Quand les Bots Déjouent l'Apprentissage : Enjeux et Défis de la Détection

Mohsine Aabid, Patrice Bellot, Simon Dumas Primbault

Quand les Bots Déjouent l'Apprentissage : Enjeux et Défis de la Détection

Mohsine AABID[1, 2] Patrice Bellot[1] Simon Dumas Primbault[2]
(1) Laboratoire d'informatique et systèmes (LIS), 52 Av. Escadrille Normandie Niemen, 13013 Marseille, France
(2) OpenEdtionLab, 22 Rue John Maynard Keynes, 13013 Marseille, France
mohsine.aabid@lis-lab.fr, patrice.bellot@lis-lab.fr,
simon.dumas-primbault@openedition.org

RÉSUMÉ

Identifier les bots d'une une bibliothèque numérique est un défi crucial pour analyser avec précision le comportement des utilisateurs afin de mieux répondre à leurs besoins. Mais que se passe-t-il lorsque les modèles de détection sont confrontés à des données provenant d'une période différente de leur période d'entraînement ? Cet article explore cette question en extrayant des caractéristiques clés, telles que la durée de l'activité et le nombre de requêtes, nous comparons plusieurs modèles d'apprentissage supervisé et évaluons la robustesse de cette approche face aux variations temporelles. Nos observations préliminaires montrent que les modèles de détection tendent à être plus confiant sur les données issues de leur période d'entraînement, ce qui soulève des questions sur leur capacité à généraliser à des périodes différentes. Cette dépendance met en lumière la nécessité de stratégies adaptatives, telles que des mises à jour régulières des modèles et de nouvelles approches d'apprentissage, afin de saisir l'évolution des comportements automatisés et améliorer la robustesse de la détection.

ABSTRACT

When Bots Outsmart Learning : Issues and Challenges of Detection

Identifying bots in a digital library is a crucial challenge for accurately analyzing user behavior to better meet their needs. But what happens when detection models are confronted with data from a period different from their training period ? This article explores this question by extracting key features (duration of activity, number of queries *etc.*), comparing several supervised learning models and assessing the robustness of this approach in the face of temporal variations. Our preliminary observations show that detection models tend to be more confident on data from their training period, raising questions about their ability to generalize to different periods. This dependency highlights the need for adaptive strategies, such as regular model updates and new learning approaches, to better capture the evolution of automated behavior and improve detection robustness.

MOTS-CLÉS : Détection des bots, Apprentissage supervisé, Filtrage des sessions, Internet, Bibliothèque numérique.

KEYWORDS: Bots detection, Supervised learning, Session filtering, Internet, Digital library.

1 Introduction

Le trafic web est aujourd'hui en grande partie automatisé, avec des bots représentant près de la moitié des interactions en ligne [1]. Ces programmes, conçus pour interagir avec des sites en suivant des instructions spécifiques, remplissent diverses fonctions, allant de l'indexation de contenu pour les moteurs de recherche aux actions malveillantes, telles que le spam et les attaques DDoS. constitue un défi majeur pour les plateformes en ligne, en particulier dans le domaine des bibliothèques numériques, où il est crucial de distinguer les sessions humaines

1. 2024 Bad Bots Report

des sessions automatisées afin d'assurer des analyses fiables et une meilleure compréhension des besoins des internautes.

Contrairement aux plateformes commerciales, où la détection des bots vise principalement à prévenir la fraude publicitaire ou les attaques sur les systèmes transactionnels, les bibliothèques numériques font face à des enjeux bien spécifiques. Leur mission première étant la diffusion du savoir et l'accessibilité des ressources académiques, elles doivent composer avec des bots légitimes, comme ceux des moteurs de recherche indexant des publications, et des bots malveillants cherchant à contourner les restrictions d'accès, à extraire massivement des contenus ou à perturber le fonctionnement des plateformes.

L'impact des bots dépasse la simple question du trafic web. Leur présence peut fausser les statistiques d'usage, surévaluer artificiellement l'intérêt pour certaines ressources et compliquer l'analyse des comportements réels des utilisateurs. Distinguer les bots permettrait de mieux apprendre des modèles pour la recommandation de contenus ou la création de profils personnalisés. Il est donc crucial de concevoir des stratégies robustes permettant d'identifier et d'écarter les bots.

