Systèmes producteurs de confiance : ouverture de droit à des services par apprentissage dynamique du comportement des utilisateurs du système d'information

Diyé Dia

▶ **To cite this version:**

Diyé Dia. Systèmes producteurs de confiance : ouverture de droit à des services par apprentissage dynamique du comportement des utilisateurs du système d'information. Autre [cs.OH]. Université Blaise Pascal - Clermont-Ferrand II, 2016. Français. NNT : 2016CLF22396 . tel-02063414

HAL Id: tel-02063414
https://theses.hal.science/tel-02063414

Submitted on 11 Mar 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

DU : 2396 – EDSPIC : 623

UNIVERSITE BLAISE PASCAL

ECOLE DOCTORALE SPI

SCIENCES POUR L'INGENIEUR

T H E S E

Présentée par

Diyé Dia

pour obtenir le titre de

Docteur d'Université

Spécialité : Informatique

Systèmes producteurs de confiance : ouverture de droit à des services par apprentissage dynamique du comportement des utilisateurs du système d'information.

---

préparée au Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systémes, CIFR en collaboration avec Almerys soutenue le 17/03/2016

Rapporteurs :	
Bénédicte Le Grand	Professeur des Universités, Université Panthéon - Sorbonne
Yannick Toussaint	Chargé de Recherche, LORIA de Nancy
Examinateurs :	
Yannick Loiseau	Maître de Conférences, Université Blaise Pascal
Olivier Raynaud	Maître de Conférences HDR, Université Blaise Pascal
Directeur :	
Engelbert Mephu Nguifo	Professeur des Universités, Université Blaise Pascal

Table des figures

1 Le processus de fouille de données 14

1.1 Définition de la confiance en sociologie 21
1.2 L'évolution de la confiance numérique en fonction du risque pris . . . 22
1.3 Système de réputation centralisé 28
1.4 Système de réputation décentralisé 29
1.5 Processus de confiance 35
1.6 eBeeHome, un espace de vie numérique 38
1.7 Diagramme de cas d'utilisation du modèle de confiance 46
1.8 L'interaction des acteurs du modèle de confiance 48
1.9 Diagramme de séquence pour la confiance 49
1.10 Le fonctionnement du modèle de réputation d'Anirban Basu adapté à notre contexte 50
1.11 Vue d'ensemble du modèle proposé 51
1.12 Vue détaillée du modèle proposé 52
1.13 Vue détaillée de l'évaluation de la réputation 53
1.14 Identification vs Authentification 58
1.15 Le processus d'identification 59
1.16 Le processus d'authentification 59
1.17 Architecture de l'authentification implicite à base d'identification . . 63
1.18 Fonctionnement de l'authentification implicite dans eBeeHome : vue globale 66
1.19 Fonctionnement de l'authentification implicite dans eBeeHome : apprentissage du profil 67
1.20 Fonctionnement de l'authentification implicite dans eBeeHome : étape d'authentification 68

2.1 Résumé des motifs utilisés 72
2.2 Base de données d'achats 73

2.3 Un environnement d'environnement $\{\mathscr{D}, \mathscr{M}, \mathscr{R}\}$ 74
2.4 Liste des articles achetés 76
2.5 Base de transaction du panier de la ménagère 77
2.6 Treillis des *itemsets* 79
2.7 Treillis des *itemsets* fréquents 79
2.8 Treillis des *itemsets* fermés fréquents 81
2.9 Un exemple de base de transaction inspiré de [1] 81
2.10 Résumé des motifs utilisés 84
2.11 Une base de données $\{\mathscr{O}, \mathscr{M}, \mathscr{R}\}$ 89

3.1 Hiérarchisation des modèles de prédictions. 93
3.2 Le processus de classification : construction du modèle 94
3.3 Le processus de classification : phase de test 94
3.4 Exemple de cadre d'étude 100
3.5 Le processus itératif du modèle 104
3.6 Treillis des fermés 108
3.7 Sélection des patterns propres 112
3.8 L'ensemble des patterns propres de tous les utilisateurs $\mathcal{P}_{all} = \bigcup_i \mathcal{P}_{u_i}$ définit un vecteur $\mathcal{V}$. 113
3.9 Les vecteurs de profils de Marion et Simon avec la mesure du support. 113

4.1 Évolution du nombre d'internautes en France 118
4.2 Fichier des données d'entrée 121

5.1 Influence d'un correctif sur la classification de Bayes. 130
5.2 Bayes à base de fermés. 131
5.3 Bayes à base de fermés avec l'hypothèse d'amélioration. 131
5.4 Bayes à base de fermés fréquents discriminants + filtre. 131
5.5 Performances comparatives des modèles naïfs de *Bayes*, *Bayes(χ^2)* et *bayésien à base d'*itemsets *fermés fréquents discriminants*. 133
5.6 Performance comparative de H_1, H_2, H_3 et Bayes. 135
5.7 Performance comparative de Bayes, *support-based profiling*, *lift-based profiling* et H_1. 138
5.8 Performance comparative de H_1 avec la similarité par cosinus et la distance euclidienne, Bayes et *lift-based profiling*. 139
5.9 Performance comparative de H_1 avec la similarité par cosinus, de Dice et de Kulczynski. 140

Liste des tableaux

1.1 La trace de *connexion* de la brique de traçabilité 41
1.2 Comparaison de différents méthodes d'authentification 65

4.1 Statistiques descriptives de la base de données 121
4.2 Statistiques descriptives d'un jeu de données de 30 utilisateurs 124

5.1 Exemple de matrice de confusion 128
5.2 Taille du vecteur profil et distribution des *itemsets* fermés fréquents propres par rapport à leur taille. 135
5.3 Performances comparatives avec l'étude décrit dans [2] 137

Table des matières

Table des figures **2**

Liste des tableaux **4**

Remerciements **9**

Résumé **10**

Abstract **11**

Introduction **13**

1 Confiance numérique **19**
1.1 Introduction 20
1.2 Définitions 24
1.2.1 Confiance à base de politiques 25
1.2.2 Réputation et recommandation 27
1.2.2.1 Système de réputation centralisé et décentralisé . . . 27
1.2.2.2 Confiance à base de réputation et contexte 30
1.2.2.3 Calcul de la confiance à base de réputation et prise en compte de paramètres 31
1.2.3 Confiance et sciences sociales 32
1.2.4 Contraintes et qualités du modèle à concevoir 36
1.3 Exemple de modèle de confiance 37
1.3.1 Contexte de l'étude 37
1.3.2 Cahier des charges : les usages du modèle 43
1.3.3 Conception 47
1.3.4 Limites et extensions 55
1.4 Authentification implicite 56

