Revue Ouverte d'Intelligence Artificielle
Brouillon d'article, version du 25 août 2026

Graphaméléon : capturer des traces de navigation Web avec des graphes de connaissances et caractériser les micro activités pour détecter des cyber-attaques

Lionel Tailhardat[a], Benjamin Stach[b], Eric Petit[c], Yoan Chabot[d], Raphaël Troncy[e], Thibault Ehrhart[f]

[a] Orange Research France
E-mail : lionel.tailhardat@orange.com

[b] Orange Research (anciennement) France
E-mail : benjaminstach.pro@gmail.com

[c] Orange Research France
E-mail : eric.petit@orange.com

[d] Orange Research France
E-mail : yoan.chabot@orange.com

[e] EURECOM France
E-mail : raphael.troncy@eurecom.fr

[f] EURECOM France
E-mail : thibault.ehrhart@eurecom.fr.

Résumé. — Les modèles comportementaux sont essentiels pour la détection d'anomalies ou d'actes malveillants sur des systèmes de télécommunication à travers le Web. Cependant, les données nécessaires ne sont pas toujours disponibles et une connaissance complète de la topologie des systèmes est nécessaire pour exploiter pleinement les inférences faites par ces modèles. Pour résoudre ce problème, nous proposons l'extension Web Graphaméléon et une représentation des traces de navigation sous forme de graphe de connaissances RDF en utilisant les ontologies UCO et NORIA-O.

Mots-clés. — Traces de navigation Web, Analyse du comportement des utilisateurs et des entités (UEBA), Analyse des processus, Graphe de connaissances.

1. Introduction

En même temps que les technologies de l'information et de la communication évoluent et posent de nouveaux défis, la cybercriminalité n'a cessée d'augmenter durant la dernière décennie. Détecter et diagnostiquer rapidement des anomalies sur les réseaux et systèmes d'information sont de fait devenus une préoccupation majeure

pour de nombreuses entreprises, notamment pour les gestionnaires de réseaux critiques et de grande envergure (téléphonie fixe et mobile, fourniture d'accès Internet, réseaux nationaux et internationaux d'échange de données). En cybersécurité, l'analyse du comportement des utilisateurs et des entités[(1)] correspond à un ensemble de techniques pour identifier et atténuer les menaces au niveau des éléments structurants des réseaux tels que les routeurs, serveurs ou applications à partir de données d'usage. Cela consiste typiquement à découvrir des motifs comportementaux nominaux (ou standards), tant au niveau des interactions entre les utilisateurs et les systèmes techniques qu'entre les éléments structurants eux-mêmes, et de s'en servir comme références pour alerter sur une utilisation potentiellement malveillante.

Une part importante des interactions utilisateurs-applications se fait désormais via une interface Web. Prenons l'exemple d'un scénario simple d'exploitation d'une vulnérabilité d'une application accessible via le Web[(2)] : après une phase de reconnaissance du système ciblé, l'attaquant accède directement à la page d'accueil de la plateforme de services, utilise une technique d'injection SQL[(3)] pour tromper le système d'authentification, exporte des données privées, puis quitte le service en naviguant directement vers une autre page Web. Analyser les interactions de l'utilisateur avec la plateforme, et ainsi détecter ce scénario, suppose l'analyse conjointe des journaux de l'application et du trafic réseau. Or les journaux peuvent être inaccessibles ou inutilisables en raison de problèmes de confidentialité ou de format. De même, le trafic réseau peut être chiffré ou inaccessible à la collecte. Ces deux aspects entraînent une perte des informations nécessaires pour qualifier le scénario d'attaque [2].

De nombreux outils de détection existent aujourd'hui dans le domaine de la cybersécurité, chacun se concentrant sur un type spécifique de source de données. Dans cet article, nous affirmons que la mise en œuvre simultanée de ces outils n'est pas suffisante pour une compréhension efficace des situations anormales, et qu'il est nécessaire d'utiliser un vocabulaire commun pour analyser les anomalies en associant les observables (journaux applicatifs, traces réseau, alertes des outils de détection) à la topologie du réseau. Dans cette optique, nous étendons le projet Dynagraph [57] (une approche combinant des outils de capture de traces avec une application Web pour un rendu graphique des données de navigation) afin d'apprendre des modèles d'activité interprétables sous forme de données liées : l'extension Web Graphaméléon collecte les traces d'activité de l'utilisateur (trafic réseau, interactions avec le navigateur Web) lors d'une session de navigation Web et sérialise ces données dans la syntaxe RDF selon le vocabulaire UCO [54]. Les données résultantes sont ensuite injectées dans un graphe de connaissances [26] pour former une base d'interprétation des traces d'activité au niveau sémantique pour diverses applications, notamment l'apprentissage de modèles d'activité sous la forme de réseaux de Petri et de réseaux bayésiens. Ces modèles d'activité peuvent ensuite être utilisés pour identifier des situations analogues par correspondance partielle et projection sur le graphe de connaissances et, sur la base de cette projection, obtenir des informations contextuelles en parcourant le graphe.

[(1)] "User and Entity Behavior Analytics" (UEBA) en Anglais.
[(2)] `https://attack.mitre.org/techniques/T1190/`
[(3)] `https://fr.wikipedia.org/wiki/Injection_SQL`

Le reste de ce document est organisé comme suit. En Section 2, nous présentons les travaux connexes du point de vue de la cartographie du Web, de la modélisation de l'activité et de la détection des anomalies. En Section 3, nous présentons notre approche pour capturer les connaissances à partir des traces de navigation Web. Cela implique une modélisation de l'activité en trois couches (HTTP, micro-activités, macro-activités) basée sur le vocabulaire UCO. Nous décrivons également le composant de collecte Graphaméléon, ainsi que l'utilisation des réseaux de Petri et des réseaux bayésiens pour la détection des anomalies dans les traces de navigation. Nos expériences et résultats sont présentés en Section 4. Enfin, nous concluons et abordons les travaux futurs en Section 5. Le code source de Graphaméléon est disponible sur `https://github.com/Orange-OpenSource/graphameleon`, ainsi que le jeu de données expérimentales sur `https://github.com/Orange-OpenSource/graphameleon-ds`.

2. Travaux connexes

Collecte et représentation des connaissances. La cartographie du Web [23] est une thématique de recherche visant la compréhension de la structure du Web et de ses utilisateurs. Les études du domaine portent sur des sujets variés tels que les méthodes de prétraitement des données [39, 64], les techniques d'identification des utilisateurs [35], les algorithmes de reconnaissance de session [12, 45], et les méthodes de découverte de motifs [15]. Du point de vue de l'analyse de l'activité, le concept de raisonnement basé sur les traces [18] guide la conception d'outils d'interprétation sémantique des artefacts de services numériques en suggérant l'utilisation de vocabulaires contrôlés et de modèles de données liées. Concernant la représentation des événements et des activités au sein de graphes de connaissances, divers modèles de données – tantôt génériques, tantôt spécifiques à un domaine d'application – sont disponibles : modélisation de processus (BBO [3], réseaux de Petri [22, 61], HTTPinRDF [31, 38]) ; analyse causale (FARO [50]) ; cybersécurité (UCO [54], MITRE D3FEND [28]) ; opérations réseau (NORIA-O [55]) ; villes intelligentes (iCity ActivityOntology [29]).

Détection d'anomalies et analyse des processus. Pour la détection d'anomalies, diverses approches ont été proposées autour d'un principe commun d'identification des écarts par rapport aux comportements normaux, notamment par des modèles statistiques [65, 51, 47], des techniques d'apprentissage automatique [66, 44] et des méthodes basées sur les graphes [42, 19]. Le domaine de l'analyse des processus se concentre sur l'extraction de modèles de processus à partir de journaux d'événements et sur l'analyse du flux réel des activités. Ces modèles fournissent des informations sur le comportement typique et la structure sous-jacente des processus de navigation Web. Des techniques de vérification de conformité [40] ont été développées dans l'exploration de processus pour comparer le comportement observé aux modèles de processus attendus et identifier les écarts.

Cependant, la sensibilité des techniques de vérification de conformité à des micro-variations des activités [56] indique qu'une approche hybride déterministe-statistique

est souhaitable. Les réseaux bayésiens sont pertinents pour l'angle statistique [5], car ils découvrent et formalisent des relations probabilistes entre les faits. Utilisés dans la santé, l'industrie et la gestion de la connaissance, ils formalisent un raisonnement rationnel en présence d'incertitude [4], adapté aux interactions homme-machine où la variabilité est grande. Ils construisent des modèles probabilistes de traces d'activité pour classifier et prédire en temps réel des comportements types, associés à une variable cible prenant un nombre fini de valeurs (normal, suspect, malveillant, etc.). Pour ce faire, différentes techniques ont été proposées, notamment basées sur une modélisation bayésienne naïve [33], reposant sur l'hypothèse d'indépendance conditionnelle des variables d'entrée étant donnée la variable cible. Cette hypothèse n'étant pas toujours valide, des améliorations ont été apportées via différentes méthodes, en particulier : la sélection de variables, la moyenne de modèles [34, 8, 9], ou plus récemment, la pondération de variables (classifieur bayésien naïf pondéré) comme développé dans les outils Khiops [43, 27]. Ces derniers sont adaptés au traitement d'un grand nombre de variables. Pour un nombre réduit d'attributs (moins de 100), une approche mathématique optimale est possible, sans l'hypothèse d'indépendance conditionnelle, avec un apprentissage au fil de l'eau. Dans la suite de cet article, nous utiliserons le système ABIT [48, 49] qui exploite la théorie bayésienne.

Positionnement. Nous étendons le concept de raisonnement basé sur les traces au domaine de la cartographie du Web en considérant l'utilisation des graphes de connaissances comme moyen de représenter les données de topologie du Web et les données d'utilisation de façon conjointe et cohérente. Nous abordons ainsi une nouvelle opportunité induite par l'émergence de modèles de données applicables dans les domaines des infrastructures réseaux (pour la description de systèmes hétérogènes) et de la cybersécurité (pour la description et la gestion des attaques et des risques). Nous supposons que, dans cette émergence, la communauté est désormais en mesure de répondre au besoin de corréler les informations d'usage du Web avec la description de la structure du Web lui-même, afin d'améliorer la compréhension et la conception de systèmes complexes tout en tenant compte du couple utilisateur-système. Par exemple, il est évident que (en particulier dans UCO), l'analyse des attaques et des vulnérabilités repose principalement sur des indicateurs de compromission par l'énumération d'artefacts issus de situations passées. Ces indicateurs ne sont cependant jamais corrélés avec la topologie des réseaux et des services, ni même avec l'organisation temporelle des artefacts, ce qui correspond à une description statique des situations anormales et met de côté la structure propre des activités (c.-à-d. la stratégie employée en rapport à la dynamique des événements). À cet égard, nous montrons notamment avec notre proposition comment incorporer le concept de traces de navigation dans l'ontologie UCO, ce qui permet de bénéficier simultanément des connaissances en cybersécurité et du contexte réseau enregistré par ailleurs (c.-à-d. par les analystes en cybersécurité et les opérateurs réseau, respectivement) via l'ontologie NORIA-O, tout en garantissant une représentation normalisée et homogène des données. De plus, nous étendons l'utilisation de l'analyse des processus et de la vérification de conformité aux graphes de connaissances, en capitalisant sur l'alignement de ces techniques avec les principes du raisonnement basé sur les traces.