Cependant, l'identification des bots à partir des sessions de navigation — définies ici comme des séquences de requêtes extraites des fichiers journaux — demeure une tâche complexe, d'autant plus lorsque peu de données labellisées sont disponibles. Contrairement aux méthodes classiques de détection reposant sur des listes d'IP ou des signatures d'agent utilisateur, L'objectif est d'évaluer dans quelle mesure ces caractéristiques — telles que la durée de l'activité ou le nombre de requêtes — permettent de distinguer efficacement les sessions humaines de celles générées par des bots, et d'évaluer la robustesse des modèles d'apprentissage automatique face à l'évolution à moyen et long terme des comportements des bots. En effet, un modèle entraîné sur les données d'une année peut-il maintenir son efficacité lorsqu'il est appliqué sur une autre ?

Dans cet article, nous comparons différentes **méthodes d'apprentissage automatique supervisées** vu dans la littérature. Nous analysons leur capacité à identifier les bots à partir de leurs comportements et évaluons la stabilité des modèles au fil du temps.

2 État de l'art

La littérature distingue deux grandes approches pour la détection des bots : les méthodes ***hors ligne***, qui analysent les sessions a posteriori, et les méthodes en ***temps réel***. Doran & Gokhale (2011) les répartissent en quatre catégories : syntaxique, fondées sur des motifs, par apprentissage automatique (hors ligne), et les tests de Turing (temps réel). **Les méthodes syntaxiques** reposent sur l'identification de motifs spécifiques, tels que la présence du terme « bot » dans l'agent utilisateur ou l'association à une adresse IP connue. Bien que faciles à déployer, elles restent efficaces uniquement contre des bots connus et requièrent des mises à jour régulières. **Les méthodes basées sur les motifs** extraient des motifs typiques dans les sessions, tels que des requêtes répétitives, et établissent des règles permettant de distinguer les humains des bots. Toutefois, elles demeurent limitées par des modèles préétablis et peuvent s'avérer inefficaces face à des bots adoptant un comportement atypique. (Geens *et al.*, 2006; Kwon *et al.*, 2012). **Les systèmes de test de Turing**, principalement utilisés pour la détection en temps réel, impliquent une interaction entre l'utilisateur et le serveur, comme les tests CAPTCHA (*Completely Automated Public Turing test to tell Computers and Humans Apart*). Bien qu'efficaces contre les bots classiques, ils le sont moins face aux bots avancés qui parviennent à les contourner (Guerar *et al.*, 2021) ou qui délèguent cette tâche à des humains via des services rémunérés.

Les méthodes d'apprentissage estiment la probabilité qu'une session soit automatisée. Elles se classent en quatre catégories, **les méthodes supervisées**, (Tan & Kumar, 2002) utilise un arbre de décision C4.5, tandis que (Stevanovic *et al.*, 2012) compare divers algorithmes (arbres de décision, réseaux de neurones, SVM, k-NN) et introduit de nouvelles caractéristiques comme le pourcentage de requêtes séquentielles. D'autres travaux exploitent des chaînes de Markov pour modéliser les séquences de requêtes (Derek Doran, 2016), combinent réseaux de neurones et tests statistiques (Suchacka *et al.*, 2021), ou appliquent une analyse sémantique des

documents visités afin de différencier bots et humains (Lagopoulos & Tsoumakas, 2020), (Jagat *et al.*, 2024). **Les méthodes non supervisées** se basent sur des techniques de partitionnement. Alam *et al.* (2014) appliquent l'optimisation par essaims particulaires pour repérer les sessions aberrantes, tandis que Zabihimayvan *et al.* (2017) utilisent un regroupement markovien (MCL) et une sélection de caractéristiques basée sur la théorie des ensembles approximatifs. Rovetta *et al.* (2020) comparent *k-Means* et *c-Means* aux approches supervisées et montrent qu'elles sont aussi efficaces, bien que certaines sessions courtes restent non détectées. En **semi-supervisé**, Jagat *et al.* (2023) proposent un *Sparse Stacked Auto Encoder* entraîné en deux phases : apprentissage des représentations latentes, puis classification supervisée. Enfin, les **méthodes par renforcement** incluent des approches simulant l'évolution des stratégies des bots pour tromper le modèle (Iliou *et al.*, 2022) ou optimisant la sélection des caractéristiques pour améliorer la détection (Gao *et al.*, 2023).