1.4.1 Identification et authentification . . . 57
1.4.2 Définitions et concepts . . . 60
1.4.2.1 Implicite vs explicite . . . 60
1.4.2.2 Authentification implicite basée sur le comportement des utilisateurs . . . 61
1.4.2.3 Questions de recherche . . . 62
1.4.3 Authentification implicite dans eBee . . . 63
1.5 Résumé et discussion . . . 64

2 *Itemsets* fermés fréquents discriminants 69
2.1 Introduction . . . 70
2.2 Définitions des notions de base . . . 72
2.3 *Itemsets* fermés fréquents . . . 75
2.3.1 Formalisation . . . 75
2.3.2 Extraction d'*itemsets* fermés fréquents . . . 81
2.4 Outils de discrimination . . . 85
2.4.1 Outils statistiques . . . 87
2.4.2 Stabilité des *itemsets* fermés fréquents . . . 88
2.5 Résumé et discussion . . . 89

3 Classification 91
3.1 Introduction . . . 92
3.2 Motivations : limites des approches existantes . . . 95
3.3 Classification naïve bayésienne . . . 99
3.3.1 Adaptation simple de la classification naïve bayésienne . . . 99
3.3.2 Classification naïve bayésienne à base d'*itemsets* fermés fréquents discriminants . . . 101
3.4 Classification à base d'*itemsets* fermés fréquents discriminants . . . 105
3.4.1 Sélection des patterns propres . . . 108
3.4.2 Calcul des profils utilisateurs . . . 112
3.4.3 Mesures de distance . . . 114
3.5 Résumé . . . 115

4 Contexte applicatif : navigation sur Internet 116
4.1 Introduction . . . 117
4.2 Description des données . . . 119
4.3 Fouille des données de navigation sur Internet . . . 121
4.4 Pré-traitements sur les données . . . 123
4.4.1 Page de publicité . . . 123

4.4.2 Définition des sessions 124
4.5 Résumé 125

5 Expérimentations **126**
5.1 Introduction 127
5.2 Application des différentes méthodes 127
5.2.1 Classification naïve bayésienne 129
5.2.2 Classification à base de d'*itemsets* fermés fréquents discriminants 133
5.3 Logiciel de reconnaissance comportementale 140
5.4 Résumé et Discussion 140

Conclusion **141**
5.5 Contributions 142
5.6 Perspectives 144

Bibliographie **147**

Celui qui trouve sans chercher est celui qui a longtemps cherché sans trouver.

Gaston Bachelard, (1884-1962).

Remerciements

Je tiens tout d'abord à remercier tous les membres du jury, plus particulièrement Bénédicte Le Grand et Yannick Toussaint les deux rapporteurs pour avoir accepté cette tâche importante de relecture minutieuse de mon manuscrit. Je remercie aussi Engelbert Mephu Nguifo d'avoir bien voulu être mon directeur de thèse ; de même que mes trois encadrants : Olivier Raynaud, Yannick Loiseau et Olivier Coupelon pour leur soutien moral, scientifique et administratif apporté durant ces années de thèse. Ils m'ont écouté, conseillé et orienté dans mon travail de recherche. Je les remercie également pour tous les moments agréables que nous avons partagés.

Je remercie aussi l'ensemble des enseignants du Master Recherche, option Recherche Opérationnelle ainsi que le corps professoral de l'ISIMA pour leur encadrement et pour m'avoir fait apprécier la recherche.

Je suis également reconnaissante envers l'ensemble des employés d'Almerys avec lesquels j'ai pu collaborer activement pendant plus d'une année et particulièrement Jean-Marie Tirbois.

Je tiens également à exprimer ma gratitude à l'ensemble des membres du Laboratoire d'Informatique de Modélisation et d'Optimisation des Systèmes (LIMOS) pour la convivialité qui a régné au sein de ce laboratoire. Mes remerciements ne seraient pas complets si je ne mentionnais pas mes collègues de bureau passés et présents : Libo, Raksmey, Johnathan, Nader, Valentin et Benjamin. Mes voisins et voisines côtoyés au sein de ce laboratoire ne sont pas oubliés (Lakhdar, Maxime, Marie, Sabeur, Salsabil, Kaoutar, Kahina, Ben, Firmin, Damien, Fabien, Vanel et Benjamin). Je remercie particulièrement Philippe Lacomme pour sa bonne humeur.

Je remercie aussi tous les membres de ma famille qui m'ont soutenu et accompagné durant de nombreuses années, ainsi que toutes les personnes qui m'ont apporté un soutien moral de près ou de loin.

Résumé

Ce manuscrit est constitué de deux études autour de la confiance au sein des services en ligne. Les premiers travaux portent sur les modèles généraux de confiance. Ceux-ci s'inspirent des mécanismes de confiance décrits en sciences humaines à savoir la réputation, la recommandation et la réciprocité. L'idée est de mettre en place au sein des sites d'e-commerce par exemple, des modèles producteurs de confiance qui vont améliorer les services fournis mais aussi le chiffre d'affaires, et permettre la diminution des coûts relatifs aux moyens de sécurité complexes. Dans ce dernier cas, la confiance est un substitut de la sécurité. Proposé dans cette thèse, le modèle général de confiance intégré dans un espace de vie numérique s'appuie sur une architecture décentralisée. Ce modèle est constitué d'outils empruntés à la statistique descriptive.

L'usurpation d'identité et le vol de données personnelles sont devenus des pratiques répandues sur Internet. Pour protéger leurs clients et leurs systèmes de ces menaces, les fournisseurs de services en ligne maintiennent un niveau de sécurité optimal. Or l'utilisation des bonnes pratiques de sécurité de plus en plus performantes peut être contraignante pour les clients. Il est donc nécessaire de proposer une technologie qui allie sécurité et convivialité. L'authentification implicite est une technologie complémentaire aux éléments de sécurité traditionnels. Elle permet de renforcer la sécurité des services en ligne mais aussi d'augmenter la convivialité car le client n'est pas sollicité lors du processus d'authentification. Dans cette deuxième étude, nous mettons en place un processus d'authentification implicite lors de la navigation sur Internet en utilisant la fouille de données. Le flux des activités des internautes est un ensemble de sites visités répartis en sessions. A partir des flux des activités des internautes, nous construisons leurs profils en utilisant la recherche d'*itemsets* fermés fréquents. Une sélection des *itemsets* fermés fréquents les plus discriminants est réalisée par trois heuristiques que nous proposons. Enfin, une classification s'appuyant sur les objets issus des heuristiques est exposée.

