# Graphaméléon : apprentissage des relations et détection d'anomalies sur les traces de navigation Web capturées sous forme de graphes de connaissances

L. Tailhardat[1,3], B. Stach[2], Y. Chabot[1], R. Troncy[3]

[1] Orange, France
[2] UTBM, Belfort, France
[3] EURECOM, Sophia-Antipolis, France

lionel.tailhardat@orange.com ; benjaminstach.pro@gmail.com ; yoan.chabot@orange.com ; raphael.troncy@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.*

## Abstract

*Behavioral models are essential for detecting anomalies or malicious activities on telecommunications systems occuring 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, Knowledge Graph.*

# 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 (p.ex. routeurs, serveurs, 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 l'Internet [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 [13].

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,

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

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 (p.ex journaux applicatifs, traces réseau, alertes des outils de détection) à la topologie du réseau. Dans cette optique, nous étendons le projet Dynagraph [23] (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 [40]. Les données résultantes sont ensuite injectées dans un graphe de connaissances [1] pour interpréter les traces d'activité à un niveau sémantique et dériver des motifs, notamment sous forme de réseaux de Petri. Ces modèles d'activité peuvent ensuite être utilisés, du côté utilisateur ou du côté réseaux, pour identifier des situations analogues en les projetant sur le graphe de connaissances et, sur la base de cette projection, obtenir des informations contextuelles en parcourant le graphe.

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 et l'utilisation des réseaux de Petri 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 [12] 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 [26, 37], les techniques d'identification des utilisateurs [21], les algorithmes de reconnaissance de session [8, 31], et les méthodes de découverte de motifs [9]. Du point de vue de l'analyse de l'activité, le concept de raisonnement basé sur les traces [3] 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 [4], réseaux de Petri [11, 18], HTTPinRDF [16, 28]) ; analyse causale (FARO [39]) ; cybersécurité (UCO [40], MITRE D3FEND [33]) ; opérations réseau (NORIA-O [25]) ; 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 [38, 34, 32], des techniques d'apprentissage automatique [41, 30] et des méthodes basées sur les graphes [7, 10]. 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 [17] ont été développées dans l'exploration de processus pour comparer le comportement observé aux modèles de processus attendus et identifier les écarts.

**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. Pour 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 (i.e. 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 (i.e. par les analystes en cybersécurité et les opérateurs réseau, respectivement) via l'ontologie NORIA-O, tout en garantissant une représentation nor-

malisée 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 des 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 1) dont le principe est d'extraire des motifs d'activité en utilisant les outils d'analyse des processus et une représentation sous la forme de réseaux de Petri (§3.3).

## 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 (p.ex. une page Web) nécessite, en règle générale, le chargement de ressources complémentaires telles que des images, des scripts ou autres. 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. 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 (p.ex. clic sur un hyperlien, renseigner un champ de formulaire, 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 [40] 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/propriétés définis dans UCO, et de faire correspondre les champs et les valeurs capturés au niveau du navigateur Web avec ces concepts/propriétés chaque fois que leur sémantique s'aligne. La Figure 2 illustre cette mise en œuvre en présentant le modèle de données. Les règles de construction de graphe correspondantes en syntaxe RML [5] 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 [36]. Une adresse IP ou une URL pouvant être communes à plusieurs requêtes (p.ex. 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 `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 des 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