Notre méthodologie suit celle proposée par Iliou *et al.* (2019) qui décrivent un processus en plusieurs étapes : extraction des sessions de navigation, sélection et annotation des caractéristiques et classification des sessions via des modèles d'apprentissage. Les méthodes d'apprentissage sont aujourd'hui prédominantes en raison de leur flexibilité et de leur capacité à s'adapter à des bots de plus en plus sophistiqués. Le principal apport de cet article est de revisiter ces approches sur des logs récents dans le cadre d'une bibliothèque numérique qui reçoit plus de 92 millions de visites annuelles [2].

Les méthodes de l'état de l'art partagent rarement leurs données d'entraînement. Dans cette revue de littérature, un seul article publie un jeu de données couvrant une période d'un mois [3]. Pourtant, plusieurs auteurs soulignent l'impact de la période d'entraînement sur la performance des modèles et la nécessité de mises à jour pour la maintenir. À notre connaissance, aucun travail n'a évalué la stabilité de ces modèles sur deux périodes distinctes. Dans cet article nous avons décidé de nous intéresser uniquement à la stabilité des méthodes d'apprentissage supervisés. Les logs de 2022 étant difficiles à annoter, nous avons choisi d'entraîner le modèle sur les logs de 2023, facilement annotables [4]. Cette approche nous permet ensuite d'évaluer la stabilité du modèle en le testant sur les données de 2022.

3 Extraction des caractéristiques & étiquetage des sessions

3.1 Extraction des caractéristiques des sessions

Les sessions étudiées proviennent des logs serveurs d'OpenEdition.org, une bibliothèque numérique en sciences humaines et sociales en accès ouvert. OpenEdition nous a fourni les données des mois d'octobre 2022 et 2023. Chaque session est constituée d'une séquence de requêtes HTTP, comprenant l'adresse IP anonymisée du client ayant effectué une requête, l'horodatage local de réception de la requête, des informations associées (méthode HTTP, URL, version HTTP), un code de statut HTTP (succès, refus, erreur…), la taille des données envoyées, l'URL de provenance (référent) et un indicateur d'accès aux ressources (accès complet, partiel ou restreint).

Les sessions doivent tout d'abord être repérées, avant d'être catégorisées. Nous réalisons cette étape en regroupant les requêtes et en tenant compte de deux critères : l'adresse IP et l'agent utilisateur. Ces requêtes sont ensuite triées par ordre chronologique afin de refléter la séquence réelle de navigation des utilisateurs et des bots. Une segmentation temporelle est ensuite appliquée. La méthode classique consiste à fixer un seuil arbitraire de 30 minutes d'inactivité entre deux requêtes successives : si l'intervalle dépasse ce seuil, une nouvelle session est identifiée. Cependant, cette approche présente une rigidité excessive, car les comportements des utilisateurs varient selon les contextes et les usages. Pour pallier cette limitation, nous adoptons une heuristique probabiliste qui évalue la continuité de la session en fonction du temps écoulé entre deux requêtes, plutôt que de s'appuyer sur

2. Rapport d'activité OpenEdition 2023
3. Web robot detection - Server log Creators Dataset
4. L'annotation automatique, limitée par l'absence d'archives externes, reste difficile sur d'anciens logs. En 2023, quelques IP institutionnelles ont pu être récupérées pour faire la tâche.

un seuil fixe. Cette méthode repose sur une fonction dont la valeur diminue progressivement avec l'augmentation de l'intervalle de temps, permettant d'ajuster dynamiquement la segmentation en fonction des données observées. De plus, nous intégrons le référent : si l'URL de la requête actuelle correspond à l'URL de provenance de la requête précédente, la probabilité d'appartenance à la même session est augmentée. Ces informations sont alors utilisées pour construire une représentation vectorielle (voir table 1).