Mots-Clés : fouille de données, sécurité, confiance, authentification implicite, *itemsets* fermés fréquents discriminants.

Abstract

In this thesis, we work on two subjects about trust and confidence in online services. Our early works are about setting general models of trust. This kind of model is adapted from model of trust in human sciences and take into account reputation, recommendation and reciprocity. The idea behind is to set up in e-commerce site for example, models of trust in order to improve provided services and sales revenue. Models of trust allow to reduce the costs of complex means for security. In this last case, trust is a substitute of security. Proposed in this thesis, general model of trust in a digital life space is based on a decentralized framework. This model uses tools from descriptive statistics.

Identity and personal data theft have become common practices in the Internet. To protect customers and systems, online service providers maintain a very high level of security. However, the use of increasingly efficient good security practices can be burdensome for customers. It is therefore necessary to propose a technology that combines security with customer-friendliness. Implicit authentication is a complementary technology for « conventional » authentication mechanisms. It allows to strengthen the security for online services but also to increase the customer-friendliness as the client is not solicited through the authentication process. In this second study, we establish an implicit authentication process when browsing the Web by mining data. The flow of activities Web users is modeled as a set of visited websites divided into sessions. From the flow of activities, we build user profiles by using frequent closed itemsets mining algorithms. We propose three heuristics to select the most discriminative frequent closed itemsets. Finally, a discriminative frequent closed itemsets based classification is presented.

Keywords : data-mining, security, trust, implicit authentication, discriminative frequent closed itemsets.

La connaissance s'acquiert par l'expérience, tout le reste n'est que de l'information.

Albert Einstein, (1879 - 1955).

Introduction

Chaque seconde, 29000 Gigaoctets d'informations sont publiés dans le monde d'après le Planetoscope [1]. Cette quantité d'informations représente 2,5 Exaoctets par jour soit 912,5 Exaoctets dans l'année.

A l'ère du « Big Data », les données produites et collectées sont de plus en plus importantes et complexes. Une collecte automatique est de plus en plus souvent mise en place sur les sites Internet. Ces données ont besoin d'être analysées et comprises afin d'améliorer les prises de décisions. Le processus de découverte de la connaissance est une méthodologie adaptée qui permet de comprendre les masses de données d'aujourd'hui. Il existe de multiples enjeux industriels. En effet, l'exploitation de la connaissance issue de ces bases de données gigantesques améliore les chiffres d'affaires des entreprises. Le processus de découverte de la connaissance a été introduit par Fayyad et al. dans [3] (cf. figure 1).

La *sélection des données* est la première étape du processus de fouille de données. Elle permet de sélectionner, dans une base de données, les informations relatives au problème pour lequel de nouvelles connaissances doivent être proposées.

Le *pré-traitement des données* qui intervient après la sélection des données, permet d'en améliorer leur qualité. Cette étape a pour objectif de nettoyer les données.

La *transformation et la fouille de données* est l'étape phare du processus représenté par la figure 1. La transformation permet de mettre les données au format d'entrée des algorithmes de fouille de données. La fouille de données, étape la plus importante du processus, permet d'explorer les données sélectionnées et pré-traitées afin d'extraire un ensemble d'informations.

L'*interprétation et l'évaluation* de l'ensemble d'informations issus de l'étape précédente permettent de les comprendre. Cette phase consiste à traiter le format de sortie des algorithmes pour rendre les résultats facilement visualisables et analysables par les utilisateurs, afin de rendre interprétables des informations inintelligibles. Cette étape nécessite l'intervention d'experts du domaine d'étude.

1. http ://www.planetoscope.com/, accès le 18/10/2015.

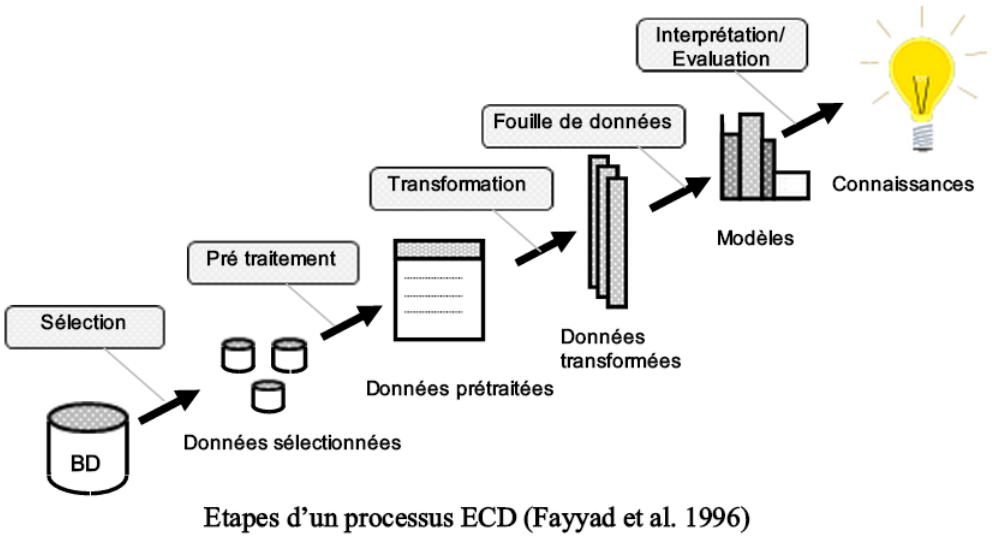


FIGURE 1 – Le processus de fouille de données

Bien que l'étape de fouille de données soit centrale, le temps passé à cette étape est évalué à moins de 20%. Par conséquent, plus de 80% du temps est dédié aux opérations de sélection, nettoyage, enrichissement et codage. Notre étude se focalise surtout sur l'étape centrale, nous n'avons pas passé beaucoup de temps sur l'étape de nettoyage des données. La fouille de données permet d'expliquer le passé et de prédire le futur grâce à l'analyse des données. A la croisée des statistiques, de l'apprentissage automatique, de l'intelligence artificielle et des technologies de gestion de base de données, cette étape est un domaine pluridisciplinaire qui permet d'extraire de la connaissance utile à partir de très grandes bases de données.