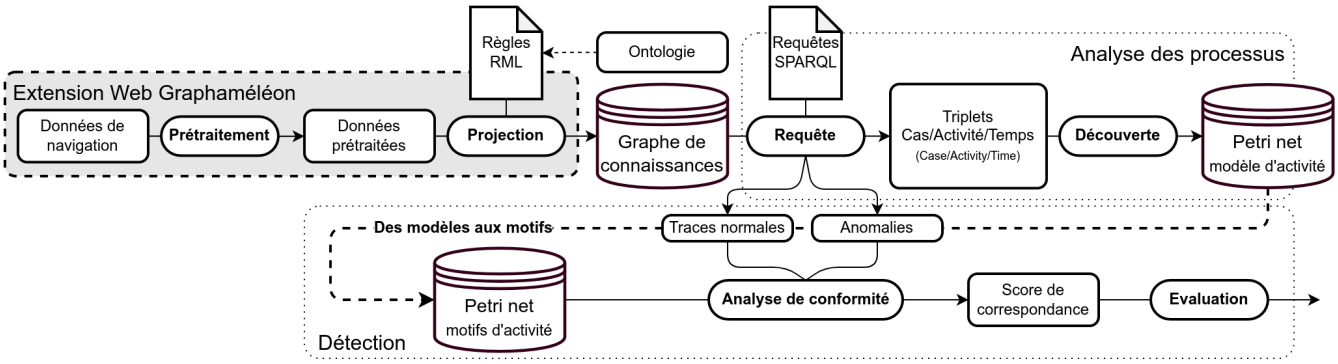


FIGURE 1 – Aperçu de la chaîne de traitement des données.
L'extension Web Graphaméléon capture et annote l'activité de l'utilisateur au niveau du navigateur Web. Un composant d'extraction des processus dérive des modèles d'activité places/transitions (Petri net) à partir du graphe de connaissances RDF 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, côté client ou côté réseaux/serveurs, pour classer de nouvelles traces d'activité comme normales ou anormales.

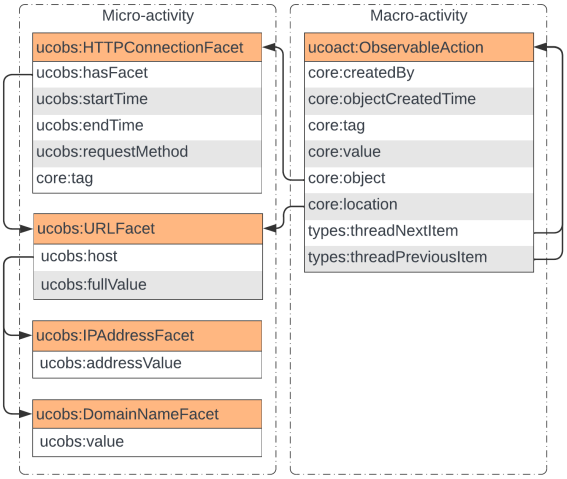


FIGURE 2 – Modèle de données.
Ce diagramme de classe définit les concepts et les propriétés utilisés pour la représentation sémantique des micro-activités (à gauche) et des macro-activités (à droite) tels que décrits dans la section 3.1. Pour les micro-activités, les classes et les propriétés présentées décrivent de manière précise une séquence de requêtes capturées au niveau du navigateur Web. Les macro-activités améliorent encore la modélisation en permettant la description des interactions. Les noms des concepts et des propriétés utilisés ici sont définis dans le vocabulaire UCO, les espaces de noms suivants s'appliquent : *core* = `https://ontology.unifiedcyberontology.org/uco/core#`, *ucobs* = `https://ontology.unifiedcyberontology.org/uco/observable#` et *types* = `https://ontology.unifiedcyberontology.org/uco/types#`.

des en-têtes de celles-ci. La Table 1 résume le type de données collectées par Graphaméléon. Ces informations incluent les URLs, les addresses 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 Web, et éviter ainsi une diversité excessive dans l'interprétation des activités pour des cas similaires. Pour 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 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* interactions *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

4. `https://www.w3.org/TR/fetch-metadata/`

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.

### 3.3 Détection d'anomalies et réseaux de Petri

Trois familles de techniques de détection d'anomalies sont présentées dans [24] 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 [14] où les modèles d'anomalies (i.e. 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 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 (i.e. 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'est-à-dire 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 modèles relèvent de la responsabilité de l'utilisateur (analyse, sélection, raffinement), et dépasse 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 : $anormal \equiv correspondance_{\{alignement|rejeu\}}(\textit{modèle}, \textit{motif}) < \eta$, avec $\eta$ 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.