Caractéristique	**Description**
Durée d'activité	Durée entre la première et la dernière requête dans une session
Nombre de requêtes	Nombre total de requêtes
Temps moyen	Moyenne des durées entre deux requêtes consécutives dans une session
Écart moyen des temps	Écart type des durées entre deux requêtes consécutives dans une session
% Requêtes uniques	Pourcentage de requêtes uniques dans une session
% Requêtes séquentielles	Pourcentage de requêtes dans le même répertoire (exemple : `/rep/url1` suivi de `rep/url2`)
% VERBE	Pourcentages de requêtes avec un verbe (GET, POST, HEAD OPTIONS), un pourcentage pour chaque verbe.
% Requêtes nuit	Pourcentage de requêtes faites la nuit (entre 21h et 5h inclus)
% Requêtes sans referer	Pourcentage de requêtes sans *référent*
% Statut	Pourcentages de requête avec un statut (2XX, 3XX, 4XX ou 5XX)
Moyenne profondeurs	Moyenne des profondeurs des URLs visitées dans une session
Écart type profondeurs	Écart type des profondeurs des URLs visitées dans une session
Taille du transfert	Taille totale en octets envoyés par le serveur dans une session
nb pdf	Nombre de fichiers PDF téléchargés
nb epub	Nombre de fichiers ePub téléchargés
BS	Vitesse de navigation
Pénalité	Nombre d'allers-retours dans une session

TABLE 1 – Caractéristiques des sessions de navigation, les caractéristiques %VERBE et %Status correspondent à plusieurs caractéristiques (exp : %GET, % HEAD... dans %VERBE).

Nous explicitons la façon avec laquelle nous avons calculé certaines caractéristiques :

— Pourcentage de requêtes séquentielles $= \frac{\text{Nombre de requêtes successives dans le même répertoire}}{\text{Nombre total de requêtes dans la session} - 1}$

— Vitesse de Navigation (BS) = $\frac{\text{Nombre de requêtes}}{\text{Durée d'activité}}$

3.2 Annotation des sessions pour l'entraînement

L'annotation consiste à attribuer une classe à chaque session : Bot, Humain ou Indéterminé. Pour constituer un jeu d'entraînement, les méthodes d'annotation peuvent être divisées en deux grandes catégories :

— **les méthodes automatiques** où les sessions sont étiquetées en se basant sur des informations comme l'adresse IP ou l'agent utilisateur. Ces méthodes s'appuient souvent sur des bases de données externes, comme des listes noires [5] (par exemple, des bases collaboratives contenant des IP identifiées comme suspectes via le *crowdsourcing*). Bien que rapides, ces méthodes ne détectent pas toujours les nouveaux types de bots ;

— **l'annotation manuelle** des sessions est effectuée par des experts qui identifient l'usage automatisé. Cette méthode est précise mais lente et coûteuse.

5. Udger, GreyNoise, AbuseIPDB, BotScout

Notre tâche d'étiquetage s'articule en quatre étapes. Dans un premier temps, nous examinons l'agent utilisateur afin de déterminer s'il correspond à un bot, en le comparant à une liste préétablie ou en recherchant des mots-clés spécifiques tels que « bot » ou « crawler ». Ensuite, nous avons sollicité OpenEdition pour vérifier si les adresses IP des sessions figuraient sur des listes noires, ce qui pourrait indiquer une activité suspecte, puis appliqué un filtre vérifiant si certaines sessions dépassent un seuil donné de requêtes. Enfin, l'identification des sessions humaines repose sur la vérification de l'appartenance de l'adresse IP à une institution, telle qu'une bibliothèque universitaire. Nous faisons l'hypothèse qu'il est peu probable que ces entités utilisent des bots pour consulter des sites. Malgré les filtres mis en place, certains bots parviennent encore à passer entre les mailles du filet. En analysant les sessions restantes, nous avons observé que certaines présentaient un nombre de requêtes anormalement élevé, bien au-delà d'une navigation humaine classique.

Afin d'améliorer la qualité de la classification, les sessions dépassant 4 500 requêtes ont été annotées comme bots, ce seuil correspondant à 90% de la plus longue session observée. Ce choix permet de maximiser la conservation des sessions humaines tout en filtrant efficacement les bots les plus actifs. La figure 1 résume le processus d'étiquetage des sessions.

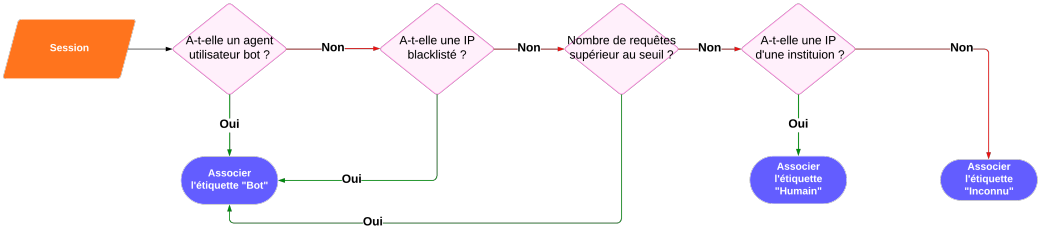

FIGURE 1 – Annotation automatique du jeu d'entraînement.