Motivations

L'entreprise Almerys offre des solutions numériques d'abord pour son cœur de métier, le tiers-payant mais aussi pour la santé et la dématérialisation. Cette thèse réalisée en collaboration avec Almerys, est motivée par le manque de confiance observé au sein de la vie numérique en général (cf. [4]). Ce manque (cf. [5]) envers les organismes et les services en ligne particulièrement est la problématique posée par le travail mené par la FING (Fondation Internet Nouvelle Génération). Plusieurs entreprises se sont réunies autour de la FING pour trouver des pistes de recherche

permettant de produire de nouveau un sentiment de confiance au sens de la sociologie. En sciences humaines, la confiance est un acte de foi qui peut reposer sur des échanges. Cette production de confiance pourra améliorer le climat entre les fournisseurs et les consommateurs. En effet, la confiance est un moteur de développement (cf. [6]) nécessaire au développement. De plus pour les organisations, la méfiance est de plus en plus coûteuse : par exemple dans la gestion quotidienne d'un service, dans les relations avec des sous-traitants, nous nous apercevons que les systèmes de contrôle sont très onéreux. Donnons un autre exemple. Le concept de la dématérialisation aujourd'hui permet à un employeur de déposer des bulletins de paie dans le coffre numérique de l'employé. Dans le coffre de ce dernier, d'autres documents y sont également déposés. La relation entre les fournisseurs de documents et les utilisateurs du coffre est fait par un tiers de confiance. Dans ce contexte, la peur des employés de se faire espionner par les employeurs est un frein à l'usage de ce produit. Le manque de confiance provient du manque de connaissance du système par les utilisateurs. Une transparence raisonnée dans la gestion du système pourra permettre de retrouver une certaine confiance.

Chaque seconde, 19 transactions d'achat sont enregistrées en France par les sites d'e-commerce. Cela représente 600 millions de transactions par an en 2013 (contre 340 millions en 2010). Il y a environ 138 000 sites marchands d'e-commerce actifs en France. Près de 1640 euros sont achetés chaque seconde sur les sites marchands français (e-commerce) soit 51,7 milliards d'euros de chiffre d'affaires annuel réalisé par 138.000 sites marchands (+17%/2011, + 15%/2012). 33 millions de Français achètent sur Internet. En 2012, le taux de satisfaction après achat, déjà haut, atteint le niveau inégalé de 98% ; et la fréquence d'achat dépasse 16 achats sur le net par an par acheteur, contre 12 il y a 2 ans et 10 il y a 4 ans. La mise en place de modèles de confiance au sein des sites de vente en ligne explique la satisfaction des clients au cours des dernières années. Une utilisation accrue des sites marchands est observée de nos jours malgré une augmentation du taux de piratage des comptes en ligne parallèlement. Le nombre de tentatives de fraudes à la carte bancaire sur Internet en France a été de 795 031 en 2012, soit près de 2200 tentatives de fraudes par jour. Le montant total des détournements a atteint 164 millions d'euros [2]. Dans le domaine de la confiance numérique, il est évident que ces risques sont un frein à la confiance pouvant s'installer entre l'utilisateur et la plateforme web ciblée. Pour contrer cela, nombreux sont les messages d'avertissements à destination des utilisateurs dont le but est de leur expliquer les bonnes pratiques comme le fait d'avoir un mot de passe suffisamment complexe pour qu'une machine de calcul ne puisse pas le trouver avant plusieurs années d'exécution. Tous les problèmes ne viennent malheureusement pas de l'ignorance des utilisateurs, ils peuvent également provenir des serveurs des sociétés

2. http ://www.planetoscope.com, accès le 18/10/2015.

qui ne sont pas assez sécurisés. Il est donc maintenant monnaie courante, dans les sites d'actualités informatiques (voire sur les plateformes d'informations générales), de voir une société comme Google, Yahoo ou Microsoft subir une attaque sur ses serveurs. Ces attaques permettent de récupérer des informations de plusieurs millions de comptes. Près de 120000 cyberattaques ont lieu chaque jour dans le monde. Un autre vecteur d'attaque des pirates informatiques se situe sur les potentielles failles des logiciels, qui permettent de prendre le contrôle partiel ou total de l'ordinateur (ou maintenant du mobile ou d'une tablette) d'un individu à son insu. En fonction de la criticité des failles, il est possible d'en prendre le contrôle sans interaction avec l'utilisateur et donc, si l'utilisateur a activé un système de connexion automatique à ses services favoris, de les utiliser sans son accord. Nous venons de présenter les trois principales raisons d'utiliser à l'insu d'un utilisateur ses services avec ses identifiants. Nous aurions également pu parler des nombreuses attaques de *fishing* ou encore d'écoute des réseaux mobiles ou publics (de plus en plus développés et dont les informations circulent en clair).

Dans ce contexte de piratage permanent, comment arriver à faire suffisamment confiance à la personne avec laquelle nous parlons via une messagerie instantanée, ou encore dans l'utilisation d'un service *critique* tel qu'un site bancaire, ou d'achats en ligne ?

Pour contrer toutes ces attaques, les sociétés de services sur Internet multiplient les envois de courriels préventifs contre les usurpations d'identités (qu'elles proviennent d'autres utilisateurs ou directement d'un site internet copié) et les éditeurs déploient de plus en plus rapidement les correctifs de failles sur leurs logiciels. Les sites de services critiques, tels que les banques, mettent maintenant en place des systèmes d'authentification multi-plateformes avec l'envoi de codes uniques, car même si le piratage d'un appareil peut être facile, le piratage de deux appareils d'un même utilisateur est quant à lui beaucoup plus difficile et les moyens pour mettre en place sa réalisation sont souvent plus coûteux que les gains rapportés à terme. Ces solutions, bien qu'efficaces, ne sont pas pour autant un gage de sécurité absolue et ne peuvent pas s'appliquer sur tous les usages, car ces démarches sont souvent lourdes pour l'utilisateur, qui doit à chaque nouvelle session attendre, par exemple, de recevoir un code de validation sur son téléphone.

Contributions

Compte tenu des motivations et des enjeux industriels, les contributions de cette thèse sont les suivantes :

- Une première approche est de mettre en place un environnement de confiance afin de faire face aux piratages et menaces. Améliorer les modèles de confiance et produire de la confiance pourraient permettre d'avantages d'échanges sur Internet. *Un modèle théorique de confiance* est proposé dans cette thèse. Ce modèle s'appuie sur les acquis de l'état de l'art et est adapté au contexte d'étude industriel. A partir d'une étude décrite dans [7], un modèle respectant les consignes de l'entreprise est construit.

