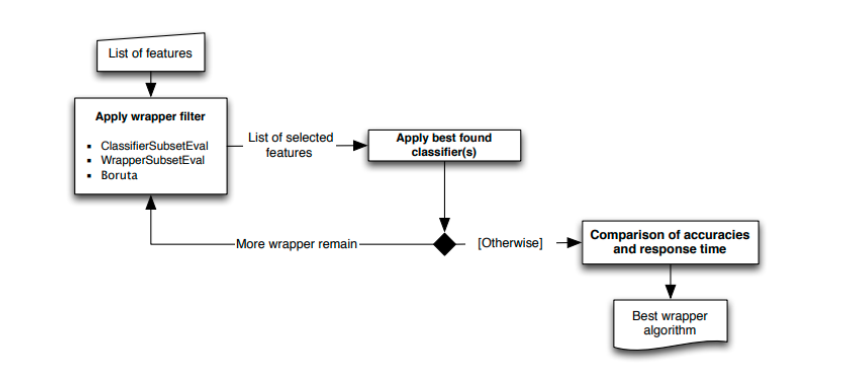


Figure 12 Processus d'identification pour le meilleur algorithme d'encapsulation pour la sélection des fonctionnalités.

Cette section évalue comment différentes méthodes d'emballage peuvent être utilisées pour sélectionner les caractéristiques optimales en fonction d'un algorithme de machine Learning choisi. Les évaluateurs ClassifierSubsetEval et WrapperSubsetEval sont examinés, et la recherche Best-First est utilisée pour guider le processus de sélection. De plus, une implémentation en Python de l'emballage Boruta est utilisée pour évaluer son efficacité dans la sélection des meilleures caractéristiques

### 3.4.2. Expérience V : Analyse de la durée d'exécution de l'extraction de caractéristiques

L'objectif est d'analyser le temps requis pour construire les vecteurs de caractéristiques utilisés par les classificateurs d'apprentissage automatique. L'analyse commence par évaluer la performance de chaque classe individuellement. Si des retards notables sont détectés, l'analyse se concentre sur les caractéristiques individuelles de chaque classe pour identifier celles qui pourraient causer ces retards. Le temps d'extraction moyen de tous les 11430 échantillons de l'ensemble de données collecté est estimé dans le cadre de cette expérience. Étant donné que le temps d'extraction peut varier entre les pages web légitimes et de phishing en raison de leurs tailles de contenu différentes, l'utilisation de toutes les instances du jeu de données est justifiée en raison de la nature équilibrée de l'ensemble de données. On peut résumer que cette expérience vise à comprendre le temps nécessaire pour extraire les caractéristiques des données, en se concentrant sur l'analyse des performances individuelles des classes. L'estimation du temps d'extraction moyen de l'ensemble du jeu de données contribue à identifier les caractéristiques qui pourraient entraîner des retards, ce qui est crucial pour la détection en temps réel de pages web potentiellement frauduleuses.

## 3.5. ANALYSE DES RÉSULTATS EXPÉRIMENTAUX

Hannousse.A et Yahiouche.S, dans cette section présentent les détails des différentes expériences réalisées et les résultats obtenus. Cinq expériences au total ont été menées, avec les quatre premières évaluant la performance des classificateurs sur différentes classes de caractéristiques, et la dernière estimant le temps d'extraction des différentes fonctionnalités.

### **3.5.1. Expérience I - Performance des Classificateurs sur Différentes Classes de Caractéristiques**

Cinq classificateurs fréquemment utilisés, à savoir ((Décision Tree, Random Forest, Logistic Regression, Naive Bayes, SVM)), sont rigoureusement évalués sur différentes classes de fonctionnalités, notamment les URL, le contenu et les caractéristiques externes. La figure 13 montre les résultats expérimentaux des classificateurs examinés sur l'URL, le contenu et les fonctionnalités externes.

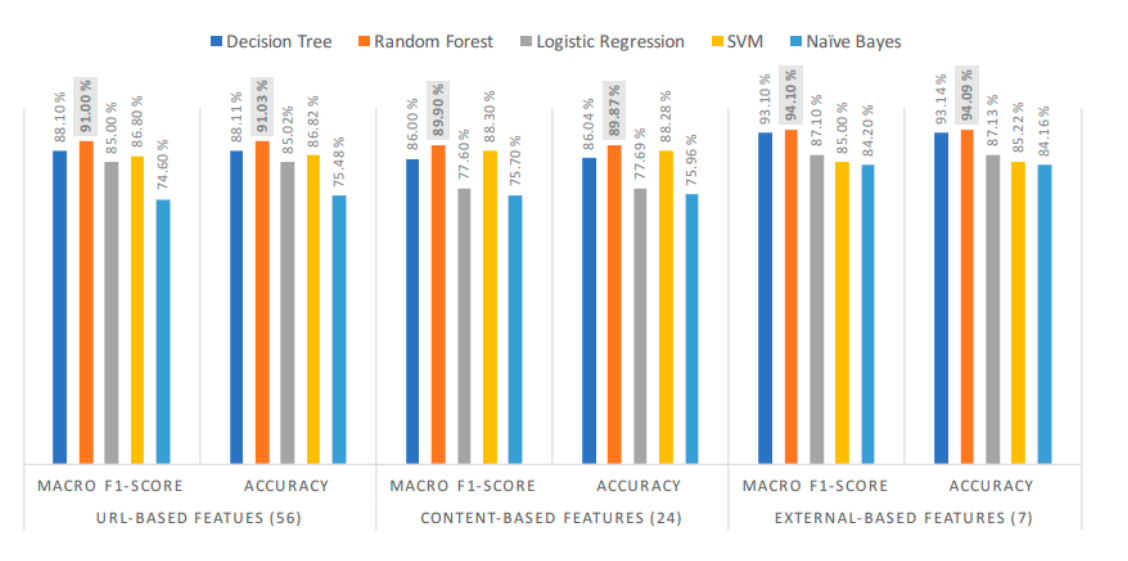


Figure 13 Performance des classificateurs formés sur la classe individuelle des caractéristiques.