4 Catégorisation selon un modèle appris

Pour comparer les sessions des mois d'octobre 2022 et 2023, nous appliquons la tâche d'annotation précédente uniquement sur les logs de 2023. Ces sessions serviront ensuite à entraîner des modèles. Nous avons choisi d'entraîner le modèle sur l'année 2023 en raison du faible nombre d'étiquettes disponibles pour 2022. Bien que l'entraînement sur une année postérieure à celle du test puisse sembler contre-intuitif, cela permet d'évaluer la capacité du modèle à détecter des comportements stables dans le temps, en supposant une évolution technologique limitée des bots sur une courte période.

Avant d'entraîner les modèles, plusieurs transformations sont appliquées afin d'améliorer la performance de la classification. Tout d'abord, un équilibrage des classes est effectué : nous sélectionnons 30 000 sessions humaines et 35 000 sessions de bots d'un ensemble contenant plus de 21 millions de sessions,puis augmentons artificiellement l'ensemble de données de 65 000 à 68 500 en faveur des sessions humaines, en utilisant la méthode SMOTE (Chawla *et al.*, 2002) afin d'améliorer la représentativité des classes. Ensuite, une normalisation des variables est appliquée pour homogénéiser les échelles et améliorer la convergence des modèles d'apprentissage. Nous utilisons le Z-score pour centrer et réduire les valeurs, ainsi que la transformation de Yeo-Johnson (kwon Yeo & Johnson, 2000) afin d'approcher une distribution normale lorsque cela est nécessaire. Pour limiter la complexité du modèle et éviter le surajustement, Nous effectuons une sélection des variables en supprimant celles qui sont fortement corrélées ou dont la variance est trop faible, car elles n'apportent que peu de valeur pour la classification. Enfin, le jeu de données (celui avec les logs de 2023) est divisé en 49 000 sessions d'entraînement et 19 500 sessions de test. Une validation croisée est appliquée sur les données d'entraînement avec 10 partitions.

Voici la liste des hyperparamètres des 5 premiers modèles qui performent le mieux :

Modèle	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
Extreme Gradient Boosting	0.9037	0.9645	0.9301	0.8952	0.9123	0.8056	0.8064
CatBoost Classifier	0.9031	0.9648	0.9292	0.8950	0.9117	0.8045	0.8053
LightGBM	0.9013	0.9637	0.9286	0.8924	0.9101	0.8007	0.8015
Random Forest Classifier	0.8952	0.9541	0.9002	0.9046	0.9024	0.7892	0.7892
Extra Trees Classifier	0.8946	0.9384	0.8980	0.9056	0.9017	0.7881	0.7882
K Neighbors Classifier	0.8944	0.9450	0.9060	0.8987	0.9023	0.7873	0.7874
Gradient Boosting Classifier	0.8912	0.9547	0.9220	0.8815	0.9013	0.7803	0.7813
Decision Tree Classifier	0.8796	0.8853	0.8821	0.8930	0.8875	0.7579	0.7580
AdaBoost Classifier	0.8672	0.9345	0.8922	0.8653	0.8786	0.7321	0.7326
Naive Bayes	0.8550	0.8982	0.8744	0.8589	0.8665	0.7078	0.7080
Quadratic Discriminant Analysis	0.8492	0.8996	0.8795	0.8465	0.8627	0.6957	0.6963
SVM - Linear Kernel	0.8480	0.8954	0.8697	0.8512	0.8603	0.6936	0.6938
Ridge Classifier	0.8471	0.8998	0.8680	0.8510	0.8594	0.6918	0.6920
Linear Discriminant Analysis	0.8471	0.8996	0.8679	0.8511	0.8594	0.6918	0.6920
Logistic Regression	0.8465	0.8998	0.8629	0.8536	0.8582	0.6909	0.6909

TABLE 2 – Performances des modèles de classification.