- Une autre idée pour garantir l'authenticité d'un utilisateur serait de surveiller ses agissements en temps réel et de les comparer à ce qu'il a l'habitude de faire sur le service en question. Il s'agit là d'une solution se greffant aux solutions déjà existantes et agissant comme un certificat prouvant l'identité de la personne. *L'authentification implicite à base de classification à l'aide d'*itemsets *fermés fréquents* est une méthodologie fondée sur des outils existants. L'originalité vient de la combinaison de ces outils pour répondre à une problématique du domaine de la sécurité. Concrètement, nous proposons d'authentifier une personne sans que celle-ci n'ait à intervenir ou ne soit interrompue dans ses actions par une demande de mot de passe, par exemple. Cette authentification s'effectuera en analysant en temps réel les actions de l'utilisateur, puis en les comparant avec les actions qu'il a l'habitude de faire. Nous pouvons effectivement facilement imaginer que les actions d'un pirate ayant usurpé l'identité du vrai utilisateur vont être très différentes des actions habituelles, comme la consultation abusive des données bancaires, ou un envoi massif de messages aux contacts de l'utilisateur. Le développement des objets connectés et intelligents, de la domotique et des systèmes automatiques nous permet d'élargir cette notion d'authentification implicite et d'actions d'un utilisateur à une suite d'actions de n'importe quel équipement informatique.

  Notons que des géants du numérique utilisent l'analyse comportementale pour faire du ciblage publicitaire, notamment, et ne dévoilent pas au public les algorithmes utilisés. Almerys a décidé de publier les résultats sur l'analyse comportementale qui a été faite.

Plan du manuscrit

Le chapitre 1 présente un état de l'art sur la confiance dans le domaine de l'informatique. Dans ce chapitre, un modèle général de confiance numérique est également proposé après avoir défini les besoins de l'entreprise avec laquelle cette thèse a été réalisée.

Le chapitre 2 aborde un état de l'art sur les méthodes de fouille de données utilisant l'Analyse Formelle des Concepts. Nous y étudierons en particulier les *itemsets* fermés fréquents.

Le chapitre 3 présente la principale contribution. La classification à base d'*itemsets* fermés fréquents y est détaillée.

L'application des différentes méthodes à un contexte de navigation sur Internet est traitée dans les chapitres 4 et 5. Le chapitre 4 pose le contexte de la navigation sur Internet par des étudiants d'une université. Le chapitre 5 présente les résultats obtenus.

Chapitre 1

Confiance numérique

> ... en contemplant le nid, nous sommes à l'origine d'une confiance au monde, nous recevons une amorce de confiance, un appel à la confiance cosmique. L'oiseau construirait-il son nid s'il n'avait son instinct de confiance au monde ?
>
> **Gaston Bachelard**, La Poétique de l'espace, 1957, p. 102.

1.1 Introduction

Du latin *con-* (qui signifie é ensemble ») et *fidere* (qui signifie « se fier », « croire »), la confiance est définie en deux parties par le Trésor de la Langue Française informatisé (TLF). En parlant d'une relation de personne à personne, la confiance est une « croyance spontanée ou acquise en la valeur morale, affective, professionnelle... d'une autre personne, qui fait que l'on est incapable d'imaginer de sa part tromperie, trahison ou incompétence ». Elle est aussi, dans une relation de soi à soi, une assurance en ses capacités. On parle alors de confiance en soi. Enfin la confiance devient un « sentiment de sécurité et d'harmonie », dans le contexte d'une relation aux choses et au monde.

La confiance se traduit en anglais par les mots *trust* ou *confidence*. Ceux-ci ont des significations sensiblement différentes. Les définitions proposées sont issues du domaine de la sociologie. *Confidence* s'apparente à la confiance assurée expliquée dans [8] comme une confiance qui se base sur des garanties, un savoir et des sanctions en cas de rupture de l'accord ou du contrat. La confiance assurée est donc basée sur la familiarité par rapport à une situation donnée comme le montre la figure 1.1. *Trust* ou la confiance décidée intervient lorsqu'il n'est pas possible d'avoir une confiance assurée. Cette confiance décidée est utilisée dans un contexte où les individus sont incapables de prédire le comportement de l'autre. Ce type de confiance permet de prendre des décisions dans un contexte méconnu en acceptant le risque. La confiance *trust* est un pari et se substitue au contrat, à la familiarité et à la sécurité. La confiance permet de faciliter les relations professionnelles ou personnelles. Son absence est donc à l'origine de tensions dans une relation.

Présentée dans ce chapitre, la notion de confiance numérique est introduite dans [4] en 2012. D'après ce dernier, la confiance numérique est une confiance décidée (*trust*). En outre, la confiance numérique est la solution à la crise de confiance généralisée dans les institutions, dans les grandes entreprises et en particulier dans l'utilisation des données personnelles. En effet des enquêtes (cf. [4]) mettent en avant une baisse de la confiance des citoyens français vis-à-vis des gouvernements et des banques. Une étude, réalisée en ligne auprès de 772 internautes en février 2013, a montré une défiance des Français à l'encontre des services numériques (cf. [5]). Plus précisément cette étude conclut que 86% des internautes utilisent des services numériques, alors que la confiance (perception du sentiment de confiance) diminue dans l'e-commerce (53% en 2013 contre 56% en 2011), dans les réseaux sociaux (32% en 2013 contre 35% en 2011) et se stabilise dans la banque en ligne (76%). L'industrialisation de la relation (utilisation de centres d'appels, sites internet...) et l'économie de l'attention sont à l'origine de cette crise de la relation, toujours d'après [4]. Dans ce contexte de crise, une expédition (un travail collectif de quelques mois) de la FING (Fondation Internet