## 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 et des réseaux P/T au sein d'un environnement contrôlé (§4.2). Les expériences sont menées à l'aide de Graphaméléon v2.1.0. Les données associées à ces expériences sont disponible sur `https://github.com/Orange-`

5. `https://developer.mozilla.org/fr/docs/Web/API/Document_Object_Model`

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 2 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 [35] (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 [19], les outils d'analyse de sites Web proposent principalement des analyses de positionnement par rapport à la concurrence [15]. 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" [6], "Garden Studio" [7] et "Mark My Images" [8] (MMI) sont identifiés dans [27] 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" [9] (EBO), "Scholarpedia" [10] et "Encyclopedia.com" [11] sont présentés dans [20] comme les trois principales alternatives à Wikipédia du point de vue de la fiabilité de l'information ;

***Content-Heavy*** où "RTI International" [12], "PrintMag" [13] et la "International Women's Media Foundation" [14] (IWMF) sont identifiés dans [22] 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 [15] (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 (i.e. l'événement `DOMContentLoaded` [16]) ; *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).

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

TABLE 2 – Statistiques pour l'expérimentation "Trafic réseau et complexité des sites Web".
Statistiques basées sur les modes de collecte "micro" (CM = $\mu$) 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.

**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

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

| | 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 3 – Moyenne du nombre d'entités en mode micro.
Comparaison de la moyenne du nombre d'entités UHC et UIP à partir de la Table 2 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`.

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 2 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 2. En ce qui concerne le mode micro (CM = $\mu$), les mesures présentent une variabilité significative entre chaque catégorie de complexité pour une politique d'anti-pistage donnée. La Table 3 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 3) 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 Catégorisation de traces de navigation

Dans cette deuxième expérience, notre objectif est de catégoriser les traces de navigation Web comme comportements normaux ou anormaux. Nous analysons les trois scénarios suivants 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.

**Scénario 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 une formulaire, puis retourne à la page d'accueil où il retrouve son livre dans la liste des ventes.

**Scénario 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.

**Scénario d'attaque XSS (anormal) :** un attaquant 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" et effectue une injection de code dans le champ "Auteur". Enfin, il retourne à la page d'accueil où le script injecté est exécuté.

Nous utilisons une simulation de site Web de librairie en ligne afin d'être en situation d'expérience contrôlée. Cela permet une exposition intentionnelle du site à diverses vulnérabilités de sécurité (une vulnérabilité XSS dans le cas présent, une forme courante d'attaque). Cela permet de même, avant l'étude, d'étiqueter chaque élément des pages Web du site, ce qui améliore l'interprétabilité des données collectées.

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 [2] (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 correspond au comportement "normal", est utilisé comme motif d'activité (i.e. 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 – Statistiques pour l'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 §4.2.

**Résultats & discussion.** En utilisant cette procédure, trois échantillons de données ont été produits. La Table 4 présente les statistiques relatives aux graphes de navigation résultants, et la Table 5 compare les résultats de diffé-

| | | 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 5 – 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.

rentes 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 (i.e. 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. Pour 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 d'avantage 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.

# 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 infrastructure réseaux. Sur les bases du projet DynaGraph [23], nous avons émis l'hypothèse que les graphes de connaissances peuvent structurer de façon adéquate les données collectées sur un navigateur Web au cours de sessions de navigation, et ce dans l'idée d'une analyse avancée des traces de navigation au travers d'une modélisation sous forme de réseaux de Petri et l'utilisation des outils associés aux techniques d'analyse des processus.

Pour tester notre approche, nous avons développé les concepts de micro-activité et de macro-activité en rapport au vocabulaire UCO [40] 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 deux 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, 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 mé-

thode fiable pour identifier les éléments HTML au sein des navigateurs Web, notamment par comparaison entre sessions de navigation ou entre utilisateurs.