Les résultats révèlent que Random Forest surpasse tous les autres classificateurs, démontrant une précision et le Macro F1-score supérieurs sur l'ensemble des classes de fonctionnalités. La figure 14 montre les résultats de l'expérience. Les résultats confirment à nouveau que Random Forest surpasse tous les autres classificateurs et fournit des scores de précision plus élevés.

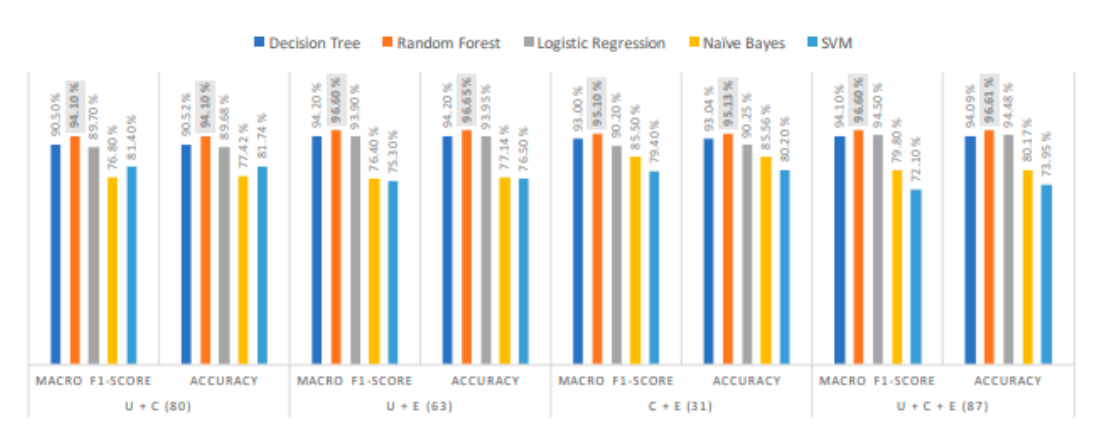


Figure 14 Performance des classificateurs formés sur des classes de fonctionnalités combinées par paires.

Notamment, les fonctionnalités externes affichent la meilleure précision à 94,09 pourcents, suivies des URL à 91,03 pourcents et du contenu à 89,87 pourcents. Ces constatations contredisent les conclusions antérieures de Choo et al. [8], soulignant une divergence potentielle due à la limitation de l'utilisation des fonctionnalités externes dans leur étude. La combinaison des URL et des fonctionnalités externes atteint la plus grande précision à 96,60 pourcents, mettant en évidence l'insuffisance des classes de fonctionnalités individuelles pour assurer une détection robuste et soulignant l'importance de l'approche hybride pour améliorer la performance des classificateurs

### **3.5.2. Expérience II. Performance des modèles combinés formés chacun sur une classe différente de caractéristiques**

L'expérience II explore la performance de modèles combinés, chacun étant formé sur une classe de fonctionnalités différente. En suivant le processus indiqué dans la Figure 10, trois modèles Random Forest sont examinés, chacun étant formé sur une classe de fonctionnalités distincte dans le but d'améliorer la précision des prédictions. Des combinaisons telles qu’AND, OR, stacking et majority voting sont étudiées. Dans chaque cas, la prédiction finale pour les instances de test dépend de la prédiction des trois modèles de base. Les résultats présentés dans la Figure 15 révèlent qu’aucune des combinaisons n'améliore la précision.

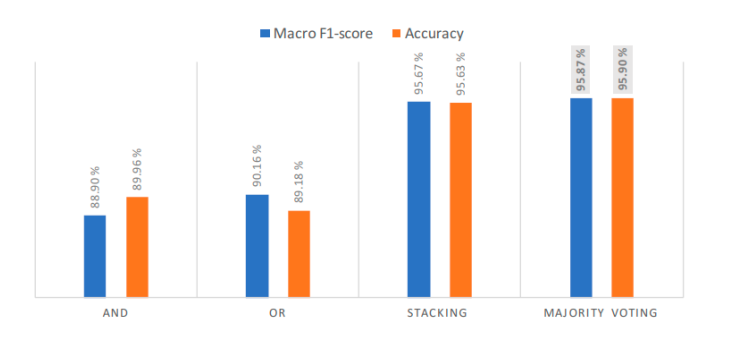


Figure 15 Performance des modèles combinés.

Cependant, le stacking et le majority voting fournissent des scores approximatifs à ceux obtenus en utilisant un seul modèle formé sur du contenu assemblé avec des fonctionnalités basées sur des éléments externes.