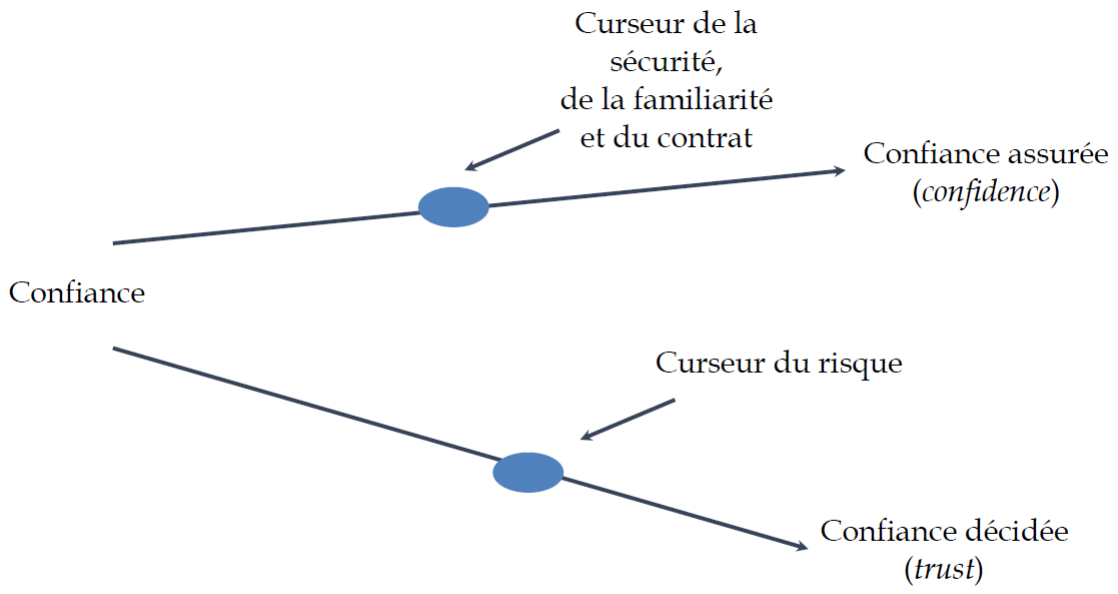


FIGURE 1.1 – Définition de la confiance en sociologie

Nouvelle Génération) et de la fondation Télécom s'est penchée sur les problématiques évoquées précédemment. La réflexion de cette expédition débouche sur une conclusion nette : « la mobilisation des chercheurs, des entreprises et des institutions sur la confiance doit changer de cible. Plutôt que de se focaliser sur les risques et sur la sécurité, elle doit aujourd'hui se recentrer sur la production même de la confiance ». La production de confiance numérique permet la prise de décision et l'action dans une société inondée d'informations de plus en plus complexes. Ce groupe de travail a proposé neuf pistes de recherche afin de produire de nouveau de la confiance numérique et de restructurer la relation entre les organisations (gouvernements, entreprises...) et les individus. Les points suivants présentent les principales pistes proposées.

1. Considérer des tiers de confiance comme pivots d'une confiance mutuelle ;
2. Rendre la confiance transitive ;
3. Faire de la conversation des marchés un actif de la confiance ;

4. Se différencier par sa frugalité en information ;
5. Ré-humaniser la relation client, source de confiance.
6. Accorder sa confiance sans se perdre ;
7. Pérenniser et étendre la confiance « de pair à pair » ;
8. Outiller les individus dans leur relation aux organisations ;
9. Utiliser la transparence raisonnée comme opportunité de différenciation.

Dans le domaine de l'informatique, la confiance se rapporte à des protocoles de sécurité permettant en particulier d'authentifier une source d'information (cf. [9]). La mise en place de la sécurité réduit la prise de risque. Paradoxalement, la mise en place de la confiance numérique en informatique augmente la prise de risque. Il est donc nécessaire de trouver un compromis entre une sécurité minimale optimale et une confiance numérique adéquate.

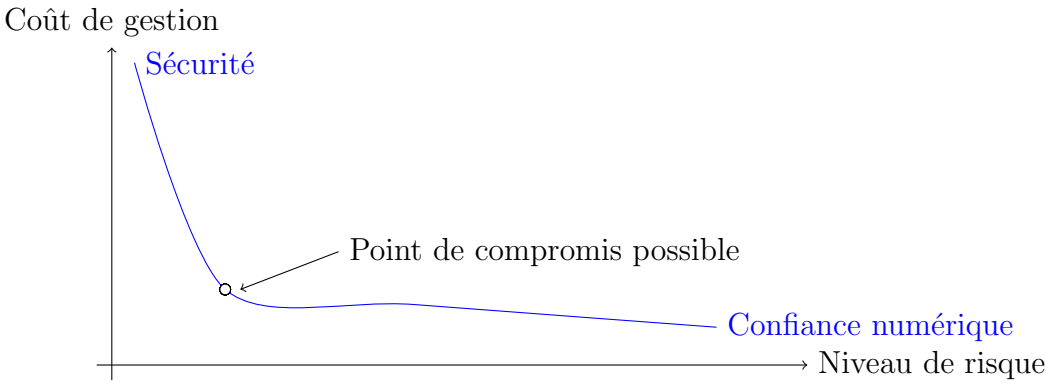


FIGURE 1.2 – L'évolution de la confiance numérique en fonction du risque pris

Le paradoxe de la confiance numérique dans le domaine de l'informatique est représenté sur la figure 1.2. Un point de compromis est choisi en fonction du niveau de risque qu'un système est prêt à prendre et des coûts de gestion associés. En effet, la sécurité a un coût non négligeable dans les systèmes informatiques. La confiance numérique est un moyen de diminuer les coûts liés à la mise en place de systèmes de sécurité.

Dans ce chapitre, nous allons proposer deux éléments permettant de produire de la confiance numérique au cœur d'une application qui propose un ensemble de services en

ligne. Pour créer un environnement de confiance numérique, il faut que le fournisseur de services en ligne ait confiance en l'utilisateur des services et réciproquement. Nous rappelons qu'en informatique, produire de la confiance numérique est équivalent à trouver un compromis entre la confiance décidée et la sécurité, sur laquelle se repose la confiance assurée. Actuellement, le fournisseur de services en ligne a une confiance assurée en l'utilisateur car il a mis en place des systèmes de sécurité développés (utilisation de protocoles cryptés, mots de passe répondant aux règles de sécurité...) qui sont des garanties. L'utilisateur a une confiance assurée envers le fournisseur car il est protégé par des institutions telles que la Commission Nationale de l'Informatique et des Libertés (CNIL) [1]. « Comment produire de la confiance numérique dans un espace de services en ligne et comment la mesurer ? » est la question à résoudre dans ce chapitre. Nous proposons deux solutions : un modèle global de production de la confiance basé sur les travaux de [7] et l'authentification implicite. Les modèles de confiance ont envahi aujourd'hui les services en ligne parce que ce sont des mécanismes de gestion du risque efficaces. Citons les modèles de confiance proposés par eBay, Yahoo Auction ou Amazon. Dans le domaine du e-commerce en particulier, l'objectif de ce type de modèle est d'assister les utilisateurs dans leurs prises de décision lors de leurs interactions. Cette aide est basée sur un comportement passé qui semble prédictif d'un comportement futur.