3. Approche

L'approche proposée comporte trois parties, le but étant de réaliser une collecte de données dont le résultat permettra à la fois d'analyser les traces de navigation Web dans leur contexte réseau et d'apprendre des motifs d'activité. La première partie consiste à développer une modélisation sémantique des activités des utilisateurs sous forme de graphe de connaissances en réutilisant l'ontologie UCO (§ 3.1). La seconde partie concerne la conception de l'outil Graphaméléon, une extension de navigateur Web permettant de capturer les données de navigation et de les sérialiser en RDF (§ 3.2). La troisième partie porte sur l'intégration de Graphaméléon dans une chaîne de traitement de données (Figure 3.1) dont le principe est d'extraire des motifs d'activité, soit en utilisant les outils d'analyse des processus et une représentation sous la forme de réseaux de Petri (§ 3.3), soit par apprentissage et analyse d'un réseau bayésien (§ 3.4).

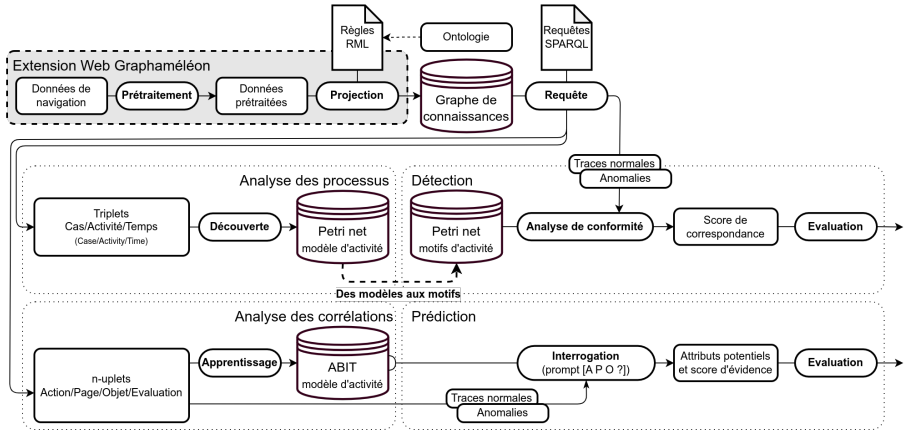

Figure 3.1 – Aperçu de la chaîne de traitement des données.
L'extension Web Graphaméléon capture et annote l'activité de l'utilisateur au niveau du navigateur Web. Un composant d'extraction des processus dérive des modèles d'activité places/transitions (P/T) à partir du graphe de connaissances RDF résultant. Ces modèles peuvent être utilisés pour construire une bibliothèque de motifs d'activité, qui sont ensuite utilisés par un composant de vérification de conformité pour classer de nouvelles traces d'activité comme des activités normales ou anormales. En parallèle des modèles P/T, un composant reposant sur l'algorithme ABIT encode les traces de navigation dans un réseau bayésien. Le réseau peut ensuite être interrogé afin de prédire la suite de traces partielles (activité en cours de réalisation) et alerter sur l'issue plausible (activité menant à une attaque).

3.1. Modélisation sémantique

Modélisation de l'activité des utilisateurs. Le concept d'activité manque de définition précise pour analyser la navigation sur le Web, car son interprétation repose fortement sur les données et l'échelle d'observation choisies. Il est en effet nécessaire de distinguer si l'identification d'une activité repose sur les interactions d'un utilisateur avec un site Web, ou si elle repose sur les échanges de paquets TCP entre le navigateur et le serveur portant ledit site Web. Pour commencer, supposons qu'une connexion HTTP est établie entre le navigateur Web de l'utilisateur et le serveur Web à partir d'une demande initiée par l'utilisateur. Le document demandé (typiquement une page

Web) nécessite, en règle générale, le chargement de ressources complémentaires telles que des médias, des feuilles de style ou des scripts. Ces dépendances impliquent un ensemble de sous-requêtes. Du point de vue de l'utilisateur, l'action consiste à naviguer vers un site Web par un clic sur un hyperlien (ou à accéder directement à la page via une URL), alors que du point de vue du navigateur Web, il s'agît d'une séquence de requêtes HTTP. De cette distinction, nous définissons deux niveaux de granularité pour discuter des traces de navigation. Celui nommé "micro-activité" correspond aux requêtes. Dans le niveau supérieur, nommé "macro-activité", nous considérons une trace comme étant un ensemble de requêtes et d'interactions. Par interaction, nous entendons toute action à l'initiative de l'utilisateur qui a une conséquence sur une page Web (un clic sur un hyperlien, renseigner un champ de formulaire, un clic sur un bouton du navigateur Web).

Projection sémantique. Les graphes de connaissances permettent de gérer de façon unifiée des données hétérogènes et issues de sources variées. Le fonctionnement type des navigateurs Web repose d'ores-et-déjà sur des normes et des protocoles établis. Par rapport à cette normalisation, nous considérons que l'apport stratégique des graphes de connaissances est de faciliter l'intégration de données provenant de sources extérieures au contexte du navigateur Web. Nous remarquons que l'ontologie UCO [54] semble bien adaptée à notre objectif car elle permet la représentation des activités de navigation Web à différentes échelles, en incluant des informations concernant les cycles d'actions, les actions individuelles, les connexions réseaux, les protocoles de communication, les ressources techniques utilisées, les noms de domaines Internet et les adresses IP.

Ainsi, notre stratégie pour construire le graphe de connaissances consiste à maximiser la réutilisation des concepts et propriétés définis dans UCO, et de faire correspondre les champs et les valeurs capturés au niveau du navigateur Web avec ces concepts et propriétés chaque fois que leur sémantique s'aligne. La Figure 3.2 illustre cette mise en œuvre en présentant le modèle de données. Les règles de construction de graphe correspondantes en syntaxe RML [20] sont disponibles dans le dépôt de code `https://github.com/Orange-OpenSource/graphameleon`.

Dans les détails, une requête HTTP est représentée par une entité de la classe `ucobs:HTTPConnectionFacet`, et ses en-têtes sont représentées par des propriétés spécifiques telles que `ucobs:startTime` et `ucobs:endTime` pour les horodatages, et `core:tag` pour les en-têtes de type Fetch Metadata [63]. Une adresse IP ou une URL pouvant être communes à plusieurs requêtes (par exemple un utilisateur répétant le même appel à un site Web, un site Web avec divers services hébergés sur le même serveur), ces éléments sont matérialisés par l'intermédiaire des classes `ucobs:IPAddressFacet` et `ucobs:URLFacet`, respectivement. Les références croisées entre les entités résultantes sont établies grâce à des propriétés telles que `ucobs:hasFacet` et `ucobs:host`. Pour les macro-activités, nous considérons les interactions de l'utilisateur comme des instances de la classe `ucoact:ObservableAction`, avec des relations vers les entités

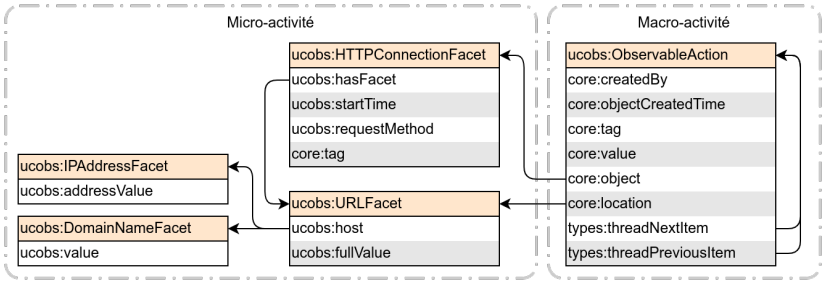

FIGURE 3.2 – Modèle de données.
Ce diagramme de classe définit les concepts et les propriétés pour la représentation sémantique des micro-activités et des macro-activités (§ 3.1). Les micro-activités décrivent précisément une séquence de requêtes capturées au niveau du navigateur Web. Les macro-activités améliorent la modélisation en décrivant les interactions. Les noms des concepts et des propriétés sont définis dans le vocabulaire UCO, avec les espaces de noms suivants : *core* = `https://ontology.unifiedcyberontology.org/uco/core#`, *ucobs* = `https://ontology.unifiedcyberontology.org/uco/observable#` et *types* = `https://ontology.unifiedcyberontology.org/uco/types#`.

`ucobs:HTTPConnectionFacet` et `ucobs:URLFacet` mentionnées ci-dessus pour décrire le contexte dans lequel elles se produisent. De plus, nous utilisons les propriétés `types:threadNextItem` et `types:threadPreviousItem` de UCO pour représenter la chronologie des traces d'activité.

3.2. Collecte de données avec Graphaméléon

Le navigateur Web étant l'interface principale entre un utilisateur et le Web, nous considérons pour la suite que la collecte de données doit porter à la fois sur les requêtes HTTP et les interactions utilisateur/navigateur pour comprendre et analyser pleinement le système utilisateur-réseau-application car ces deux ensembles reflètent l'intention directe et indirecte de l'utilisateur.

Collecte des requêtes. À son activation au sein du navigateur, l'outil Graphaméléon associe des fonctions de rappel aux processus d'envoi et de réception du navigateur. Cela permet d'intercepter toutes les requêtes gérées par le navigateur pour récupérer des informations à partir des en-têtes de celles-ci. La Table 3.1 résume le type de données collectées par Graphaméléon. Ces informations incluent les URLs, les adresses IP et les noms de domaines associés, l'horodatage de la requête et les Fetch Metadata[(4)]. Les Fetch Metadata nous permettent de déduire des connaissances indirectes à partir des traces de navigation. Par exemple, le champ `Sec-Fetch-Site` indique la relation entre l'initiateur de la requête et sa cible, fournissant ainsi des informations sur la topologie du réseau. De même, le champ `Sec-Fetch-Mode` aide à différencier les requêtes initiées par l'utilisateur de celles correspondant à des sous-requêtes pour charger des images et autres ressources. Enfin, nous tokenisons les URLs utilisées dans les requêtes en remplaçant tous les arguments présents par les noms de leurs paramètres respectifs. Cela permet d'abstraire d'éventuelles informations de contexte définies par les sites

[(4)] `https://www.w3.org/TR/fetch-metadata/`

Web, et éviter ainsi une diversité excessive dans l'interprétation des activités pour des cas similaires. Par exemple, les URLs `https://www.shop.com/?client_id=2313` et `https://www.shop.com/?client_id=346`, indépendamment de l'utilisateur initiant ces requêtes, reflètent le même comportement. Après tokenisation, ces URLs sont représentées par `https://www.shop.com/?client_id=[client_id]`.