Hyperparameter	CatBoost	XGBoost	LightGBM	Random Forest	Extra Trees
Learning Rate	0.1	0.3	0.1	-	-
Max Depth	6	no limit	no limit	no limit	no limit
n_Estimators	-	100	100	100	100
Colsample_bytree	-	1	1	sqrt	sqrt
Bootstrap	-	-	-	True	False
Criterion	-	-	-	gini	gini
Min Samples Split	-	-	-	2	2
Min Samples Leaf	-	-	-	1	1
Subsample	-	1	1	-	-

TABLE 3 – Liste des hyperparamètres des 5 premiers modèles les plus performants

Comme on le voit sur la table 2, l'approche *Extreme Gradient Boosting* permet d'obtenir les meilleurs résultats. Les graphiques 2 et 3 illustrent la distribution des sessions et des requêtes entre les bots et les humains pour les années 2022 et 2023 à partir des prédictions du modèle. Nous observons que la proportion des requêtes et des sessions attribuées aux bots en 2023 est supérieure à celles de 2022. Les proportions de bots obtenus en 2023 et 2022 sont proches des proportions qu'on trouve dans les études faites sur les bots [6]. Avec une proportion un peu supérieure pour les données de 2023. Une analyse plus profonde doit être faite pour savoir si cette distribution est propre au site ou par le fait que le modèle soit entraîné sur des sessions de la même année, ce qui pourrait favoriser la détection des sessions et requêtes bots.

La figure 4 présente l'importance de chaque variable utilisée pour la prédiction. On constate que la *durée moyenne entre requêtes* (mean_duration_time_between_requests) est le facteur déterminant. Notre hypothèse est qu'un bot, cherchant à parcourir un maximum de ressources en un temps restreint, enchaîne généralement des requêtes à intervalles très rapprochés, tandis qu'un humain prend plus de temps pour consulter le contenu,

6. 2022 The Big Bad Bot Problem, 2024 Bad Bots Report

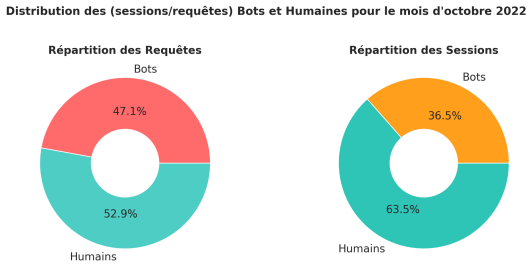

FIGURE 2 – Distribution des sessions et requêtes pour l'année 2022

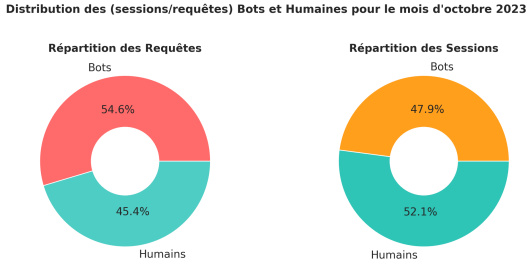

FIGURE 3 – Distribution des sessions et requêtes pour l'année 2023

augmentant ainsi l'intervalle moyen entre ses requêtes. En seconde position, le *nombre total d'octets transférés* (total_bytes_sent) reflète la quantité de données échangées au cours d'une session. Un bot peut télécharger ou demander un grand volume de ressources, alors qu'un utilisateur humain se limite souvent à quelques pages ou documents. La *profondeur moyenne* (mean_depth) et la *durée totale d'activité* (activity_duration) complètent ces facteurs : un bot peut, par exemple, explorer de nombreux niveaux d'un site pour moissonner les liens, avec une durée longue, alors qu'un humain navigue plus sélectivement et a une durée d'activité courte.

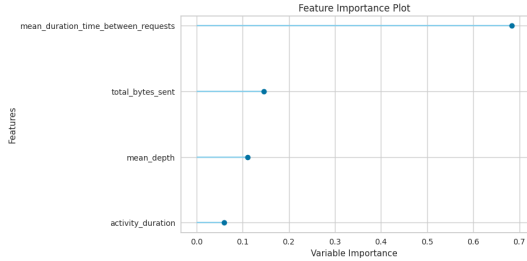

FIGURE 4 – Importance des caractéristiques dans les prédictions

Les figures 5 et 6 montrent la distribution des probabilités de classification pour les sessions humaines et bots en 2022 et 2023. Le modèle est très confiant pour les sessions humaines, avec un pic marqué près de 1. En revanche, pour les bots, la distribution est plus étalée : en 2022, le pic principal est autour de 0,75, indiquant une incertitude plus grande ; tandis qu'en 2023, un pic plus prononcé à 1 montre une meilleure détection des bots. Cette différence suggère que le modèle, entraîné sur les sessions de 2023, s'adapte mieux à ces données.