### **3.5.3. Expérience III. Performance par formation sur les fonctionnalités sélectionnées à l'aide de méthodes de classement par filtre**

Dans cette expérience, ils évaluent la précision du meilleur classificateur trouvé (c'est-à-dire Random Forest) sur la sélection décrémentale de l'ensemble des caractéristiques. Les caractéristiques sont d'abord classées à l'aide de filtres de chi-carré, de corrélation de Pearson, de gain d'information et de relief. Ensuite, les performances du classificateur Random Forest sont évaluées sur les caractéristiques sélectionnées, les caractéristiques les moins bien classées étant éliminées une à une à chaque étape. Le tableau 6 présente la liste des 25 caractéristiques les mieux classées à l'aide des différents filtres examinés en ce qui concerne la classe des URL de pages web. D'après le tableau, google index (f86) est la première caractéristique identifiée par tous les filtres et 7 caractéristiques (f86, f87, f21, f26, f79, f83, f58) figurent parmi les 25 premières caractéristiques de tous les filtres. En examinant uniquement ces 7 caractéristiques, ils ont atteint une précision maximale de 94,67 % avec Random Forest. En outre, les 25 premières caractéristiques de tous les filtres proviennent de trois classes différentes, ce qui justifie la nécessité d'utiliser des caractéristiques hybrides pour l'identification des pages web de phishing. Plus précisément, IU2 s'avère être la meilleure classe où la présence des caractéristiques incorporées dans les 25 premières listes varie de 20 % à 60 %. À l'inverse, la classe IC2 s'avère être la pire puisque la présence de ses caractéristiques dans les 25 premières listes ne varie que de 8 % à 12 %.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Pearson correlation | | | Information gain | | Relief | | Chi-Square | |
| Order | Feature | Class | Feature | Class | Feature | Class | Feature | Class |
| 1 | f86 | E | f86 | E | f86 | E | f86 | E |
| 2 | f87 | E | f84 | E | f87 | E | f84 | E |
| 3 | f21 | IU2 | f57 | IC1 | f21 | IU2 | f57 | IC1 |
| 4 | f26 | IU2 | f87 | E | f83 | E | f87 | E |
| 5 | f79 | IC2 | f83 | E | f3 | IU1 | f83 | E |
| 6 | f57 | IC1 | f59 | IC1 | f75 | IC2 | f59 | IC1 |
| 7 | f51 | IU2 | f58 | IC1 | f33 | IU2 | f58 | IC1 |
| 8 | f83 | E | f75 | IC2 | f79 | IC2 | f75 | IC2 |
| 9 | f3 | IU1 | f82 | E | f71 | IC1 | f21 | IU2 |
| 10 | f7 | IU2 | f47 | IU2 | f80 | IC2 | f63 | IC1 |
| 11 | f1 | IU2 | f21 | IU2 | f68 | IC1 | f82 | E |
| 12 | f58 | IC1 | f63 | IC1 | f25 | IU1 | f47 | IU2 |
| 13 | f14 | IU2 | f68 | IC1 | f59 | IC1 | f68 | IC1 |
| 14 | f2 | IU2 | f26 | IU2 | f70 | IC1 | f26 | IU2 |
| 15 | f10 | IU2 | f51 | IU2 | f58 | IC1 | f41 | IU2 |
| 16 | f27 | IU2 | f41 | IU2 | f31 | IU1 | f43 | IU2 |
| 17 | f43 | IU2 | f45 | IU2 | f34 | IU1 | f51 | IU2 |
| 18 | f34 | IU1 | f43 | IU2 | f52 | IU1 \* | f45 | IU2 |
| 19 | f47 | IU2 | f50 | IU2 | f67 | IC1 | f50 | IU2 |
| 20 | f31 | IU1 | f1 | IU2 | f38 | IU2 | f65 | IC1 |
| 21 | f78 | IC2 | f49 | IU2 | f26 | IU2 | f79 | IC2 |
| 22 | f4 | IU2 | f2 | IU1 | f36 | IU1 \* | f1 | IU2 |
| 23 | f45 | IU2 | f65 | IC1 | f84 | E | f70 | IC1 |
| 24 | f50 | IU2 | f4 | IU1 | f7 | IU2 | f4 | IU2 |
| 25 | f49 | IU2 | f79 | IC2 | f63 | IC1 | f49 | IU2 |

Tableau 6 Les 25 caractéristiques les mieux classées pour chaque filtre utilisé

La figure 16 montre les graphiques de précision concernant les caractéristiques sélectionnées sur la base des résultats du classement des différents filtres. Les caractéristiques sont sélectionnées de manière à ce que les caractéristiques les moins bien classées soient éliminées en premier et une par une à chaque étape. La première valeur de la figure 16 désigne la précision du modèle lorsque seule la caractéristique la mieux classée est sélectionnée, tandis que la dernière valeur désigne la précision lorsque toutes les 87 caractéristiques sont sélectionnées. Les résultats montrent que le taux de précision maximal de 96,83 % est atteint en sélectionnant 73 caractéristiques à l'aide du filtre chi-carré. Cette valeur est légèrement réduite par l'ajout 33 de caractéristiques de rang inférieur.