Portée	Paramètre ou nom de l'en-tête HTTP	**Micro**	**Macro**
Requête	Method	✓	✓
	URL	✓	✓
	IP	✓	✓
	Domain	✓	✓
	Sec-Fetch-Dest	✓	✓
	Sec-Fetch-Site	✓	✓
	Sec-Fetch-User	✓	✓
	Sec-Fetch-Mode	✓	✓
Interaction	EventType	-	✓
	Element	-	✓
	Base URL	-	✓
Les deux	User-Agent	✓	✓
	Start time	✓	✓
	End time	✓	✓

TABLE 3.1 – Données collectées par Graphaméléon.
Types de données collectées par l'extension Web Graphaméléon en fonction du mode de capture (micro-activité *vs* macro-activité), et regroupées selon leur portée (requête *vs* interaction *vs* les deux).

COLLECTE DES INTERACTIONS. Afin de collecter les interactions entre l'utilisateur et le navigateur, l'outil Graphaméléon lie un "script de contenu" à chaque onglet actif du navigateur. Ces scripts associent des fonctions de rappel à tous les éléments interactifs des pages Web, tels que les hyperliens, les boutons, les formulaires, etc. Cette approche minimise l'impact de la collecte sur les performances du navigateur et évite de capturer des interactions indésirables, telles que des clics erronés sur des éléments non interactifs.

Afin d'identifier les interactions, nous prenons en compte le type d'événement enregistré, l'élément avec lequel l'utilisateur a interagi, et l'URL de la ressource correspondante. Lorsqu'un élément a un attribut *id*, définir une référence vers celui-ci est évident. Ce n'est cependant pas le cas général et il est donc nécessaire de construire une référence grâce à la position absolue de l'élément dans la hiérarchie du DOM[(5)] ; pour exemple, `body > maindiv[2] > div > div > a`. Bien que cette méthode permette de faire référence de manière déterministe aux éléments de la page, il est important de noter que les références sous forme de chemin hiérarchique sont difficiles à interpréter sans une capture de la page Web et des interactions car ces références portent peu d'information sur la finalité de l'élément. Une alternative pour générer ces références consisterait à injecter des attributs *id* dans les éléments de la page Web à l'aide de fonctions de rappel, mais cela ne résout pas le problème de la stabilité des références entre chaque session de navigation pour les pages ayant un contenu dynamique.

[(5)] `https://developer.mozilla.org/fr/docs/Web/API/Document_Object_Model`

3.3. Détection d'anomalies et réseaux de Petri

Trois familles de techniques de détection d'anomalies sont présentées dans [58] pour analyser des données de réseau représentées à l'aide d'un graphe de connaissances : *Model-Based Design*, où le graphe de connaissances contient les données nécessaires et suffisantes pour déduire les situations indésirables à l'aide de requêtes ; *Process Mining*, pour les situations liées à un modèle de décision et limitées dans le temps et l'espace, en utilisant des outils de vérification de conformité et une représentation des cas de détection sous forme de réseaux de Petri (réseaux P/T) ; *Statistical Learning* (apprentissage statistique) à l'aide de techniques de plongement de graphes [16] où les modèles d'anomalies (c.-à-d. une généralisation du contexte pour un ensemble de situations anormales) sont dérivés de la structure du graphe de connaissances.

Dans ce travail, nous nous concentrons d'abord sur l'approche du *Process Mining* (analyse des processus), en considérant que la collecte de données à l'aide de Graphaméléon correspond à des sessions de navigation Web relativement bien définies en termes de durée et d'activités : un seul utilisateur génère une trace d'activité capturée au niveau du navigateur Web, trace qui peut être directement annotée par l'utilisateur en termes de but de l'activité à la fin de la session de navigation Web. Nous postulons que les traces d'activité sont similaires à des modèles de décision, car la séquence d'actions de l'utilisateur lors d'une session de navigation (p.ex. cliquer sur un hyperlien, utiliser le bouton de retour du navigateur Web, remplir une zone de saisie) conditionne l'atteinte d'un objectif spécifique (c.-à-d. le but de l'activité) en fonction des informations présentées sur les pages Web. Nous supposons de même que les réseaux P/T sont une représentation adaptée pour analyser et catégoriser les traces de navigation car : *1)* ils possèdent une explicabilité intrinsèque par leur nature graphique ; *2)* les modèles de décision associés aux réseaux P/T peuvent se généraliser à différentes situations indépendamment de la représentation des connaissances sous-jacente ; *3)* les modèles de décision peuvent être facilement dérivés à partir de documents de spécifications produits par des experts métiers (p.ex. ingénieurs et techniciens réseaux), et implémentés sous forme de réseaux P/T à l'aide d'outils tels que TINA [6]. L'utilisation des réseaux P/T permet de tirer parti des techniques de détection d'anomalies couramment appelées "vérification de conformité", c.-à-d. d'évaluer la pertinence d'une trace par rapport à un modèle donné (score de correspondance) ou rejouer une trace à travers un modèle pour analyser les étapes incohérentes au sein de l'activité.

Dans ce qui suit, nous définissons deux concepts pour clarifier la notion d'anomalie par rapport à ce qui est observé et à ce qui est attendu du point de vue des activités. Tout d'abord, nous définissons un "modèle d'activité" comme la traduction de toute trace d'activité (obtenue lors de la phase de collecte de données) en une représentation de type réseau P/T à l'aide d'un algorithme de découverte de processus (process mining). Selon cette définition, les collectes de données réalisées avec l'extension Web Graphaméléon (§ 3.2) permettent aux utilisateurs d'établir un catalogue de modèles d'activité. Ensuite, nous définissons un "motif d'activité" comme un modèle universel d'activité représenté avec des réseaux P/T. Un motif est établi en se basant soit sur une spécification du comportement attendu du couple utilisateur-système pour une situation

spécifique, soit sur un comportement idéal dérivé de l'agrégation et du raffinement de plusieurs traces d'activité provenant du catalogue de modèles d'activité. Nous considérons que la gestion et la conversion des modèles d'activité en motifs relèvent de la responsabilité de l'utilisateur (analyse, sélection, raffinement), et dépassent le cadre de cet article.

La détection d'anomalies est donc définie par la comparaison d'un modèle d'activité à un motif d'activité. Ainsi, en supposant un "motif d'activité normale" (p.ex. l'authentification à une messagerie Web suivie d'une phase de consultation des e-mails), une mesure de correspondance inférieure à un seuil d'acceptation équivaut à détecter une situation anormale : $anormal \equiv correspondance_{\{alignement|rejeu\}}(modèle, motif) < \eta$, avec η un paramètre de seuil. Dans ce cas, nous pouvons déclencher une alerte, sans être pour autant en mesure de fournir plus de détails sur la nature de l'anomalie. En pratique, nous considérons qu'il est nécessaire de tester par rapport à un ensemble de "motifs d'activité anormaux" complémentaires dans une deuxième phase appelée phase de "qualification" afin de catégoriser l'anomalie.

3.4. Détection d'anomalies et réseaux bayésiens

L'observabilité des activités (§ 3.2) permet, grâce à l'accumulation des traces, de cartographier les chemins de navigation sur le Web et d'anticiper les étapes de navigation. Par exemple, dans le cas d'une attaque de site Web, il serait possible d'évaluer le risque d'une séquence d'activités futures de l'attaquant (n-uplets $\langle page, objet, action, deltaT \rangle$ potentiels) pour une session de navigation donnée, en fonction des activités déjà observées et avec une incertitude dépendant de la volonté de l'utilisateur, de la dynamique du Web et de la complétude de la carte.

Dans ce travail, nous utilisons l'algorithme ABIT[(6)] pour produire une modélisation probabiliste des séquences d'activité sur la base d'un réseau bayésien complet. ABIT met en œuvre une technique d'inférence adaptative combinant la théorie des réseaux bayésiens avec celle du filtrage numérique [32]. L'approche bayésienne non paramétrique (c.-à-d. sans loi de probabilité), s'appuyant sur le théorème de Bayes, détermine la classe la plus probable pour une nouvelle observation en se basant sur la distribution *a priori* des classes (la variable dépendante) et la distribution conditionnelle des données. La théorie du filtrage est exploitée pour construire des estimateurs adaptatifs de probabilités sur la base de fréquences relatives, chaque nouvelle observation venant mettre à jour la valeur des probabilités du réseau ; les distributions de probabilité sont ainsi estimées au fil de l'eau. Le critère de maximisation de la probabilité *a posteriori* minimise l'erreur de classification globale, offrant une robustesse face à l'évolution au cours du temps des lois statistiques des données ("data drift").

Dans ce schéma, les données d'activité sont modélisées par un échantillon de $(l + 1)$ variables aléatoires : $\{X_1, X_2, \cdots X_l, Y\}$, soit un enchaînement de l variables aléatoires explicatives et une variable dépendante Y (ou étiquette). Les données sont

(6) ABIT : "Adaptive Bayesian Inference Technique" [48, 49], un algorithme pour la classification adaptative multi-label, employé pour la modélisation de séquences d'actions.

transmises successivement à ABIT pour construire incrémentalement le modèle probabiliste, résultant en un réseau bayésien complet codant l'ensemble des dépendances statistiques entre les variables. Dans cette approche optimale, la probabilité conjointe d'une séquence se décompose de la sorte (Eq. 3.1) :

$$P(x_1, x_2, \cdots, x_l, y) = P(x_1) \times P(x_2|x_1) \times P(x_3|x_2, x_1) \times \cdots P(y|x_l, x_{l-1}, \cdots x_1) \qquad (3.1)$$

Différentes probabilités peuvent être extraites de ce réseau à tout moment, en particulier la probabilité *a posteriori* : $P(y|x_1, x_2, \cdots, x_l)$. Cette dernière fournit la probabilité de l'évaluation y étant donnée la séquence d'interaction : $x_1, x_2, \cdots x_l$. L'algorithme permet aussi de prédire la conclusion Y à partir d'une séquence partielle de prémisses (assimilée à un prompt), par exemple, les valeurs d'une micro-activité passée : x_1, x_2, x_3, x_4. Cela correspond à calculer la plausibilité $P(y|x_1, x_2, x_3, x_4)$ pour les différentes valeurs d'évaluation possibles.

En résumé, le modèle permet d'inférer l'étiquette du scénario à partir d'un jeu de prémisses complet ou partiel, et par conséquent d'anticiper la détection d'un événement non désirable (p.ex. attaque). La détection d'anomalies est donc définie en tant que prédiction $P(y \in Y_{\text{non désirables}}|x_1, x_2, \cdots x_l) > \eta$, avec η un paramètre de seuil. En complément, une anomalie est définie en tant qu'observation d'un ou plusieurs n-uplets (de type $\langle x_1, x_2, x_3, x_4 \rangle$) non prévus par le modèle.