> **Exemple 1** eBay encourage les acteurs d'une transaction à se donner des notes. Ces notes sont utilisées comme des références pour les prochaines transactions.

Les études que nous avons menées s'inscrivent dans deux pistes de recherche de l'expédition citée plus haut. Le modèle global de confiance permet la transparence raisonnée. En effet, il autorise les utilisateurs à voir comment leur est attribué la confiance en fonction de leur activité. Le but est de développer un modèle de confiance général permettant des prises de décisions optimales, adapté à notre contexte. L'authentification implicite proposée permet aussi d'accorder sa confiance sans se perdre. En effet, elle augmente la convivialité lors de l'utilisation des services tout en préservant un niveau minimum de sécurité.

Ce chapitre se présente de la façon suivante. Dans la section 1.2, une vue d'ensemble des travaux réalisés dans le domaine de la confiance en informatique, les concepts du domaine de recherche ainsi que nos objectifs sont présentés. Ensuite, un modèle de confiance est proposé dans la section 1.3 avant d'introduire l'authentification implicite dans la section 1.4.

1. www.cnil.fr/, accès 10/07/2015

1.2 Définitions

Avant de présenter l'état de l'art sur les modèles de confiance, nous allons d'abord définir les fondements de tout modèle de confiance.

Entités

Nous pouvons identifier trois types d'entités dans un modèle de confiance :

- l'entité qui fait confiance, cette entité peut être un fournisseur de services ou un client (un internaute) ;
- l'entité en qui il fait confiance, généralement un client (un internaute) ou un fournisseur de services ;
- les entités qui vont faire de la recommandation éventuellement (un internaute, un fournisseur de services ou un serveur).

En général, la problématique que l'on retrouve dans la littérature est la suivante. Le potentiel client doit satisfaire des besoins et doit donc choisir des fournisseurs de services à qui il va faire confiance pour répondre à ses besoins. Le modèle de confiance doit faciliter cette prise de décision (cf. [10]).

Types de confiance

Il existe trois types de confiance dans la littérature. La confiance *situationnelle* ou *contextuelle* est définie à moment donné. L'entité se trouvant dans une certaine situation, qui requière différentes considérations par rapport à la confiance n'aura pas la même décision de confiance que si elle se trouve dans une autre situation. Je fais confiance à mon banquier lorsqu'il s'agit de me donner des conseils sur les meilleurs produits et services financiers à acquérir. Quand il s'agit de garder mon fils, je ne lui fais plus confiance. Le deuxième type de confiance est la confiance dite *générale*. La confiance *générale* représente la confiance qu'une entité a envers une autre entité indépendamment d'une situation donnée. Enfin, la confiance *basique* qui est le troisième type de confiance, est la confiance initiale lorsqu'il n'existe aucune relation entre deux entités (cf. [11]).

Représentation de la confiance

Dans un modèle de confiance, il est nécessaire de regarder comment la confiance va être représentée. La représentation peut être qualitative (en définissant des catégories)

ou quantitative (en utilisant des nombres). La représentation qualitative est suffisante pour spécifier une différence entre les catégories mais n'est pas assez riche lorsqu'on veut montrer une différence plus subtile. La représentation quantitative est utilisée dans deux cas. La confiance peut être mesurée par une valeur continue ou par une valeur discrète délimitée par une limite supérieure et inférieure. La confiance est très complexe et très dure à représenter en utilisant une valeur quantitative seulement ou quelques notations qualitatives. Dans tous les cas, la confiance est graduelle. Cette propriété est à l'origine de perte d'informations (cf. [11]) et d'ambiguïté (cf. [12]). A travers l'état de l'art, nous verrons ces deux types de représentation de la confiance.

Rappelons que l'objectif de ce chapitre est la mise en place d'un environnement de confiance au cœur d'une application offrant un ensemble de services en ligne. Nous avons évoqué ci-dessus que dans le domaine de l'informatique, la confiance renvoie en particulier à des technologies permettant de connaitre la source d'une information au sein d'un système informatique (cf. [9]). Cette confiance assurée a fait l'objet de nombreux travaux notamment dans le domaine de la sécurité et du contrôle d'accès dans les réseaux (cf. [7]), de la fiabilité des systèmes distribués (cf. [13]) ou des systèmes d'agents (cf. [14] et [15]) mais aussi dans les politiques de décision dans un contexte incertain (cf. [16]). Particulièrement vaste, le thème de la confiance en science informatique est traité de manière approfondie dans [9], [17] et [18]. Néanmoins nous pouvons identifier quelques axes majeurs sur lesquels des travaux, ayant un lien direct avec notre problématique, ont été menés. Le premier axe regroupe les politiques de confiance. On développera dans le second axe la réputation et les systèmes de recommandation. Les travaux sur la façon de représenter et d'évaluer la confiance en informatique, inspirés de la sociologie et de la psychologie, ainsi que des modèles généraux de confiance, constituent le troisième et dernier axe.

1.2.1 Confiance à base de politiques

On appelle intuitivement « politique de confiance » l'ensemble des processus qui déterminent la confiance en une entité. Ces processus sont capables de répondre aux questions suivantes : quand, pourquoi et comment la confiance est accordée à une entité. Dans [9], les auteurs définissent plusieurs éléments comme étant des politiques de confiance : les systèmes à base d'*identifiants* appelés en anglais *credentials*, la *négociation* de la confiance et aussi les *politiques de sécurité* (cf. [19]).

Les systèmes à base d'*identifiants* dont le plus connu est l'authentification faible (utilisation d'un couple identifiant et mot de passe pour s'authentifier et avoir accès à des ressources) permettent de vérifier qu'un utilisateur a obtenu la confiance assurée d'un système informatique. Des problèmes de sécurité, par exemple l'usurpation d'identité, ont motivé la mise en place d'autres types de systèmes tels que la signature électronique (cf. [20]) qui permet de vérifier l'intégrité d'un document et d'en authentifier son auteur. La mise en place d'une Infrastructure de Gestion des Clés (IGC) est une solution face aux vulnérabilités de sécurité inhérentes à la signature électronique (problème de la diffusion et/ou d'authentification de la clé publique). Le protocole d'authentification réseau de Kerberos (cf. [21]) évite l'interception illicite des mots de passe des utilisateurs et est donc une solution face aux problèmes d'usurpation d'identité.