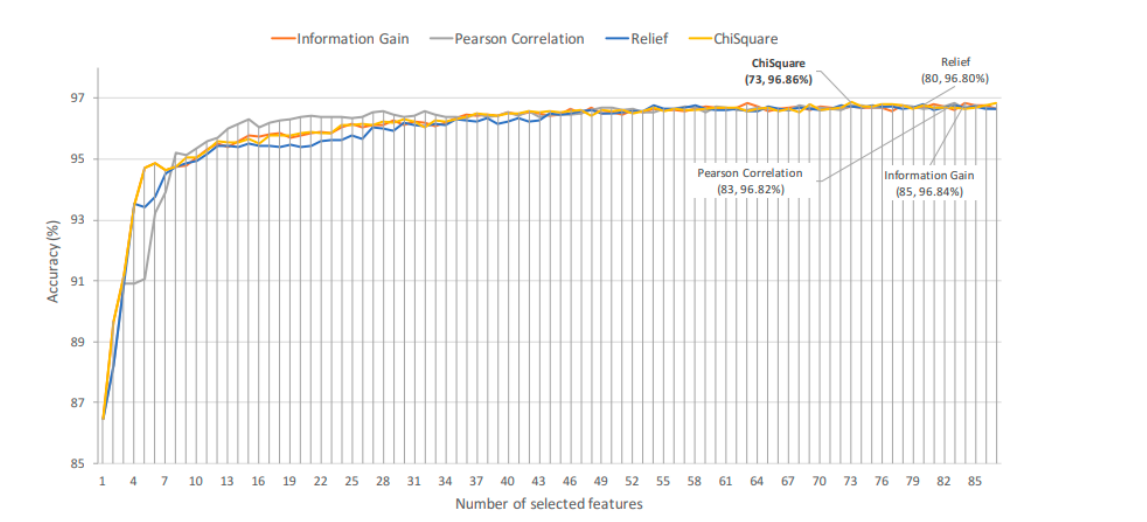


Figure 16 Précision du modèle de prédiction aléatoire des forêts par rapport à la sélection par étapes des caractéristiques.

Ces résultats permettent de conclure que l'utilisation de filtres de classement améliore la précision de Random Forst. Plus précisément, le filtre chi-carré donne une meilleure précision que les autres filtres avec un nombre inférieur de caractéristiques. Le tableau 7 présente l'ensemble des caractéristiques les moins importantes détectées par chaque filtre. Les résultats montrent que les caractéristiques basées sur le contenu sont moins importantes puisque 37,50 % de ces caractéristiques ont été détectées comme moins importantes par trois filtres ayant une seule caractéristique commune (c.-à-d. f2.10). En outre, toutes les caractéristiques basées sur les services externes sont jugées importantes, ce qui indique l'efficacité de cette classe de caractéristiques pour distinguer les pages web de phishing

|  |  |  |  |
| --- | --- | --- | --- |
| Filter | IU | IC | E |
| Pearson Correla-  tion | f10 | f60, f64, f69 | None |
| Information gain | f10, f29 | f60 | None |
| Relief | f19, f18, f28, f10 | f69, f72, f62 | None |
| ChiSquare | f10, f20, f35, f29, f37 | f69, f62, f72, f64, f77, f60, f73, f66, f76 | None |
| Total (De-  duplicated) | 8 | 9 | 0 |
| Ratio | 14.29% | 37.50% | 0% |

Tableau 7 Liste des caractéristiques moins importantes identifiées par les filtres de classement utilisés

### **3.5.4. Expérience IV. Performance par la formation sur les fonctionnalités sélectionnées à l'aide des méthodes d'emballage**

Ici les auteurs présentent la quatrième expérience menée, axée sur l'impact des méthodes d'enveloppe (wrapper Methods) sur la sélection des caractéristiques. Les méthodes d'enveloppe sont connues pour leur coût en temps, car elles impliquent l'entraînement d'un modèle sélectionné en utilisant différents sous-ensembles de caractéristiques. Cependant, elles sont reconnues comme étant plus efficaces dans la sélection de caractéristiques discriminatives et pour rendre les modèles plus susceptibles de surajustement (overtting). Les 34 auteurs ont choisi d'évaluer l'efficacité des méthodes d'enveloppe pour améliorer la précision du classifieur Random Forest par rapport au jeu de données collecté. Deux évaluateurs d'enveloppe fournis par la plateforme Weka sont testés : ClassifierSubsetEval et WrapperSubsetEval. Le premier évaluateur évalue la valeur des sous-ensembles de caractéristiques en fonction de leur cohérence avec les valeurs de classe lorsque les instances d'entraînement sont projetées sur le sous-ensemble de caractéristiques. Le second utilise la technique de validation croisée pour estimer la précision du classifieur pour chaque sous-ensemble de caractéristiques. De plus, les auteurs examinent l'algorithme d'enveloppe Boruta [9], qui identifie l'importance des caractéristiques en créant des caractéristiques fantômes. Contrairement aux évaluateurs précédents, l'algorithme Boruta se concentre sur l'identification de toutes les caractéristiques ayant un impact sur la prédiction du classifieur, plutôt que sur la recherche de sous-ensembles causant des erreurs minimales sur les performances du classifieur. Le tableau 8 compare les résultats obtenus des trois évaluateurs de wrapper indiquant la liste des caractéristiques sélectionnées et leurs performances

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Evaluation | Selected | Features | Macro F1 score | Accuracy |
| ClassifierSubsetEval | 11 | f49, f68, f71, f75, f81, f82, | 94.80% | 94.84% |
|  |  | f83, f84, f85, f86, f87 |  |  |
| WrapperSubsetEval | 81 | All except f25, f43, f65, | 96.70% | 96.69% |
|  |  | f67, f71, f76 |  |  |
| Boruta | 61 | All except f9, f12, f15-17, | 96.60% | 96.64% |
|  |  | f19, f23, f28-30, f35, f37, |  |  |
|  |  | f39, f53-54, f60, f62, f64, |  |  |
|  |  | f66, f69, f72-74, f76-77, f81 |  |  |

Table 8 Résultats des évaluateurs du wrapper.