4. Experimentations et résultats

Dans cette section, nous détaillons les expériences menées sur la base des approches décrites précédemment (§ 3), et présentons les résultats associés. Tout d'abord, nous analysons la corrélation entre le volume de triplets RDF générés par Graphaméléon et l'objet de sites Web visités (§ 4.1). Ensuite, nous modélisons et identifions trois scénarios de navigation Web en utilisant Graphaméléon au sein d'un environnement contrôlé (§ 4.2) en association avec des réseaux P/T (§ 4.3), puis avec des réseaux bayésiens (§ 4.4). Les expériences sont menées à l'aide de la version v2.1.0 de Graphaméléon. Les données associées à ces expériences sont disponible sur `https://github.com/Orange-OpenSource/graphameleon-ds`.

4.1. Trafic réseau et complexité des sites Web

Dans cette première expérience, nous cherchons à comprendre dans quelle mesure le comportement d'un site Web varie lors d'une première connexion et de fait, génère des indicateurs significatifs pour créer une signature du site utilisable ultérieurement pour la détection d'anomalies. Pour cela, nous étudions la relation entre la complexité *a priori* d'un ensemble de sites Web et les ressources téléchargées, en termes de nombre et de taille. Nous étudions cette complexité en mesurant le nombre de triplets RDF générés par Graphaméléon lors de la connexion initiale. La Table 4.1 présente les mesures enregistrées.

À notre connaissance, il n'existe actuellement aucune étude décrivant des groupes (clusters) de complexité de sites Web bien connus, sauf d'un point de vue marketing [60] (p.ex. secteur d'activité *vs* nombre moyen de connexions à la page d'accueil

du site, poids moyen de la page en octets, indice de vitesse de chargement). De plus, avec plus d'un milliard de sites Web référencés à ce jour [24], les outils d'analyse de sites Web proposent principalement des analyses de positionnement par rapport à la concurrence [46]. Cela souligne le défi de sélectionner des exemples représentatifs pour chaque groupe. Pour cette expérience, nous proposons d'établir un corpus de sites Web organisé selon trois groupes de complexité arbitraires. L'idée sous-jacente est que la complexité est liée au volume du contenu éditorial à afficher. Pour chaque catégorie, nous sélectionnons un sous-ensemble de trois sites Web de référence sur la base d'opinions d'experts tiers :

***One-Page*:** où "Swappa Bottle"[7], "Garden Studio"[8] et "Mark My Images"[9] (MMI) sont identifiés dans [41] comme les trois meilleurs exemples de sites Web d'une seule page dont s'inspirer dans le cadre de projets de conception de sites ;

***Encyclopedia*:** où "Encyclopedia Britannica Online"[10] (EBO), "Scholarpedia"[11] et "Encyclopedia.com"[12] sont présentés dans [13] comme les trois principales alternatives à Wikipédia du point de vue de la fiabilité de l'information ;

***Content-Heavy*:** où "RTI International"[13], "PrintMag"[14] et la "International Women's Media Foundation"[15] (IWMF) sont identifiés dans [25] comme les trois principaux sites Web présentant une grande quantité de contenu tout en offrant une expérience intuitive.

Ensuite, nous réalisons la collecte et l'analyse des données de traces de navigation pour chaque page d'accueil des sites Web de la manière suivante : *1)* dans une instance de Firefox sur ordinateur (anti-pistage $\in \{stricte, standard\}$), charger Graphaméléon et activer la capture de données (mode de collecte $\in \{micro, macro\}$, type de sortie $= semantize$) ; *2)* ouvrir un onglet de navigation et la console Network Monitor[16] (mise en cache = *désactivée*) ; *3)* accéder au site Web cible en saisissant son URL dans la barre de navigation ; *4)* arrêter la capture par Graphaméléon 10 secondes après la détection de l'événement de chargement complet de la page dans la console Network Monitor pour garantir l'exécution cohérente des scripts intégrés à la page Web (c.-à-d. l'événement `DOMContentLoaded`[17]) ; *5)* enregistrer les données dans un fichier (sérialisation = $Turtle$) ; *6)* recueillir les statistiques de collecte de données (nombre de requêtes, nombre de réponses, nombre d'interactions, nombre de sommets, nombre d'arêtes) à partir de l'interface utilisateur de Graphaméléon, ainsi que celles du graphe de connaissances résultant grâce à un ensemble de requêtes SPARQL (nombre de triplets, nombre de sujets, nombre d'instances de classe).

---

[7] `https://swappabottle.com/`
[8] `https://gardenestudio.com.br/`
[9] `https://www.markmyimages.com/`
[10] `https://www.britannica.com/`
[11] `http://www.scholarpedia.org/`
[12] `https://www.encyclopedia.com/`
[13] `https://www.rti.org/`
[14] `https://www.printmag.com/`
[15] `https://www.iwmf.org/`
[16] `https://firefox-source-docs.mozilla.org/devtools-user/network_monitor/`
[17] `https://developer.mozilla.org/en-US/docs/Web/API/Document/DOMContentLoaded_event`

Famille de site	Site Web	CM-Trk.	TC	SC	UDN	UHC	UIP	UURL
One-Page	Swappa Bottle	μ-Str.	n.a.	-	-	-	-	-
		μ-Std.	n.a.	-	-	-	-	-
		M-Str.	n.a.	-	-	-	-	-
	Garden Studio	μ-Str.	886	163	5	84	5	69
		μ-Std.	985	189	11	89	11	78
		M-Str.	21	5	1	1	1	1
	MMI	μ-Str.	427	81	3	38	3	37
		μ-Std.	423	80	3	38	3	36
		M-Str.	21	5	1	1	1	1
Encyclopedia	EBO	μ-Str.	599	122	13	54	13	42
		μ-Std.	2195	472	71	194	70	137
		M-Str.	21	5	1	1	1	1
	Scholarpedia	μ-Str.	452	111	4	55	4	48
		μ-Std.	579	143	11	64	11	57
		M-Str.	n.a.	-	-	-	-	-
	Encyclopedia	μ-Str.	350	66	2	31	2	31
		μ-Std.	1483	320	44	125	144	
		M-Str.	21	5	1	1	1	1
Content-Heavy	RTI	μ-Str.	381	76	6	33	6	31
		μ-Std.	562	118	14	48	14	42
		M-Str.	21	5	1	1	1	1
	PrintMag	μ-Str.	552	111	9	47	8	47
		μ-Std.	1143	234	25	101	24	84
		M-Str.	21	5	1	1	1	1
	IWMF	μ-Str.	362	72	5	31	5	31
		μ-Std.	388	78	6	33	6	6
		M-Str.	21	5	1	1	1	1

TABLE 4.1 – Expérimentation “Trafic réseau et complexité des sites Web".
Statistiques basées sur les modes de collecte “micro” (CM = μ) et “macro” (CM = M), et en fonction de la politique de blocage des traqueurs du navigateur Web. Abréviations : CM = mode de collecte, Trk. = politique de blocage des traqueurs (strict *vs* standard), TC = nombre de triplets, SC = nombre de sujets, UOA = nombre d'entités *ucobs :DomainNameFacet*, UDN = nombre d'entités *ucobs :DomainNameFacet*, UHC = nombre d'entités *ucobs :HTTPConnectionFacet*, UIP = nombre d'entités *ucobs :IPAddressFacet*, UURL = nombre d'entités *ucobs :URLFacet*, n.a. = non applicable.

	Stricte		**Standard**		**Std. / Str.**	
	UHC	UIP	UHC	UIP	UHC	UIP
One-Page	61,0	4,0	63,5	7,0	1,04	1,8
Encyclopedia	46,7	6,3	127,7	41,7	**2,73**	**6,6**
Content-Heavy	37,0	6,3	60,7	14,7	1,64	2,3

TABLE 4.2 – Moyenne du nombre d'entités en mode micro.
Comparaison de la moyenne du nombre d'entités UHC et UIP à partir de la Table 4.1 en fonction du niveau de complexité et de la politique anti-pistage. Seules les valeurs “Garden Studio” et “MMI” sont prises en compte pour la catégorie “One-Page". Abréviations : UHC = nombre d'entités *ucobs :HTTPConnectionFacet*, UIP = nombre d'entités *ucobs :IPAddressFacet*.

RÉSULTATS & DISCUSSION. En utilisant cette procédure, 27 échantillons de données ont été produits (trois groupes × trois sites × trois configurations du mode de collecte), dont 23 ont permis une analyse et quatre sont inexploitables (une erreur d'accès `SSL_ERROR_NO_CYPHER_OVERLAP` côté serveur pour “Swappa Bottle” en mode micro et macro, et une erreur de traitement indéterminée de l'extension Web pour “Scholarpedia” en mode macro). La Table 4.1 présente les statistiques relatives aux triplets RDF. Pour les échantillons issus du mode macro (CM = M), nous observons que les

statistiques sur les triplets RDF restent cohérentes quel que soit le site visité. Une analyse des fichiers Turtle résultants révèle également que la structure de données RDF est conforme au modèle de données de la Figure 3.2. En ce qui concerne le mode micro (CM = μ), les mesures présentent une variabilité significative entre chaque catégorie de complexité pour une politique d'anti-pistage donnée. La Table 4.2 permet de préciser ce point en présentant le nombre moyen d'entités pour les classes d'objets `ucobs:HTTPConnectionFacet` (UHC) et `ucobs:IPAddressFacet` (UIP), ce pour chaque scénario. La comparaison des valeurs moyennes du nombre d'entités en fonction de la politique d'anti-pistage (colonne "Std. / Str." dans la Table 4.2) révèle une augmentation du nombre moyen de connexions et de serveurs distants vers lesquels une connexion a été faite lorsque les politiques sont assouplies, et ce quel que soit le niveau de complexité. De ces mesures, nous concluons à la fois sur le bon fonctionnement de Graphaméléon et sur sa pertinence pour l'étude des primo-connexions. Bien que les groupes de complexité proposés puissent être discutés en raison de la taille limitée de l'échantillon et de la variabilité du contenu des sites Web, l'augmentation des échanges réseau en fonction des politiques d'anti-pistage fournit une base pour de futurs travaux de catégorisation par les stratégies de suivi mises en œuvre par les sites Web (tracking & analytics) et de la topologie de réseau associée.

4.2. Situation d'expérience contrôlée avec l'environnement LIVROS

Pour nos expérimentations de catégorisation de traces de navigation, nous avons développé Livros, une simulation de site Web de librairie en ligne accessible via `https://livros.tools.eurecom.fr/`. Basée sur Django[(18)], cette simulation offre des services comme l'authentification, un catalogue de livres, un formulaire de vente et un panier d'achat. Livros permet une exposition contrôlée à diverses vulnérabilités de sécurité, comme les attaques "cross-site scripting" (XSS) [37], et facilite l'étiquetage des éléments des pages Web, améliorant ainsi l'interprétabilité des données collectées.

Nous avons défini les *scénarios fondamentaux de navigation* suivants pour les expérimentations de catégorisation de traces de navigation (§ 4.3 & 4.4). Les deux premiers scénarios (normal et alternatif) ne modifient pas l'état de fonctionnement de Livros, tandis que le troisième scénario (attaque XSS) corrompt Livros, faisant passer son état de "clean" à "infected" et influençant ainsi les navigations normales subséquentes.