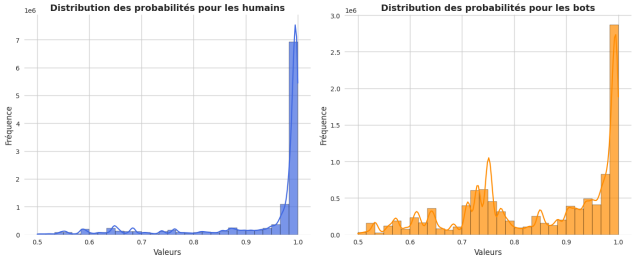

FIGURE 5 – Distribution des probabilités de prédiction pour l'année 2023.

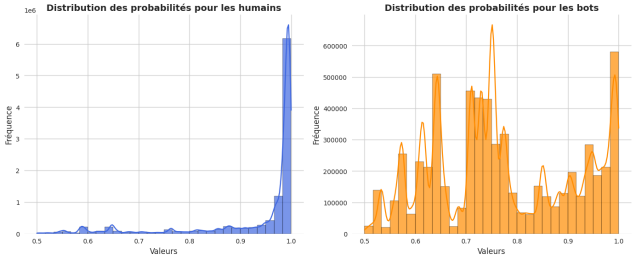

FIGURE 6 – Distribution des probabilités de prédiction pour l'année 2022.

5 Discussion

La détection des bots dans le trafic web, notamment au sein des bibliothèques numériques, reste un défi évolutif en raison des changements constants dans les comportements automatisés. Nos observations préliminaires suggèrent que les modèles supervisés sont fortement influencés par les données sur lesquelles ils ont été entraînés, ce qui interroge leur capacité à généraliser sur des périodes ultérieures. Cette dépendance temporelle pourrait être liée aux biais d'étiquetage — tels que l'utilisation de listes d'IP non archivées ou certaines hypothèses sur la nature des sessions institutionnelles — ainsi qu'aux limites des caractéristiques actuellement exploitées (durée entre requêtes, volume d'octets transférés, profondeur de navigation, etc.), qui ne permettent pas toujours de discriminer efficacement les bots des utilisateurs humains.

Plusieurs pistes sont en cours d'exploration pour mieux comprendre ces limites et améliorer la robustesse des approches de détection. Un axe de travail consiste à estimer plus précisément la proportion réelle de bots et d'humains dans les sessions, afin d'obtenir une évaluation plus fiable des performances des modèles. Par ailleurs, tester ces modèles sur des périodes successives permettra d'analyser l'évolution de leur confiance et d'identifier d'éventuelles tendances dans leur capacité à détecter les bots au fil du temps. D'autres perspectives incluent l'intégration d'analyses comportementales plus fines et le développement de modèles capables de s'adapter aux évolutions des bots. Une distinction plus précise entre les bots malveillants et les bots légitimes (tels que les crawlers des moteurs de recherche ou les outils d'archivage) pourrait également affiner les stratégies de détection et réduire les erreurs de classification. Enfin, plusieurs questions restent ouvertes : peut-on détecter des bots adoptant des comportements humains ? Quelles caractéristiques additionnelles pourraient permettre une distinction plus fiable ? Une collaboration entre experts en sécurité et en apprentissage automatique est cruciale pour relever les défis posés par l'évolution rapide des bots en ligne.

Remerciements

Ce travail a bénéficié d'une aide de l'État gérée par l'Agence Nationale de la Recherche au titre du Programme d'Investissements d'Avenir portant la référence ANR-21-ESRE-0045

Références

ALAM S., DOBBIE G., KOH Y. S. & RIDDLE P. J. (2014). Web bots detection using particle swarm optimization based clustering. *2014 IEEE Congress on Evolutionary Computation (CEC)*, p. 2955–2962. DOI : 10.1109/CEC.2014.6900644.

CHAWLA N. V., BOWYER K. W., HALL L. O. & KEGELMEYER W. P. (2002). Smote : Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, **16**, 321–357. DOI : 10.1613/jair.953.