Les auteurs expriment que les résultats indiquent que ClassifierSubsetEval n'a pas réussi à identifier des fonctionnalités susceptibles d'améliorer les performances du classifeur Random Forest. En revanche, WrapperSubsetEval et Boruta ont présenté des résultats meilleurs et similaires, bien que Boruta soit significativement plus rapide que WrapperSubsetEval. Le temps requis par Boruta pour produire des résultats était de seulement 122 secondes, tandis que WrapperSubsetEval a pris environ 24 heures. Cependant, malgré ces performances prometteuses, les trois évaluateurs ont échoué à améliorer la précision du classifeur Random Forest par rapport à la méthode de sélection décroissante de caractéristiques basée sur le classement des filtres utilisée dans l'Expérience III. Ce constat souligne un aspect critique où les méthodes wrapper n'ont pas surpassé les méthodes de filtrage précédemment appliquées dans le contexte spécifique de l'amélioration de la précision du modèle Random Forest.

### **3.5.5. Expérience V. Analyse de l'exécution de l'extraction de caractéristiques**

Dans cette expérience, ils examinent le temps nécessaire à l'extraction des différentes caractéristiques et classes de caractéristiques. L'expérience est réalisée sur un MacBook Pro équipé d'OS X, d'un processeur Intel Core i7 de 2,9 Ghz et de 8 Go de mémoire. La figure 17 montre le temps moyen nécessaire à l'extraction des différentes classes de caractéristiques. Les résultats de l'expérience V montrent que le temps d'extraction des fonctionnalités varie considérablement selon la classe de fonctionnalités. Les fonctionnalités basées sur l'URL (IU1, IU2) et les fonctionnalités basées sur le contenu anormal (IC2) sont les plus rapides à extraire, nécessitant environ 41,5 ms. Les fonctionnalités basées sur les liens hypertextes (IC1) sont plus lentes, en particulier f63 et f65, qui nécessitent de vérifier chaque lien de la page Web. Les fonctionnalités basées sur des services externes (E) nécessitent environ 5 secondes, mais peuvent être réduites à 4 secondes en extrayant f81 et f82 en un seul appel au service WHOIS.

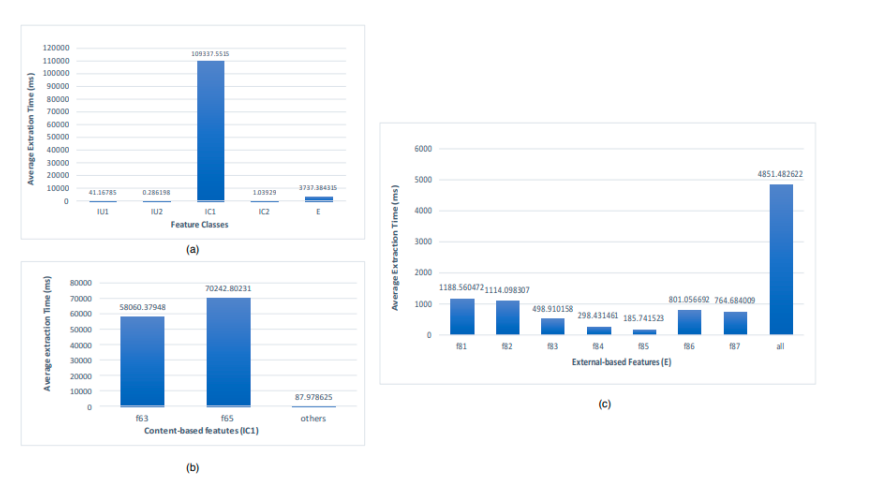


Figure 17 Temps moyen pour l'extraction des caractéristiques

Les résultats de la figure 17 (a) indiquent clairement que les deux sous-classes de caractéristiques basées sur l'URL (IU1, IU2) et les caractéristiques basées sur le contenu anormal (IC2) prennent moins de temps. En particulier, l'extraction de toutes les caractéristiques basées sur l'URL nécessite environ 41,5 ms, ce qui les rend suffisamment adaptées aux systèmes de détection en cours d'exécution. Les caractéristiques basées sur les hyperliens (IC1) sont beaucoup plus lentes que les services externes (E). C'est pourquoi ils ont décidé d'approfondir l'étude et d'inspecter les performances des caractéristiques individuelles de cette classe. La figure 17 (b) montre que f63 et f65 sont à l'origine de ce retard. Ces deux caractéristiques nécessitent de vérifier chaque lien de la page web pour déterminer s'il s'agit d'un faux ou d'un authentique ; le problème avec f63 et f65 devient sérieux lorsque les pages web utilisent un nombre considérable d'hyperliens et donc une grande valeur de f57. La figure 17 (c) montre que l'extraction de toutes les caractéristiques externes (E) nécessite environ 5 secondes. Cette durée peut être réduite à environ 4 secondes car f81 et f82 proviennent du même service (c'est-à-dire le service WHOIS) et peuvent donc être extraites en une seule fois par le biais d'un seul appel au service WHOIS (voir figure 17 (a)). Hannousse.A et Yahiouche.S recommandent de privilégier les fonctionnalités basées sur l'URL pour les systèmes de détection en temps réel. Ils suggèrent d'optimiser l'extraction des fonctionnalités basées sur les liens hypertextes et les services externes pour améliorer les performances. Les auteurs discutent des implifications des résultats pour la conception de systèmes de détection de phishing de sites Web en temps réel, et ils soulignent l'importance de sélectionner des fonctionnalités qui peuvent être extraites rapidement et efficacement.