Navigation de base (normal) : un utilisateur accède au site Web, se connecte à son compte en utilisant son nom d'utilisateur et son mot de passe, navigue vers la page "Vendre un livre", renseigne un formulaire, puis retourne à la page d'accueil où il retrouve son livre dans la liste des ventes.

Navigation alternatif (normal alternatif) : un utilisateur accède au site Web, se connecte à son compte en utilisant un système à authentification unique (SSO), navigue vers la page "Vendre un livre", renseigne un formulaire, puis retourne à la page d'accueil où il retrouve son livre.

[(18)] `https://www.djangoproject.com/`

Navigation avec attaque XSS (anormal) : un attaquant accède au site Web et se connecte à son compte. Il navigue vers la page "Vendre un livre". Il remplit les champs requis et injecte un script malveillant dans le champ "Auteur", par exemple, `Kevin Mitnick<script src="http://104.18.2.24:8085/exploit"></script>`. Une fois le livre validé, les visiteurs de la page d'accueil activeront malgré eux le script malveillant, qui vole leurs cookies, du fait de l'affichage du livre avec l'auteur modifié. L'attaquant utilisera ces cookies pour usurper des identités et effectuer des achats frauduleux. Pour conclure son parcours, l'attaquant retourne à la page d'accueil.

Nous avons également défini les *scénarios de navigation par simulation* suivants pour compléter les évaluations avec des traces générées en nombre via l'outil Selenium[19]. Dans ces scénarios, le livre cible, "Javascript : The Definitive Guide", est toujours acheté, indépendamment du scénario utilisé ou de l'état de Livros ("clean" ou "infected").

full_shopping_session : l'utilisateur se connecte, parcourt trois à cinq catégories aléatoires, ajoute des livres au panier (incluant le livre cible), consulte son panier et finalise l'achat.

category_exploration : après connexion, l'utilisateur navigue dans les catégories, cherche le livre cible, l'ajoute au panier, consulte son panier et finalise l'achat.

random_walk : l'utilisateur se connecte, effectue dix actions aléatoires (scrolling, navigation, ajout de livre, visualisation du panier ou de la page compte, retour à l'accueil). A la fin, il s'assure d'ajouter le livre cible au panier et finalise l'achat.

4.3. Catégorisation de traces de navigation par réseaux de Petri

Dans cette expérience, notre objectif est de catégoriser les traces de navigation Web comme comportements normaux ou anormaux. Nous analysons les trois scénarios fondamentaux de l'environnement Livros (§ 4.3 : base *vs* alternatif *vs* attaque XSS) en utilisant la modélisation de macro-activité (§ 3.1) et les réseaux de Petri (§ 3.3), puis rendons compte de la capacité à identifier une anomalie.

Nous réalisons la collecte et l'analyse des données de traces de navigation pour chaque scénario de la manière suivante : *1)* dans une instance de Firefox sur ordinateur, charger Graphaméléon et activer la capture de données (mode de collecte = $macro$, type de sortie = $semantize$) ; *2)* ouvrir un onglet de navigation et parcourir le site Web simulé selon le scénario de navigation ; *3)* arrêter la capture par Graphaméléon et enregistrer les données dans un fichier (sérialisation = $Turtle$) ; *4)* recueillir les statistiques de collecte de données (nombre de requêtes, nombre de réponses, nombre d'interactions, nombre de sommets, nombre d'arêtes) à partir de l'interface utilisateur de Graphaméléon ; *5)* calculer le modèle d'activité à partir de la trace enregistrée en utilisant la bibliothèque PM4PY Process Mining [7] (méthode $\in \{Inductive, Alpha, LogSkeleton, Heuristic, AlphaPlus\}$) ; *6)* calculer la correspondance du modèle d'activité au motif de référence en utilisant la bibliothèque PM4PY (méthode $\in \{TokenBasedReplay, Alignment\}$). Le scénario de base, qui

[19] `https://www.selenium.dev/`

correspond au comportement “normal", est utilisé comme motif d’activité (c.-à-d. le modèle d’activité du scénario de base en tant que référence).

	Base	**Alternatif**	**Attaque XSS**
Requêtes	10	13	11
Interactions	18	14	18
Nœuds	263	283	277
Arcs	404	431	426

TABLE 4.3 – Expérimentation “catégorisation de traces de navigation".
Statistiques en termes du nombre de requêtes réseau, des interactions de l’utilisateur avec le navigateur Web, des nœuds et des arcs du graphe de navigation résultant, tel que rapporté par l’interface utilisateur de Graphaméléon pour les scénarios de navigation définis au § 4.3.

		Alternatif	**Attaque XSS**
Token-Based	Alpha	0,886	0,968
	Alpha+	0,890	0,969
	Inductive	0,923	1,000
	Heuristic	0,923	1,000
Alignement	Alpha	-	-
	Alpha+	-	-
	Inductive	0,718	0,976
	Heuristic	0,718	0,976
Log Skeleton		0,684	0,999

TABLE 4.4 – Scores de correspondance au motif de référence.
Comparaison des scores de correspondance au motif de référence (modèle d’activité du scénario de base) pour les modèles d’activité des scénarios “alternatif” et “attaque XSS”. Différentes techniques et algorithmes de vérification de conformité sont utilisés pour calculer les scores de correspondance. Les techniques “token-based” et “alignement” nécessitent une découverte préalable du modèle d’activité ; les algorithmes “Alpha/Alpha+”, “Inductive” et “Heuristic Miner” sont utilisés pour cela. La technique “Log Skeleton” fournit directement les scores de correspondance en utilisant les traces d’activité.

RÉSULTATS & DISCUSSION. En utilisant cette procédure, trois échantillons de données ont été produits. La Table 4.3 présente les statistiques relatives aux graphes de navigation résultants, et la Table 4.4 compare les résultats de différentes techniques d’évaluation de la correspondance au motif d’activité. Du point de vue des statistiques des graphes de navigation, nous observons que le scénario alternatif implique moins d’interactions mais plus de transactions réseau que pour le scénario de base. Cela correspond au fait que l’utilisateur n’a qu’un seul bouton à cliquer pour l’authentification, et que l’authentification est déléguée à diverses entités externes fournissant le service d’authentification. Pour le scénario “attaque XSS", le nombre d’interactions reste le même, mais le nombre de requêtes augmente d’une unité. Cela correspond à la séquence d’authentification identique à celle du scénario de base, mais avec une requête supplémentaire causée par l’injection SQL. Toujours pour ce même scénario, nous remarquons une légère variation dans les scores de correspondance (une correspondance moyenne de 98 % au motif d’activité), ce qui correspond également à la requête supplémentaire causée par l’injection SQL. Nous observons en outre que cette requête supplémentaire est facilement identifiable par l’utilisation des techniques d’alignement de séquences sur les traces sémantisées, le modèle de données proposé

en § 3.1 et appliqué au niveau de l'extension Graphaméléon permettant en effet de standardiser l'interprétation des traces.

Par conséquent, bien que notre approche fournisse une représentation formelle des traces de navigation, nous observons que son utilisation directe n'est pas adaptée à la détection d'anomalies lorsque des micro-changements se produisent par rapport à un motif d'activité (c.-à-d. lorsque les éléments de différenciation pour qualifier les écarts sont relativement rares dans la séquence). De même, nous remarquons que, bien que les algorithmes de découverte utilisent généralement plusieurs échantillons de traces pour générer un modèle généralisé de l'activité, nous avons dans notre cas considéré un motif parfait déduit d'une seule réalisation de trace. Or un modèle de comportement normal en situation réelle est potentiellement plus complexe. Par exemple, lors de la saisie d'un formulaire, l'ordre de lecture conventionnel est généralement suivi. Cependant, en raison de biais cognitifs, un utilisateur pourrait le remplir dans un ordre différent tout en restant dans les limites d'un comportement normal réel.

Enfin, en prenant du recul sur la collecte de données et le traitement sémantique, nous remarquons une faible compression lexicale des données de trace de navigation en raison d'un formatage cohérent (p.ex. l'URL de la requête est toujours située dans l'en-tête "url"). Cependant, cette compression concerne davantage la sémantique des interactions. En effet, l'un des défis de l'alignement des modèles d'activité réside dans le manque d'une méthode fiable pour identifier les éléments HTML (surtout en l'absence d'un ID explicite) à travers les navigateurs, les sessions et les utilisateurs. Ce défi devient apparent lorsque le DOM du contenu de la page change à chaque visite du site, en particulier lorsque des insertions publicitaires dynamiques se produisent.

4.4. Catégorisation de traces de navigation par réseaux bayésiens

Dans cette expérience, nous évaluons la catégorisation des traces de navigation via une modélisation probabiliste bayésienne (§ 3.4). Nous analysons trois scénarios (base, alternatif, attaque XSS) sur le même jeu de données utilisé pour évaluer l'approche par réseaux de Petri (§ 4.3) en supposant que les derniers n-uplets de la séquence suffisent à qualifier le comportement (hypothèse des "n-uplets dépendants sur séquence terminale courte"). Ensuite, nous effectuons une analyse de sensibilité en simulant de nombreux parcours de navigation sur l'environnement Livros (§ 4.2). Enfin, nous discutons de l'exploitation d'un modèle bayésien pour détecter des anomalies.

Pour la phase d'apprentissage du modèle, nous présentons les micro-activités des scénarios à l'algorithme ABIT à travers un ensemble de quatre attributs[(20)] : `Page` (identifiant de la page Web associée à l'activité) ; `Objet` (ucoact :ObservableAction ⊓ core :value.x), `Action` (ucoact :ObservableAction ⊓ core :tag.x) et `DeltaT` (le temps écoulé depuis la micro-activité précédente). `DeltaT` est discrétisé en trois niveaux $\in$ {Low, Medium, High} à partir des 33^e et 66^e centiles des valeurs sur l'ensemble du jeu de données.

[(20)]La formulation complète des attributs en logiques de description est disponible dans le dépôt `https://github.com/Orange-OpenSource/graphameleon-ds`

Résultats sur les scénarios fondamentaux. Nous retenons les cinq dernières micro-activités de chaque scénario et en concaténons les attributs en une séquence qui se termine par le nom du scénario. Ainsi, chaque portion de trace sera considérée comme une réalisation de l'échantillon $\{X_1, X_2, \cdots X_{20}, Y\}$, soit un enchaînement de vingt variables aléatoires explicatives (cinq quadruplets de quatre attributs) et une variable dépendante (étiquette). Trois séquences distinctes sont ainsi formées, correspondant aux trois scénarios. Elles sont transmises successivement à ABIT pour construire le modèle probabiliste. Ce jeu de données réduit est rejoué 10 000 fois, avec mélange aléatoire à chaque fois, afin d'obtenir une précision élevée sur les distributions de probabilité du réseau bayésien. Une fois l'apprentissage du réseau réalisé, nous interrogeons le modèle à partir d'une séquence de prémisses d'activités (ou prompt).