DEREK DORAN S. S. G. (2016). An integrated method for real time and offline web robot detection. *Expert Systems*, **33**, 592–606. DOI : 10.1111/exsy.12184.

DORAN D. & GOKHALE S. S. (2011). Web robot detection techniques : overview and limitations. *Data Min. Knowl. Discov.*, **22**(1–2), 183–210. DOI : 10.1007/s10618-010-0180-z.

GAO Y., FENG Z., WANG X., SONG M., WANG X., WANG X. & CHEN C. (2023). Reinforcement learning based web crawler detection for diversity and dynamics. *Neurocomputing*, **520**, 115–128. DOI : 10.1016/j.neucom.2022.11.059.

GEENS N., HUYSMANS J. & VANTHIENEN J. (2006). Evaluation of web robot discovery techniques : A benchmarking study. In P. PERNER, Éd., *Advances in Data Mining. Applications in Medicine, Web Mining, Marketing, Image and Signal Mining*, p. 121–130, Berlin, Heidelberg : Springer Berlin Heidelberg.

GUERAR M., VERDERAME L., MIGLIARDI M., PALMIERI F. & MERLO A. (2021). Gotta captcha 'em all : A survey of 20 years of the human-or-computer dilemma. *ACM Comput. Surv.*, **54**(9). DOI : 10.1145/3477142.

ILIOU C., KOSTOULAS T., TSIKRIKA T., KATOS V., VROCHIDIS S. & KOMPATSIARIS I. (2022). Web bot detection evasion using deep reinforcement learning. In *Proceedings of the 17th International Conference on Availability, Reliability and Security*, ARES '22, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/3538969.3538994.

ILIOU C., KOSTOULAS T., TSIKRIKA T., KATOS V., VROCHIDIS S. & KOMPATSIARIS Y. (2019). Towards a framework for detecting advanced web bots. In *Proceedings of the 14th International Conference on Availability, Reliability and Security*, ARES '19, New York, NY, USA : Association for Computing Machinery. DOI : 10.1145/3339252.3339267.

JAGAT R. R., SISODIA D. S. & SINGH P. (2023). Web-s4ae : a semi-supervised stacked sparse autoencoder model for web robot detection. *Neural Comput. Appl.*, **35**(24), 17883–17898. DOI : 10.1007/s00521-023-08668-w.

JAGAT R. R., SISODIA D. S. & SINGH P. (2024). Exploiting web content semantic features to detect web robots from weblogs. *Journal of Network and Computer Applications*, **230**, 103975. DOI : 10.1016/j.jnca.2024.103975.

KWON S., KIM Y.-G. & CHA S. (2012). Web robot detection based on pattern-matching technique. *J. Inf. Sci.*, **38**(2), 118–126. DOI : 10.1177/0165551511435969.

KWON YEO I. & JOHNSON R. A. (2000). A new family of power transformations to improve normality or symmetry. *Biometrika*, **87**, 954–959.

LAGOPOULOS A. & TSOUMAKAS G. (2020). Content-aware web robot detection. *Applied Intelligence*, **50**, 4017 – 4028.

ROVETTA S., SUCHACKA G. & MASULLI F. (2020). Bot recognition in a web store : An approach based on unsupervised learning. *Journal of Network and Computer Applications*, **157**, 102577. DOI : 10.1016/j.jnca.2020.102577.

STEVANOVIC D., AN A. & VLAJIC N. (2012). Feature evaluation for web crawler detection with data mining techniques. *Expert Systems with Applications*, **39**(10), 8707–8717. DOI : 10.1016/j.eswa.2012.01.210.

SUCHACKA G., CABRI A., ROVETTA S. & MASULLI F. (2021). Efficient on-the-fly web bot detection. *Knowl. Based Syst.*, **223**, 107074. DOI : 10.1016/j.jnca.2020.102577.

TAN P.-N. & KUMAR V. (2002). Discovery of web robot sessions based on their navigational patterns. *Data Mining and Knowledge Discovery*, **6**(1), 9–35. DOI : 10.1023/A 1013228602957.

ZABIHIMAYVAN M., SADEGHI R., RUDE H. N. & DORAN D. (2017). A soft computing approach for benign and malicious web robot detection. *Expert Systems with Applications*, **87**, 129–140. DOI : 10.1016/j.eswa.2017.06.004.