## 3.6. Rapport d'évaluation et résultats

Les auteurs proposent un schéma général pour la construction de jeux de données reproductibles et extensibles destinés à la détection de phishing sur les sites Web. Ils présentent six directives principales pour la génération de jeux de données de référence visant à assurer l'homogénéité, la cohérence et la diversité des échantillons de données. Les auteurs ont constitué une liste d'URLs selon ces directives et ont développé leurs propres scripts Python pour extraire 87 caractéristiques examinées, qu'ils ont regroupées dans un ensemble de données unique. Ils ont également créé un ensemble de données distinct au format pickle contenant la liste d'objets d'arborescence DOM générés pour chaque page Web d'URL. Les auteurs préconisent davantage de recherches pour identifier des caractéristiques importantes basées sur le contenu pour la détection de phishing sur les pages Web. Ils constatent que les caractéristiques basées sur des services externes sont cruciales et que les caractéristiques basées sur les hyperliens sont nettement plus lentes que toutes les caractéristiques basées sur des services externes.

Les auteurs ont mené des expériences avec différentes combinaisons de modèles et ont exploré les effets des méthodes de filtre et d'enveloppe sur la sélection des caractéristiques discriminantes. Ils concluent que Random Forest est le classificateur le plus prédictif et que les caractéristiques obtenues à partir de services externes sont les plus discriminantes. Les auteurs suggèrent que le classement des caractéristiques améliore la précision des modèles Random Forest et, associé à une sélection décroissante, permet d'obtenir de meilleures performances que celles obtenues avec des évaluateurs d'enveloppe et avec une consommation de temps moindre, sauf pour l'algorithme Boruta.

Hannousse.A et Yahiouche.S[10], expliquent comment ils ont concrétisé leurs résultats en développant un plugin passif destiné au navigateur Google Chrome. Ce plugin a pour fonction d'ouvrir une fenêtre contextuelle informant les utilisateurs du statut de légitimité de la page actuellement affichée dans le navigateur, comme illustré dans la Figure 18. Le plugin tire parti du meilleur modèle identifié au cours de la présente étude, à savoir le Random Forest entraîné sur 73 caractéristiques classées selon le filtre du chi carré.

De plus, les auteurs mentionnent que le plugin peut également être utilisé pour évaluer son efficacité par rapport à l'avertisseur actif intégré dans Google Chrome, comme le montre la Figure 18(a). Cela suggère que le plugin offre une alternative ou un complément à la fonction de sécurité active déjà présente dans le navigateur, fournissant ainsi aux utilisateurs une double couche de protection contre les pages potentiellement malveillantes.

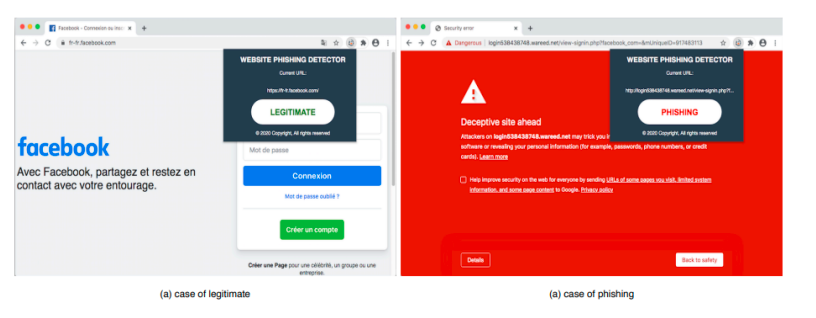


Figure 18 Captures d'écran du plugin passif développé pour le navigateur Chrome

## 3.7. Conclusion

En résumé, cet article apporte une contribution importante au domaine de la détection des attaques de phishing. Les auteurs ont développé une solution robuste qui intègre l'analyse avancée et l'apprentissage automatique pour corriger les défauts des méthodes traditionnelles. Leur méthodologie, associée à une validation en règle, ouvre la porte à des systèmes de sécurité plus efficaces qui s'adaptent à un environnement en constante évolution.

# CHAPITRE 4 : Analyse comparative avec les travaux existants

## 4.1. Introduction

L'article « Towards Benchmark Datasets for Machine Learning Based Website Phishing Detection » propose une approche innovante pour développer des ensembles de données reproductibles et extensibles, destinés à la détection de sites web de phishing. Ce chapitre analyse les différences entre cette étude et les travaux existants, en s'appuyant sur la section "Related Work" de l'article.