L'apprentissage automatique produit un modèle prédictif formé de 2059 paramètres[21], correspondant à l'ensemble des distributions de probabilité du réseau bayésien. En fournissant un prompt correspondant à un début de séquence, tel que `C1P8` pour désigner la `Page` d'index 8 associée à la variable X_1, le résultat de l'inférence est une liste de séquences ordonnée par ordre décroissant de probabilité conjointe (Eq. 4.1 – 4.3) :

$$\xrightarrow[0,33]{} C2O25 \xrightarrow[1,00]{} C3A4 \xrightarrow[1,00]{} C4Low \cdots \xrightarrow[0,99]{} C20Medium \xrightarrow[0,99]{} ATTACK \quad (-4\,\text{dB}) \quad (4.1)$$

$$\xrightarrow[0,67]{} C2O24 \xrightarrow[1,00]{} C3A2 \xrightarrow[0,50]{} C4High \cdots \xrightarrow[0,97]{} C20Low \xrightarrow[0,99]{} NORMAL \quad (-5\,\text{dB}) \quad (4.2)$$

$$\xrightarrow[0,67]{} C2O24 \xrightarrow[1,00]{} C3A2 \xrightarrow[0,50]{} C4Medium \cdots \xrightarrow[0,97]{} C20Low \xrightarrow[0,99]{} ALT \quad (-5\text{dB}) \quad (4.3)$$

Certaines transitions possèdent une probabilité conditionnelle proche de 1, traduisant une relation déterministe entre les attributs-valeurs, comme entre `C2O24` et `C3A2`. Pour chaque séquence prédite, la probabilité conjointe (en fin de chemin) est ici exprimée sur l'échelle des *évidences* en décibans (dB)[22] et s'obtient par la transformation $Ev_{dB} = 10 \times log\big(p/(1-p)\big)$. Dans le cas présent, étant donné que toutes ces séquences sont équiprobables, elles ont la même probabilité conjointe, soit en théorie −3 dB et en pratique légèrement moins du fait des erreurs d'approximations. Une représentation graphique peut être obtenue sous la forme d'un graphe orienté, codé en langage DOT[23], d'où est tirée la Figure 4.1.

Dans l'objectif d'évaluer la séquence le plus tôt possible – notamment afin d'anticiper une attaque – nous interrogeons le modèle à différentes étapes de l'interaction. Cela revient à calculer la probabilité *a posteriori* de chacune des évaluations possibles (`NORMAL`, `ALT`, `ATTACK`) pour chaque contexte d'interaction. Par exemple, le long du chemin `[C1P8, C2O24, C3A2, C4Medium]`, le scénario de détection consiste à évaluer $P(y|\texttt{C1P8})$, puis $P(y|\texttt{C1P8}, \texttt{C2O24})$, puis $P(y|\texttt{C1P8}, \texttt{C2O24}, \texttt{C3A2})$ et enfin $P(y|\texttt{C1P8}, \texttt{C2O24}, \texttt{C3A2}, \texttt{C4Medium})$ (Table 4.5 et Figure 4.1). Le résultat montre que le système ne peut rien prédire au départ, c.-à-d. à partir de `C1P8`, car l'inférence donne trois issues équiprobables. La connaissance de `C2O24` permet d'éliminer l'issue

[21] La complexité du modèle peut être estimée par sa borne inférieure théorique, qui correspond à $N \times (N+1)/2$ relations, où N est le nombre de variables explicatives. En pratique, ce nombre est plus élevé et dépend de l'entropie des données.

[22] Une probabilité de 50 % correspond à $Ev_{dB} = 0$ dB ; 10 contre 1 → 10 dB ; 1 contre 10 → −10 dB.

[23] `https://www.graphviz.org/doc/info/lang.html`

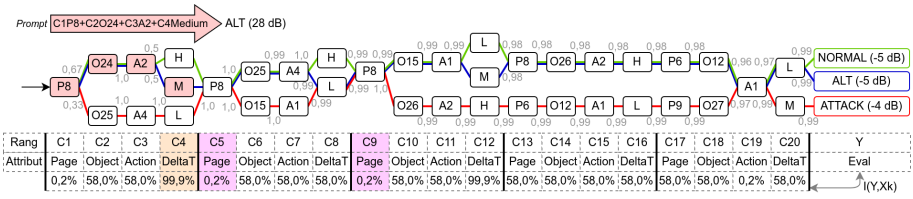

FIGURE 4.1 – Chemins plausibles à partir de X_1 = P8.
Ce graphe est construit par ABIT à partir de la prémisse X_1 = P8. Chaque nœud représente la valeur d'un attribut et chaque chemin une suite de valeurs prédites, ce qui correspond à la combinaison des séquences (Eq. 4.1) – (4.3). Sous chaque nœud figure le rang de la variable aléatoire (p.ex. $X_1 \equiv$ C1), le nom de l'attribut que la variable représente (p.ex. $X_1 \rightarrow$ Page) et l'information mutuelle $I(Y, X_k)$. Les fortes valeurs d'information mutuelle (p.ex. $I(Y, X_4) = 99{,}9\,\%$) font ressortir les variables déterminantes par rapport à l'issue probable de la séquence, comme ici la valeur `C4Medium`. On peut vérifier dans la Table 4.5, que la séquence `C1P8 C2O24 C3A2 C4Medium`, mise en valeur, conduit à anticiper avec une très forte probabilité le scénario `ALT`.

Prompt	**Base (Normal)**		**Alternatif**		**Attaque XSS**	
	Prob.	Ev [dB]	Prob.	Ev [dB]	Prob.	Ev [dB]
C1P8	0,333 299	−3,011	0,333 299	−3,011	0,333 299	−3,011
+C2O24	0,499 558	−0,008	0,499 558	−0,008		
+C3A2	0,499 308	−0,012	0,499 308	−0,012		
+C4Medium			**0,997 921**	26,814		

TABLE 4.5 – Plausibilité des scénarios en fonction des prémisses.
Cette table rend compte de la probabilité *a posteriori* de chaque scénario en fonction du prompt utilisé pour interroger le modèle. Le prompt est construit de façon incrémentale à partir de `C1P8`. Le prompt `C1P8 C2O24 C3A2 C4Medium` prédit l'issue `ALT` avec une certitude absolue (0,997 921), tandis que les prompts précédents ne permettent pas de lever l'ambiguité entre les issues.

`ATTACK`, mais il reste encore les deux possibilités `NORMAL` et `ALT`, toutes deux également probables. L'information `C3A2` ne lève toujours pas le doute. Enfin, sachant X_4 (valeur `C4Medium`) l'ambiguïté est levée : l'issue `ALT` est prédite, ici avec une certitude quasi absolue (probabilité de 0,997 921).

Cette modélisation probabiliste permet également, sur la base du réseau bayésien produit, de calculer à tout moment l'information mutuelle entre la variable dépendante et chacune des variables explicatives, conformément à son expression théorique :

$$I(Y, X_k) = \sum_{y \in Y, x \in X_k} p(y, x) \times log\left(\frac{p(y, x)}{p(y) \times p(x)}\right) \tag{4.4}$$

L'information mutuelle mesure la quantité d'information qui peut être obtenue au sujet d'une variable aléatoire par l'observation des valeurs d'une autre variable. Elle capture toute relation structurelle – linéaire ou non – entre deux variables, ce qui en fait un outil puissant pour évaluer la dépendance statistique entre elles. Elle est nulle si et seulement si les deux variables sont indépendantes, et elle croît lorsque la dépendance statistique augmente. Cette mesure peut être utilisée en vue d'identifier quelles variables sont déterminantes dans la discrimination des séquences, afin notamment de créer une signature d'attaque. Les valeurs $I(Y, X_k)$, reportées en Figure 4.1, font apparaitre les attributs qui déterminent l'évaluation finale, comme au niveau de la variable X_4 (`Delta-T`; valeur `C4Medium`). De même, elles font ressortir celles qui n'exercent

aucune influence, notamment là où les chemins se croisent, comme au niveau de X_5 ou X_9 (`C5` ou `C9`).

Résultats sur traces simulées. Pour mesurer la sensibilité de l'approche bayésienne, nous reprenons la démarche précédente en utilisant un jeu de données plus grand et varié, comprenant 3308 traces d'activité obtenues via un simulateur sur l'environnement Livros. Ces traces sont basées sur les parcours "full_shopping_session", "category_exploration" et "random_walk". Le jeu de données inclut 2646 traces "clean" et 662 traces "infected", soit un ratio de $4/1$ entre les deux situations. Le délai entre chaque action est aléatoire, compris entre 0,1 et 10 secondes. Pour simplifier le modèle bayésien, nous ne retenons que les dix dernières micro-activités de chaque trace, le simulateur étant configuré pour que l'action malveillante apparaisse dans le dernier tiers de la trace. Nous supposons également que l'attaque se concentre sur moins de cinq micro-activités. Ainsi, chaque trace est représentée par 40 variables explicatives, sans inclure la variable dépendante. Les traces sont transmises séquentiellement à l'algorithme ABIT. Une seule passe suffit à créer le modèle prédictif. Mais pour affiner l'estimation des probabilités et rendre l'apprentissage indépendant de l'ordre des traces, l'opération a été répétée 20 fois en mélangeant les données à chaque itération.

L'apprentissage du réseau ne prend que quelques secondes et aboutit à un modèle prédictif composé de 54 646 paramètres. Nous testons ce modèle avec quatre prompts (Table 4.6) issus du jeu de données afin de confirmer qu'il restitue des conclusions valides. Le premier test avec le prompt Pr_1 prédit une issue "clean" avec une confiance de 50 %, traduisant un bon niveau de connaissance, l'hypothèses alternative étant d'emblée écartée. Cette séquence conduit bien à une évaluation "clean" sans ambiguïté au niveau du jeu de données. Le test avec Pr_2 prédit également une issue "clean", mais avec une confiance très faible (-50 dB), reflétant un niveau de connaissance très faible, logiquement associé à une non-décision. Le test avec Pr_3 prédit les deux issues possibles, mais l'issue "clean" est plus plausible, avec un écart significatif d'environ 10 dB entre les deux prédictions. Le prompt Pr_4, similaire à Pr_3 jusqu'à la troisième micro-activité mais avec deux micro-activités supplémentaires, montre que ce complément d'information est suffisant pour écarter l'hypothèse "clean" et valider uniquement l'hypothèse "infected", malgré une plausibilité faible (-15 dB). Cette prédiction est là encore vérifiée dans les données.