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. En ce qui concerne l'analyse de conformité, trois options se présentent pour réduire la sensibilité de notre approche. La première consiste à partitionner le motif de référence en sous-motifs, ce qui devrait réduire l'amplitude de variation du score de correspondance en cas de non-conformité. La seconde consiste à utiliser des motifs spécifiques pour qualifier un groupe d'actions et localiser le groupe par alignement de séquence (p.ex. un motif décrivant une injection SQL plutôt qu'une description générale du comportement normal). La troisième approche consiste à pondérer l'importance des actions dans le calcul du score de correspondance activité/motif en utilisant le graphe de connaissances pour fournir du contexte (p.ex. une adresse IP source peu fréquente lors d'une attaque par injection SQL, un saut réseau impossible, un même utilisateur connecté depuis deux pays) ; l'idée étant d'utiliser une pondération qui masquerait les variations mineures dues au "bruit" par rapport aux variations causées par des erreurs réelles. Enfin, nous envisageons d'intégrer les modèles d'activité – via des vocabulaires appropriés pour les réseaux de Petri [11, 18] – dans un graphe RDF structuré par l'ontologie NORIA-O [25], ce afin de calculer des contextes d'anomalies enrichis par un processus de décision en utilisant la technique de plongement de graphes [24]. Nous analyserons notamment en quoi les modèles d'activité renforcent l'aide à la décision (p.ex. performance de la détection, interprétabilité) dans une situation de gestion d'incident avec connaissance partielle de l'activité des utilisateurs, comme cela peut être le cas lorsque l'analyse est menée côté réseaux/serveurs.

# Références


[1] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, José Emilio Labra Gayo, Sabrina Kirrane, Sebastian Neumaier, Axel Polleres, Roberto Navigli, Axel-Cyrille Ngonga Ngomo, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. Knowledge Graphs, 2020.

[2] Alessandro Berti, Sebastiaan van Zelst, and Wil 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)*, 2019.

[3] Amélie Cordier, Marie Lefevre, Pierre-Antoine Champin, Olivier Georgeon, and Alain Mille. Trace-Based Reasoning - Modeling Interaction Traces for Reasoning on Experiences. In *The 26th International FLAIRS Conference*, 2013.

[4] Amina Annane, Nathalie Aussenac-Gilles, and Mouna Kamel. BBO : BPMN 2.0 Based Ontology for Business Process Representation. In *20th European Conference on Knowledge Management (ECKM)*, Lisbon, Portugal, 2019.

[5] Anastasia Dimou, Miel Vander Sande, Pieter Colpaert, Ruben Verborgh, Erik Mannens, and Rik 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 23$^{rd}$ International World Wide Web Conference (WWW 2014)*. CEUR-WS.org, 2014.

[6] Bernard Berthomieu, Pierre-Olivier Ribet, and François Vernadat. The tool tina – construction of abstract state spaces for petri nets and time petri nets. *International Journal of Production Research*, 2004.

[7] Caleb C. Noble and Diane J. Cook. Graph-based anomaly detection. 2003.

[8] Maria Carla Calzarossa and Luisa Massari. Analysis of header usage patterns of http request messages. In *IEEE International Conference on High Performance Computing and Communication*, 2014.

[9] Giovanna Castellano, Anna M. Fanelli, and Maria A. Torsello. *Web Usage Mining : Discovering Usage Patterns for Web Applications*. Springer Berlin Heidelberg, 2013.

[10] Pierre Dagnely, Tom Ruette, Tom Tourwé, and Elena Tsiporkova. Ontology-driven multilevel sequential pattern mining : mining for gold in event logs of photovoltaic plants. In *2018 International Conference on Intelligent Systems (IS)*, 2018.

[11] Dragan Gašević and Vladan Devedžić. Petri net ontology. *Knowledge-Based Systems*, 2006.

[12] Franck Ghitalla, Dominique Boullier, and Mathieu Jacomy. *Qu'est-Ce Que La Cartographie Du Web ?: Expéditions Scientifiques Dans l'univers Des Données Numériques et Des Réseaux*. 2021.

[13] Iman Akbari, Mohammad A. Salahuddin, Leni Ven, Noura Limam, Raouf Boutaba, Bertrand Mathieu, Stephanie Moteau, and Stéphane Tuffin. Traffic classification in an increasingly encrypted web. *Communications of the ACM*, 2022.

[14] Ines Chami, Sami Abu-El-Haija, Bryan Perozzi, Christopher Ré, and Kevin Murphy. Machine Learning on Graphs : A Model and Comprehensive Taxonomy, 2020.

[15] James 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.

[16] Johannes Koch, Carlos A. Velasco, and Philip Ackermann. Http vocabulary in rdf 1.0. W3c working group note, W3C, 2017.