## 4.2. TRAVAUX CONNEXES

La section "Related Work" examine un ensemble significatif de travaux antérieurs axés sur la détection de phishing à l'aide de techniques d'apprentissage automatique. La figure 19 montre l'évolution du nombre de publications dans le domaine de la détection de phishing par apprentissage automatique au cours des dix dernières années. On observe une croissance constante du nombre de publications, ce qui témoigne de l'intérêt croissant porté à ce domaine. Les caractéristiques les plus couramment utilisées sont les caractéristiques basées sur l'URL et le contenu. Cependant, la plupart des études existantes négligent les performances d'exécution et n'expérimentent pas l'importance des différentes classes de caractéristiques ni l'effet de la combinaison de classificateurs.

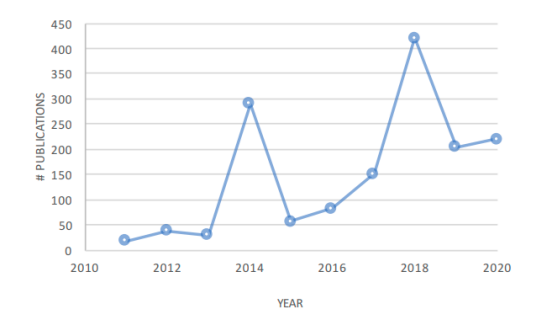


Figure 19 Nombre de publications par an portant sur l'utilisation de l'apprentissage automatique pour la détection des sites web d'hameçonnage.

Les résultats ont été obtenus le 16 septembre 2020 à partir de https ://app.dimensions.ai en utilisant la requête suivante : « site web » AND « phishing detection » AND « machine learning ».

Voici un résumé des principales contributions et des limites de ces travaux :

* Revues Systématiques : Dou et al. [11] et Das et al. [2] ont effectué des revues systématiques. Dou et al. Ont souligné la négligence des performances en temps réel par la plupart des systèmes et l'utilisation fréquente de caractéristiques basées sur les URL et le contenu. Das et al. Ont classé les classifieurs en fonction du nombre d'études les adoptant.
* Caractéristiques et Performances : Jain et al. [13] ont extrait des caractéristiques basées sur la nature des hyperliens dans le contenu des pages Web, montrant que la régression logistique avait le meilleur score de précision. D'autres travaux ont utilisé des ensembles hybrides de caractéristiques, montrant des performances allant jusqu'à 98.25 pourcents.
* Sélection de Caractéristiques : Différentes méthodes de sélection de caractéristiques ont été explorées, telles que l'utilisation de filtres de classement et d'algorithmes d'enveloppe. Certains travaux ont introduit de nouvelles métriques de classement, comme Rajab et al. [14], qui a combiné les rangs estimés par des algorithmes de gain d'information et de chi carré.
* Ensembles et Combinaisons de Classifieurs : Certains travaux ont exploré la combinaison de plusieurs classifieurs pour améliorer la précision, notamment l'utilisation d'ensembles tels que proposé par Nagaraj et al. [15], qui a développé un ensemble à deux volets en utilisant les sorties d'un classifieur de Random Forest pour alimenter un classifieur de réseau neuronal.
* Les travaux existants sur la détection de phishing basée sur l'apprentissage automatique présentent des lacunes notables. Tout d'abord, la négligence des performances d'exécution dans la plupart des systèmes actuels limite leur applicabilité dans des scénarios nécessitant une réactivité en temps réel. De plus, le manque d'expérimentation sur l'importance des différentes classes de caractéristiques, telles que celles basées sur l'URL, le contenu, le comportement de l'utilisateur et les interactions utilisateur-site, laisse des questions cruciales sans réponse. En outre, le défaut d'expérimentation sur l'effet bénéfique de la combinaison de classificateurs entrave le potentiel d'amélioration des performances de la détection de phishing.

# CONCLUSION GENIRALE

L'étude propose des lignes directrices pour la création de jeux de données reproductibles et extensibles pour la détection du phishing sur les sites Web, qui peuvent servir de points de référence pour les systèmes basés sur l'apprentissage automatique. Les résultats montrent que les systèmes basés sur un classificateur Random Forest peuvent être utilisés pour prédire efficacement les pages Web de phishing, avec une précision plus élevée que tous les autres classificateurs pour différentes classes d'entité. Les fonctionnalités externes basées sur les services sont plus efficaces pour distinguer les pages Web de phishing, mais peuvent provoquer un retard du réseau. Les fonctionnalités basées sur le contenu sont moins discriminantes, et certaines fonctionnalités basées sur des hyperliens peuvent également causer des retards sur le réseau. Les chercheurs sont encouragés à examiner le contenu des sites Web de phishing pour identifier les fonctionnalités plus efficaces. L'utilisation de fonctionnalités hybrides offre une meilleure précision que l'utilisation d'une seule classe de fonctionnalités. Les travaux futurs incluent la validation des conclusions tirées en expérimentant davantage de jeux de données construits de la même manière et l'intégration d'approches de Deep Learning pour la détection du phishing

# Références

[1] El Aassal, A., Baki, S., Das, A., & Verma, R. M. (2020). An in-depth benchmarking and evaluation of phishing detection research for security needs. IEEE Access, 8, 22170–22192.

[2] Frank, E., Hall, M., Holmes, G., Kirkby, R., Pfahringer, B., Witten, I. H., & Trigg, L. (2010). Weka-a machine learning workbench for data mining. In O. Maimon, and L. Rokach (Eds.), Data Mining and Knowledge Discovery Handbook (pp. 1269–1277). Boston, MA: Springer US.

[2] Das, A., Baki, S., Aassal, A. E., Verma, R. M., & Dunbar, A. (2020). Sok: A comprehensive reexamination of phishing research from the security perspective. IEEE Commun. Surv. Tutorials, 22, 671–708.