Discussion. Le premier volet d'expérimentation de l'approche bayésienne, utilisant un jeu de données commun avec l'approche par réseaux de Petri, permet de produire un modèle prédictif à partir de peu de données et peu de valeurs communes entre scénarios. Bien que ce modèle ne soit pas adapté pour une mise en production, il établit les fondements de l'exploitation d'un modèle bayésien. En effet, l'extraction de séquences caractéristiques (Figure 4.1) et l'analyse de l'évolution de la plausibilité des scénarios (Table 4.5) valident l'hypothèse des "n-uplets dépendants sur séquence terminale courte" pour une prise de décision sur un horizon de temps court. De plus, l'observation de transitions tantôt déterministes, tantôt équiréparties, montre que c'est

Prompt	Clean		Infected	
	Prob.	Ev [dB]	Prob.	Ev [dB]
Pr_1	0,527 119	0,471		
Pr_2	0,000 012	−49,000		
Pr_3	0,265 586	−4,417	0,038 726	−13,948
$Pr_4 = Pr_3$ + ext.			0,028 145	−15,382

TABLE 4.6 – Plausibilité des scénarios sur les parcours simulés.
Cette table rend compte de la probabilité *a posteriori* de chaque scénario en fonction du prompt utilisé pour interroger le modèle. Les états "clean" et "infected" sont définis en § 4.2. Quatre prompts sont utilisés : Pr_1 = `P10-O22-T1-low P10-O118-T3-high P10-O22-T1-low`, Pr_2 = `P5-O32-T3-high P8-O18-T1-low P5-O9-T1-medium`, Pr_3 = `P7-O17-T3-high P8-O18-T1-low P5-O9-T1-high`, et $Pr_4 = Pr_3+$ `P5-O25-T3-high P5-O9-T1-medium`. Les prompts Pr_1, Pr_2 et Pr_3 décrivent trois micro-activités, et Pr_4 en décrit cinq.

l'attribut et sa position dans la séquence qui sont déterminants. L'information mutuelle confirme cette analyse. Le second volet d'expérimentation sur données simulées confirme l'utilité de l'approche bayésienne : il est possible de tester simultanément plusieurs hypothèses, en leur attribuant chacune un degré de confiance. Sur la base de ces mesures, il devient possible de prendre une décision robuste. Le degré de confiance n'a pas besoin d'être élevé si la prédiction mène à une seule hypothèse (Pr_4 en Table 4.6). Si plusieurs hypothèses sont en concurrence, l'écart des probabilités, mesuré sur l'échelle des évidences, fournit un critère fiable de discrimination (Pr_3 en Table 4.6).

5. CONCLUSIONS ET TRAVAUX FUTURS

Dans ce travail, nous avons cherché des moyens d'analyser les traces d'activités de navigation sur le Web dans le but de caractériser les activités des utilisateurs et le comportement des infrastructures réseaux. Les domaines d'application types envisagés dans cette recherche sont la gestion d'incident concernant les systèmes de télécommunication, la cybersécurité, et l'ingénierie des infrastructures réseaux. Sur les bases du projet DynaGraph [57], nous avons émis l'hypothèse que les graphes de connaissances peuvent structurer de manière adéquate les données collectées sur un navigateur Web au cours de sessions de navigation, cette structuration permettant une analyse avancée des traces de navigation à l'aide de modèles prédictifs.

Pour tester notre approche, nous avons développé les concepts de micro-activité et de macro-activité en rapport au vocabulaire UCO [54] pour la représentation sémantique des activités. Nous avons également mis au point l'outil Graphaméléon, une extension Web en open source disponible à l'adresse `https://github.com/Orange-OpenSource/graphameleon`, permettant la collecte en direct de données au niveau du navigateur Web et la sémantisation des traces de navigation. Enfin, nous avons analysé des traces d'activité collectées via Graphaméléon selon un plan expérimental en trois parties. Nous avons montré, dans l'expérience d'analyse de trafic par famille de complexité des sites Web, que l'augmentation des volumes d'échanges est fonction des politiques d'anti-pistage et fournit une base de travail intéressante pour la catégorisation des sites selon les stratégies d'analyse d'audience employées et la

topologie de réseau associée. Ensuite, avec l'expérience de catégorisation des traces de navigation par réseaux de Petri, nous avons montré les limites de la technique de vérification de conformité pour la détection d'anomalies lorsque des micro-changements se produisent par rapport à un motif de référence. Nous avons également remarqué le défi que représente l'harmonisation des modèles d'activité en raison de l'absence d'une méthode fiable pour identifier les éléments HTML au sein des navigateurs Web, notamment par comparaison entre sessions de navigation ou entre utilisateurs. Enfin, avec l'expérience de catégorisation par modèle bayésien, nous avons montré qu'un modèle probabiliste permet d'absorber la variabilité des interactions, tout en gérant l'incertitude avec rigueur : pour un site Web donné, un tel modèle permet de condenser une connaissance empirique sur le comportement attendu des utilisateurs ou du site, d'en mesurer les écarts sur une échelle objective et de les classifier automatiquement. Son apprentissage requiert toutefois de pouvoir collecter une quantité d'information suffisamment représentative de l'usage du site, de la traduire dans une syntaxe attribut-valeurs simple et de pouvoir la qualifier lors d'une phase d'évaluation. L'utilisation de l'algorithme ABIT [48, 49] en simplifie le processus, car le modèle se construit de façon dynamique à partir d'un flux de données d'entrée, en étant "au mieux" à tout moment. Cela signifie que la phase d'inférence peut démarrer très tôt, en même temps que se poursuit son apprentissage. En pratique, le modèle pourrait être interrogé au fil de l'eau à partir d'un contexte d'interaction courant. Ce contexte serait formé d'un nombre limité de micro-activités – par exemple entre cinq et dix – jugé suffisant pour produire des inférences fiables et pouvoir détecter le cas échant une action anormale ou malveillante.

Sur la base de ces développements et résultats, nous envisageons des travaux futurs approfondissant les aspects de la cartographie du Web, de l'analyse du comportement du couple utilisateur-système, et de la détection d'anomalies. En ce qui concerne l'*outil Graphaméléon*, des aspects techniques spécifiques nécessitent des développements complémentaires, tels que la génération de graphe en flux, l'annotation des activités via l'interface utilisateur et la gestion simultanée de plusieurs sessions de navigation Web. Concernant la *détection d'anomalies*, nous envisageons d'hybrider les approches par réseaux de Petri et modèles bayésiens. Cela pourrait réduire la sensibilité des premiers (p.ex. en utilisant la mesure de plausibilité) ou étendre le domaine d'analyse des seconds pour des données hors du jeu d'apprentissage (p.ex. en utilisant les capacités de simulation des réseaux de Petri). Un autre axe important consiste à lever l'hypothèse des n-uplets sur séquence terminale courte pour capturer des signatures d'anomalies nécessitant un horizon de prédiction plus large. Enfin, concernant les *graphes de connaissances*, nous envisageons également d'intégrer les modèles d'activité dans un graphe RDF [22, 61, 14] afin de calculer des contextes d'anomalies enrichis par un processus de décision pour des applications d'aide à la décision utilisant la technique de plongement de graphes [58].

Bibliographie

[1] W. Aalst, A. Weijters & L. Măruşter, « Workflow Mining : Discovering Process Models from Event Logs », *IEEE Transactions on Knowledge and Data Engineering* (2004).

[2] I. Akbari, M. A. Salahuddin, L. Ven, N. Limam, R. Boutaba, B. Mathieu, S. Moteau & S. Tuffin, « Traffic Classification in an Increasingly Encrypted Web », *Communications of the ACM* (2022).

[3] A. Annane, N. Aussenac-Gilles & M. Kamel, « BBO : BPMN 2.0 Based Ontology for Business Process Representation », in *20th European Conference on Knowledge Management (ECKM)* (Lisbon, Portugal), 2019.

[4] V. Barra, A. Cornuéjols & L. Miclet, *Apprentissage artificiel : concepts et algorithmes de Bayes et Hume au deep learning*, 4ème éd., Algorithmes, Éditions Eyrolles, Paris, 2021.

[5] A. Becker & P. Naïm, *Les réseaux bayésiens modèles graphiques de connaissance*, Eyrolles, Paris, 1999.

[6] B. Berthomieu, P.-O. Ribet & F. Vernadat, « The tool TINA – Construction of abstract state spaces for petri nets and time petri nets », *International Journal of Production Research* (2004).

[7] A. Berti, S. van Zelst & W. van der Aalst, « Process Mining for Python (pm4py) : Bridging the Gap between Process-and Data Science », in *Proceedings of the ICPM Demo Track 2019, co-located with 1st International Conference on Process Mining (ICPM)*, 2019.

[8] M. Boullé, « Khiops : A Statistical Discretization Method of Continuous Attributes », *Machine Learning* **55** (2004), n° 1, p. 53-69.

[9] ———, « Compression-Based Averaging of Selective Naive Bayes Classifiers », *J. Mach. Learn. Res.* **8** (2007), p. 1659–1685.

[10] Z. D. Bozorgi, I. Teinemaa, M. Dumas, M. La Rosa & A. Polyvyanyy, « Process Mining Meets Causal Machine Learning : Discovering Causal Rules from Event Logs », `https://arxiv.org/abs/2009.01561`, 2020.

[11] M. Bush, « When it comes to mobile, it's time to stop making excuses », `https://www.thinkwithgoogle.com/marketing-strategies/app-and-mobile/mobile-shopping-ecosystem/`, 2018, Accessed : 2023-08-10.

[12] M. C. Calzarossa & L. Massari, « Analysis of Header Usage Patterns of HTTP Request Messages », in *IEEE International Conference on High Performance Computing and Communication*, 2014.

[13] K. Campbell, « Seven Free Wikipedia Alternatives », `https://blog.reputationx.com/wikipedia-alternatives`, 2023, Accessed : 2023-08-10.

[14] V. A. Carriero, M. Scrocca, I. Baroni, A. Azzini & I. Celino, « Procedural Knowledge Ontology (PKO) », in *The Semantic Web*, vol. 15719, Springer Nature Switzerland, Cham, 2025, p. 334-350.

[15] G. Castellano, A. M. Fanelli & M. A. Torsello, « Web Usage Mining : Discovering Usage Patterns for Web Applications », Springer Berlin Heidelberg, 2013.

[16] I. Chami, S. Abu-El-Haija, B. Perozzi, C. Ré & K. Murphy, « Machine Learning on Graphs : A Model and Comprehensive Taxonomy », `https://arxiv.org/abs/2005.03675`, 2020.

[17] P.-A. Champin, A. Mille & Y. Prié, « Vers des traces numériques comme objets informatiques de premier niveau », *Intellectica – La revue de l'Association pour la Recherche sur les sciences de la Cognition (ARCo)* (2013).

[18] A. Cordier, M. Lefevre, P.-A. Champin, O. Georgeon & A. Mille, « Trace-Based Reasoning – Modeling Interaction Traces for Reasoning on Experiences », in *The 26th International FLAIRS Conference*, 2013.

[19] P. Dagnely, T. Ruette, T. Tourwé & E. Tsiporkova, « Ontology-driven multilevel sequential pattern mining : mining for gold in event logs of photovoltaic plants », in *International Conference on Intelligent Systems (IS)*, 2018.