[17] Jorge Munoz-Gama. *Conformance Checking and Diagnosis in Process Mining : Comparing Observed and Modeled Processes*. PhD thesis, Universitat Politècnica de Catalunya – BarcelonaTech, Barcelona, 2014.

[18] Juan C. Vidal, Manuel Lama, and Alberto Bugarın. A High-level Petri Net Ontology Compatible with PNML. 2006.

[19] Kathy Haan. Top website statistics for 2023. https://www.forbes.com/advisor/business/software/website-statistics/, 2023. Accessed : 2023-08-10.

[20] Kent Campbell. Seven free wikipedia alternatives. https://blog.reputationx.com/wikipedia-alternatives, 2023. Accessed : 2023-08-10.

[21] Pierre Laperdrix, Nataliia Bielova, Benoit Baudry, and Gildas Avoine. Browser fingerprinting : A survey, 2019.

[22] Laura 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.

[23] Lionel Tailhardat, Raphaël Troncy, and Yoan Chabot. Walks in cyberspace : Towards better web browsing and network activity analysis with 3d live graph rendering. Association for Computing Machinery, 2022.

[24] Lionel Tailhardat, Raphael Troncy, and Yoan Chabot. Leveraging knowledge graphs for classifying incident situations in ict systems. In *18th International Conference on Availability, Reliability and Security (ARES)*, 2023.

[25] Lionel Tailhardat, Yoan Chabot, and Raphaël Troncy. NORIA-O : an Ontology for Anomaly Detection and Incident Management in ICT Systems. In *Semantic Web – 21$^{st}$ International Conference, ESWC 2024, Hersonissos, Crete, Greece, May 26 - 30, 2024, Proceedings*, 2024.

[26] Vítor Santos Lopes and João 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.

[27] Madhu 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.

[28] Mathieu Lirzin and Béatrice Markhoff. Vers Une Ontologie Des Interactions HTTP. In *31$^{emes}$ Journées Francophones d'Ingénierie Des Connaissances*, Angers, France, 2020.

[29] Megan Katsumi and Mark Fox. iCity Transportation Planning Suite of Ontologies. Technical report, University of Toronto, 2020.

[30] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. Deep learning for anomaly detection. *ACM Computing Surveys*, 2021.

[31] Heeryon Park and Doo-Kwon Baik. Web log session identification based on cluster-based classification. In *7th International Conference on Advanced Information Networking and Applications (AINA)*. IEEE, 2011.

[32] Stephen Pauwels and Toon Calders. Extending dynamic bayesian networks for anomaly detection in complex logs, 2018.

[33] Peter E. Kaloroumakis and Michael J. Smith. Toward a Knowledge Graph of Cybersecurity Countermeasures. Technical report, The MITRE Corporation, 2021.

[34] Sasan Saqaeeyan, Hamid Haj Seyyed Javadi, and Hossein Amirkhani. Anomaly detection in smart homes using bayesian networks. *KSII Transactions on Internet and Information Systems*, 2020.

[35] 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.

[36] W3C. Fetch metadata request headers. Working draft, W3C, July 2021.

[37] Xindong Wu, Xingquan Zhu, Gongqing Wu, and Wei Ding. Data mining with big data. *IEEE Transactions on Knowledge and Data Engineering*, 2004.

[38] Nong Ye. A markov chain model of temporal behavior for anomaly detection. 2000.

[39] Youssra Rebboud, Pasquale Lisena, and Raphael Troncy. Beyond Causality : Representing Event Relations in Knowledge Graphs. In *Knowledge Engineering and Knowledge Management*. Springer International, 2022.

[40] Zareen Syed, Ankur Padia, M. Lisa Mathews, Tim Finin, and Anupam Joshi. UCO : A Unified Cybersecurity Ontology. In *AAAI Workshop on Artificial Intelligence for Cyber Security*. AAAI Press, 2016.

[41] Yan Zhao, Liwei Deng, Xuanhao Chen, Chenjuan Guo, Bin Yang, Tung Kieu, Feiteng Huang, Torben Bach Pedersen, Kai Zheng, and Christian S. Jensen. A comparative study on unsupervised anomaly detection for time series : Experiments and analysis, 2022.