[3] Korkmaz, M., Sahingoz, O. K., & Diri, B. (2020). Feature selections for the classification of webpages to detect phishing attacks: A survey. In 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA) (pp. 1–9). Ankara, Turkey.

[4]Das, A., Baki, S., Aassal, A. E., Verma, R. M., & Dunbar, A. (2020). Sok: A comprehensive reexamination of phishing research from the security perspective. IEEE Commun. Surv. Tutorials, 22, 671–708.

[5] Althobaiti, K., Rummani, G., & Vaniea, K. (2019). A review of human- and computer-facing url phishing features. In 2019 IEEE European Symposium on Security and Privacy Workshops (EuroS PW) (pp. 182–191). Stockholm, Sweden: IEEE.

[6] Kira, K., & Rendell, L. A. (1992). A practical approach to feature selection. In D. Sleeman, and P. Edwards (Eds.), Machine Learning Proceedings 1992 (pp. 249 – 256). San Francisco (CA): Morgan Kaufmann.

[7] Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. Artificial Intelligence, 97, 273 – 324.

[8] Choo, X., Chiew, K., Ibrahim, D., Musa, N., Sze, S., & Tiong, W. (2016). Feature-based phishing detection technique. Journal of Theoretical and Applied Information Technology, 91, 101–106.

[9] Kursa, M. B., & Rudnicki, W. R. (2010). Feature selection with the boruta package. Journal of of Statistical Software, 36, 1–13.

[10] Hannousse, A., & Yahiouche, S. (2020). Web page phishing detection. Mendeley Data,, V2. doi:doi:10.17632/c2gw7fy2j4.2.

[11] Dou, Z., Khalil, I., Khreishah, A., Al-Fuqaha, A., & Guizani, M. (2017). Systematization of knowledge (sok): A systematic review of software-based web phishing detection. IEEE Communications Surveys and Tutorials, 19, 2797–2819.

[12] Jain, A. K., & Gupta, B. B. (2018). Phish-safe: Url features-based phishing detection system using machine learning. In M. U. Bokhari, N. Agrawal, and D. Saini (Eds.), Cyber Security (pp. 467–474). Singapore: Springer Singapore.

[13] Jain, A. K., & Gupta, B. B. (2019). A machine learning based approach for phishing detection using hyperlinks information. Journal of Ambient Intelligence and Humanized Computing, 10, 2015–2028.

[14] Rajab, K. D. (2017). New hybrid features selection method: A case study on websites phishing. Security and Communication Networks, (pp. 1–10).

[15] Nagaraj, K., Bhattacharjee, B., Sridhar, A., & GS, S. (2018). Detection of phishing websites using a novel twofold ensemble model. Journal of Systems and Information Technology, 20, 321–357.

[16] Kursa, M. B., & Rudnicki, W. R. (2010). Feature selection with the boruta package. Journal of of Statistical Software, 36, 1–13.

[17] Picard, R. R., & Cook, R. D. (1984). Cross-validation of regression models. Journal of the American Statistical Association, 79, 575–583.

[18] Rao, R. S., Vaishnavi, T., & Pais, A. R. (2019). Catchphish: detection of phishing websites by inspecting urls. J. Ambient Intell. Humaniz. Comput., 11, 813–825.

[19] Sahingoz, O. K., Buber, E., Demir, O., & Diri, B. (2019). Machine learning based phishing detection from urls. Expert Systems with Applications, 117, 345 – 357.

[20] Sameen, M., Han, K., & Hwang, S. O. (2020). Phishhaven—an efficient real-time ai phishing urls detection system. IEEE Access, 8, 83425–83443.

[21] Shirazi, H., Bezawada, B., & Ray, I. (2018). "know thy domain name”: Unbiased phishing detection using domain name based features. In Proceedings of the 23nd ACM on Symposium on Access Control Models and Technologies SACMAT ’18 (pp. 69–75). New York, NY, USA: Association for Computing Machinery.

[22] Wandera (2020). Mobile Threat Landscape 2020: Understanding the key trends in mobile enterprise security in 2020. Technical Report. <https://www.wandera.com/mobile-threat-landscape/>.

[23] Witten, I. H., Frank, E., & Hall, M. A. (2011). Data Mining: Practical Machine Learning

Tools and Techniques. (3rd ed.). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. [24] Zaini, N., Stiawan, D., Faizal, M., Firdaus, A., Wan Din, W. I. S., Kasim, S., & Sutikno, T. (2020). Phishing detection system using machine learning classifiers. Indonesian Journal of Electrical Engineering and Computer Science, 17, 1165–1171.

UCI repository: <https://archive.ics.uci.edu/>

Dimensions research grants database: <https://app.dimensions.ai>

Alexa website: <https://www.alexa.com/>

Yandex search engine: <https://tech.yandex.com.tr/xml/>

Phishtank website: https://www.phishtank.com/

Openphish website: <https://openphish.com/>

HTML DOM Parser package for Python: <http://thehtmldom.sourceforge.net>

Python pickle module: <https://docs.python.org/3/library/pickle.html>

WHOIS service: https://www.domain.com/whois

Openpagerank website: https://openpagerank.com

Google search engine: <https://www.google.com>

Python implementation of Boruta wrapper algorithm: <https://pypi.org/project/Boruta/>

chatgpt