[20] A. Dimou, M. Vander Sande, P. Colpaert, R. Verborgh, E. Mannens & R. van de Walle, « RML : A Generic Language for Integrated RDF Mappings of Heterogeneous Data », in *Proceedings of the Workshop on Linked Data on the Web, LDOW 2014, co-located with the 23rd International World Wide Web Conference (WWW 2014)*, vol. 1184, CEUR-WS.org, 2014.

[21] A. Echraibi, J. Flocon-Cholet, S. Gosselin & S. Vaton, « An Infinite Multivariate Categorical Mixture Model for Self-Diagnosis of Telecommunication Networks », in *23rd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, 2020, p. 258-265.

[22] D. Gašević & V. Devedžić, « Petri Net Ontology », *Knowledge-Based Systems* (2006).
[23] F. Ghitalla, D. Boullier & M. Jacomy, *Qu'est-ce que la cartographie du Web ? : expéditions scientifiques dans l'univers des données numériques et des réseaux*, 2021.
[24] K. Haan, « Top Website Statistics For 2023 », `https://www.forbes.com/advisor/business/software/website-statistics/`, 2023, Accessed : 2023-08-10.
[25] L. Held, « Examples of Content Heavy Editorial Website Designs », `https://www.newmediacampaigns.com/blog/best-examples-of-content-heavy-editorial-website-designs`, 2021, Accessed : 2023-08-10.
[26] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. de Melo, C. Gutierrez, J. E. Labra Gayo, S. Kirrane, S. Neumaier, A. Polleres, R. Navigli, A.-C. Ngonga Ngomo, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab & A. Zimmermann, « Knowledge Graphs », 2020.
[27] C. Hue & M. Boullé, « Fractional Naive Bayes (FNB) : non-convex optimization for a parsimonious weighted selective naive Bayes classifier », `https://arxiv.org/abs/2409.11100`, 2024.
[28] P. E. Kaloroumakis & M. J. Smith, « Toward a Knowledge Graph of Cybersecurity Countermeasures », Technical report, The MITRE Corporation, 2021.
[29] M. Katsumi & M. Fox, « iCity Transportation Planning Suite of Ontologies », Technical report, University of Toronto, 2020.
[30] G. Khodabandelou, C. Hug, R. Deneckère & C. Salinesi, « Process Mining Versus Intention Mining », 2013.
[31] J. Koch, C. A. Velasco & P. Ackermann, « HTTP Vocabulary in RDF 1.0 », W3C Working Group note, W3C, 2017.
[32] M. Kunt, « Traitement de l'information. Vol. 1 : Techniques modernes de traitement numérique des signaux » (Murat Kunt, éd.), Collection électricité, Presses Polytechniques et Univ. Romandes, Lausanne, 1ère éd., 1991.
[33] P. Langley, W. Iba & K. Thompson, « An analysis of Bayesian classifiers », in *Proceedings of the Tenth National Conference on Artificial Intelligence*, AAAI'92, AAAI Press, 1992, p. 223–228.
[34] P. Langley & S. Sage, « Induction of selective Bayesian classifiers », in *Proceedings of the Tenth International Conference on Uncertainty in Artificial Intelligence* (San Francisco, CA, USA), UAI'94, Morgan Kaufmann Publishers Inc., 1994, p. 399–406.
[35] P. Laperdrix, N. Bielova, B. Baudry & G. Avoine, « Browser Fingerprinting : A survey », 2019.
[36] N. Lazzari, A. Poltronieri & V. Presutti, « Classifying Sequences by Combining Context-Free Grammars and OWL Ontologies », in *The Semantic Web*, Springer Nature, 2023.
[37] B. Leonard, « Ukraine Remains Russia's Biggest Cyber Focus in 2023 », `https://blog.google/threat-analysis-group/ukraine-remains-russias-biggest-cyber-focus-in-2023/`, 2023.
[38] M. Lirzin & B. Markhoff, « Vers Une Ontologie Des Interactions HTTP », in *31èmes Journées Francophones d'Ingénierie des Connaissances* (Angers, France), 2020.
[39] V. S. Lopes & J. Mendes-Moreira, « A Comparative Analysis of Data Preprocessing Techniques in Web Usage Mining », in *IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, IEEE, 2019.
[40] J. Munoz-Gama, « Conformance Checking and Diagnosis in Process Mining : Comparing Observed and Modeled Processes », Thèse, Universitat Politècnica de Catalunya – BarcelonaTech, Barcelona, 2014.
[41] M. Murali, « 11 Examples of One-Page Websites to Inspire You », `https://blog.hubspot.com/website/11-examples-of-one-page-websites-for-inspiration`, 2023, Accessed : 2023-08-10.
[42] C. C. Noble & D. J. Cook, « Graph-Based Anomaly Detection », in *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM.
[43] Orange-Labs, « Khiops : Orange Innovation's autoML suite v10 », `https://khiops.org`, 2021.
[44] G. Pang, C. Shen, L. Cao & A. van den Hengel, « Deep Learning for Anomaly Detection », *ACM Computing Surveys* (2021).

[45] H. Park & D.-K. Baik, « Web Log Session Identification Based on Cluster-Based Classification », in *7th International Conference on Advanced Information Networking and Applications (AINA)*, IEEE, 2011.

[46] J. Parsons, « Alexa.com Is Dead – Here Are 20 of the Best Alternatives », `https://www.contentpowered.com/blog/alexa-com-dead-alternatives/`, 2023, Accessed : 2023-08-10.

[47] S. Pauwels & T. Calders, « Extending Dynamic Bayesian Networks for Anomaly Detection in Complex Logs », 2018.

[48] E. Petit & D. Chêne, « Navigation adaptative dans les systèmes interactifs : paradigme et solution », in *Proceedings of the 17th "Ergonomie et Informatique Avancée" Conference* (Bidart France), ACM, 2021, p. 1-8.

[49] ———, « Apprentissage automatique robuste et continu des habitudes d'usage pour adapter les interfaces numériques aux besoins des utilisateurs », `https://hal.science/hal-05204331`, 2025.

[50] Y. Rebboud, P. Lisena & R. Troncy, « Beyond Causality : Representing Event Relations in Knowledge Graphs », in *Knowledge Engineering and Knowledge Management*, Springer International, 2022.

[51] S. Saqaeeyan, H. Haj Seyyed Javadi & H. Amirkhani, « Anomaly Detection in Smart Homes Using Bayesian Networks », *KSII Transactions on Internet and Information Systems* (2020).

[52] D. Sola, H. van der Aa, C. Meilicke & H. Stuckenschmidt, « Activity Recommendation for Business Process Modeling with Pre-Trained Language Models », in *The Semantic Web*, Springer Nature, 2023.

[53] D. Sola, C. Warmuth, B. Schäfer, P. Badakhshan, J.-R. Rehse & T. Kampik, « SAP Signavio Academic Models : A Large Process Model Dataset », `https://arxiv.org/abs/2208.12223`, 2022.

[54] Z. Syed, A. Padia, M. L. Mathews, T. Finin & A. Joshi, « UCO : A Unified Cybersecurity Ontology », in *AAAI Workshop on Artificial Intelligence for Cyber Security*, AAAI Press, 2016.

[55] L. Tailhardat, Y. Chabot & R. Troncy, « NORIA-O : an Ontology for Anomaly Detection and Incident Management in ICT Systems », in *Semantic Web – 21st International Conference, ESWC 2024, Hersonissos, Crete, Greece, May 26 - 30, 2024, Proceedings*, 2024.

[56] L. Tailhardat, B. Stach, Y. Chabot & R. Troncy, « Graphaméléon : Apprentissage Des Relations et Détection d'anomalies Sur Les Traces de Navigation Web Capturées Sous Forme de Graphes de Connaissances », in *35èmes Journées Francophones d'Ingénierie des Connaissances (IC 2024) @ Plate-Forme Intelligence Artificielle (PFIA 2024)* (La Rochelle, France), 2024.

[57] L. Tailhardat, R. Troncy & Y. Chabot, « Walks in Cyberspace : Towards Better Web Browsing and Network Activity Analysis with 3D Live Graph Rendering », Association for Computing Machinery, 2022.

[58] ———, « Leveraging Knowledge Graphs For Classifying Incident Situations in ICT Systems », in *18th International Conference on Availability, Reliability and Security (ARES)*, 2023.

[59] A. Terragni & M. Hassani, « Optimizing Customer Journey Using Process Mining and Sequence-Aware Recommendation », in *34th ACM/SIGAPP Symposium on Applied Computing*, 2019.

[60] thinkwithgoogle.com, « Find Out How You Stack Up to New Industry Benchmarks for Mobile Page Speed », `https://think.storage.googleapis.com/docs/mobile-page-speed-new-industry-benchmarks.pdf`, 2017, Accessed : 2023-08-10.

[61] J. C. Vidal, M. Lama & A. Bugari n, « A High-level Petri Net Ontology Compatible with PNML », (2006).

[62] W3C, « Level 1 Document Object Mode Specificationl », Working draft, W3C, July 1998.

[63] ———, « Fetch Metadata Request Headers », Working draft, W3C, July 2021.

[64] X. Wu, X. Zhu, G. Wu & W. Ding, « Data mining with big data », *IEEE Transactions on Knowledge and Data Engineering* (2004).

[65] N. Ye, « A Markov Chain Model of Temporal Behavior for Anomaly Detection », in *Proceedings of the 2000 IEEE Workshop on Information Assurance and Security*, 2000.

[66] Y. Zhao, L. Deng, X. Chen, C. Guo, B. Yang, T. Kieu, F. Huang, T. B. Pedersen, K. Zheng & C. S. Jensen, « A Comparative Study on Unsupervised Anomaly Detection for Time Series : Experiments and Analysis », `https://arxiv.org/abs/2209.04635`, 2022.

ABSTRACT. — Behavioral models are essential for detecting anomalies or malicious activities on telecommunications systems occurring through the Web. However, the necessary data is not always available, and a complete understanding of the system's topology is required to fully exploit the inferences made by these models. To address this issue, we propose the Graphameleon Web extension and a representation of navigation traces in the form of an RDF knowledge graph using the UCO and NORIA-O ontologies.

KEYWORDS. — Web Browsing Traces, User and Entity Behavior Analytics (UEBA), Process Mining, Conformance Checking, Bayesian Networks, Knowledge Graph.

RESUMEN. — Los modelos de comportamiento son esenciales para la detección de anomalías o actos malintencionados en sistemas de telecomunicaciones a través de la Web. Sin embargo, los datos necesarios no siempre están disponibles y se requiere un conocimiento completo de la topología de los sistemas para optimizar al máximo las inferencias realizadas por estos modelos. Para resolver este problema, proponemos una extensión Web llamada Graphameleon y un grafo de conocimiento para representar las trazas de navegación utilizando las ontologias UCO y NORIA-O.

PALABRAS CLAVES. — Trazas de Navegación Web, Análisis de Comportamiento de Usuarios y Entidades (UEBA), Minería de Procesos, Verificación de Conformidad, Red Bayesiana, Grafo de Conocimiento.