D'après le TLF, la *négociation* est définie par « des démarches, ou une série d'entretiens pour parvenir à un accord ». En informatique, le mécanisme de négociation de la confiance traité dans [22], [23] et [24], renvoie aux processus permettant d'obtenir la confiance par le respect des règles et des contraintes imposées par une politique de confiance. La négociation est une transaction (échange d'informations) entre des entités (par exemple service et utilisateur du service) dont l'objectif est de permettre d'obtenir un niveau de confiance suffisant (cf. [18]). Dans un contexte donné, la confiance est accordée à condition de révéler au système un certain nombre et type d'identifiants. La confidentialité de ces identifiants est perdue lorsque ceux-ci sont révélés. Il existe des systèmes où il ne suffit de révéler qu'un nombre suffisant d'identifiants pour garder la confidentialité des identifiants.

Les *politiques de sécurité* sont à la base de la plupart des premiers modèles de confiance. Les notions de sécurité et de confiance sont interdépendantes en informatique. La confiance dans les politiques de sécurité passe particulièrement par le contrôle d'accès (authentification, autorisation et traçabilité : cf. [25], [26] et [27]). Avec l'ouverture d'accès aux ressources sur Internet et le nomadisme des employés consistant à leur permettre d'accéder aux ressources à partir de n'importe quel endroit, les entreprises sont confrontées à des problèmes de sécurité importants. Les politiques de sécurité, solutions à ces problèmes ont donc pour objectif la non-répudiation (soit garantir qu'aucun des participants à une transaction ne pourra la nier), l'authentification (soit assurer que seules les personnes autorisées aient accès aux ressources) et la disponibilité (soit maintenir le bon fonctionnement du système d'information) entre autres. Enfin les politiques de sécurité regroupent également les politiques de confidentialité (consistant à assurer que seules les personnes autorisées aient accès aux ressources échangées lors d'une transaction) et d'intégrité (consistant à garantir que les données sont bien celles que l'on croit être) basées sur la cryptographie (cf.[28]).

1.2.2 Réputation et recommandation

La réputation est définie comme une « notoriété qui affecte une personne ou une chose » d'après le TLF. Alors que la recommandation est une « intervention ou un conseil par lequel on introduit une entité auprès d'une autre en attirant son attention ». Dans cette sous-section, nous introduisons la confiance à base de réputation et de recommandation. En effet, la recommandation est un des fondements de la confiance (cf. [29]) et la réputation est une mesure de la confiance. Dans la suite de l'étude, nous appellerons cette confiance, la confiance à base de réputation. La confiance en une entité basée sur la réputation se définit comme une confiance assurée par une connaissance de l'historique des échanges de l'entité, appelée aussi l'expérience (cf. [30], [31] et [32]). En effet, chaque entité peut maintenir en mémoire des informations relatives à la réputation des autres entités. En outre, la confiance à base de réputation dépend du contexte de l'interaction entre les entités. C'est une confiance *contextuelle*. Elle est utilisée dans différents domaines. Par exemple, elle définit la réputation des internautes (cf. [33]) dans le cadre du e-commerce afin d'encourager les interactions et de supprimer des comportements malveillants (cf. [34]). Les mécanismes de la confiance à base de réputation sont aussi utilisés pour augmenter la fiabilité et la performance des interactions dans les sociétés virtuelles comme les réseaux sociaux. Enfin les modèles proposés pour établir ce genre de confiance peuvent être à base de logique floue (cf. [35] et [36]) ou de systèmes bayésiens (cf. [37] et [38]). Dans la suite, nous allons présenter la littérature sur les modèles de confiance à base de réputation centralisés et décentralisés afin de discuter des avantages et inconvénients de ces types de modèles. Des études de la confiance à base de réputation dépendante du contexte sont ensuite citées. Enfin les méthodes de calcul de la confiance à base de réputation sont mises en avant.

1.2.2.1 Système de réputation centralisé et décentralisé

Le site de vente aux enchères eBay [2] met en œuvre un modèle de confiance basé sur la réputation et la recommandation simple et efficace. Ce système est basé sur l'historique des transactions passées de chaque acteur. Après chaque enchère, le vendeur et l'acheteur ont la possibilité de s'évaluer mutuellement en émettant un avis positif, neutre ou négatif sur la transaction. La réputation d'un utilisateur (acheteur ou vendeur) dépend de l'ensemble de ces avis, cette information étant partagée dans le système. Grâce à l'ensemble des avis disponibles, le système peut déterminer le pourcentage de transactions évaluées positivement par les utilisateurs.

2. www.ebay.fr, accès 09/07/2015

Cette information permettra aux utilisateurs (acheteurs ou vendeurs) de prendre la décision de réaliser ou non une vente ou un achat.

Dans le cadre des sites de vente en ligne, il est possible d'utiliser un autre aspect de la réputation : l'appréciation ou la note des produits ou des marchands. Quand le client achève une transaction, il peut la noter sous la forme d'une appréciation (de 1 à 5 étoiles). L'ensemble de ces appréciations permet de disposer de la réputation des produits ou des vendeurs et les futurs clients peuvent consulter ces informations avant de se décider pour un achat (cf. [18]). À présent, ce modèle est appliqué dans presque tous les sites de vente en ligne. Le site de vente en ligne, Amazon [3] par exemple, utilise ce modèle. Ces schémas de réputation sont *centralisés*, toutes les informations concernant la réputation sont gérées par eBay ou Amazon qui connait donc les avis émis par chacun des utilisateurs. Dans un modèle de confiance à base de réputation centralisé, toutes les appréciations ou avis disponibles sont collectés par une autorité centrale qui calcule ensuite un score de réputation global pour chaque participant (acheteur ou vendeur). Ce score est disponible et les participants peuvent l'utiliser comme on l'a expliqué dans le cas d'eBay, pour décider de faire une transaction.

> **Exemple 2** Le processus décrit ci-dessus est illustré par la figure 1.3, tirée de [31]. A et B sont des participants qui peuvent avoir à effectuer une nouvelle transaction sachant qu'ils ont déjà un historique de transactions. Après une transaction, A et B envoient leurs avis à une autorité centrale (schéma à gauche sur la figure 1.3). Cette dernière calcule un score global qui sera éventuellement consulté si A et B veulent à nouveau faire une nouvelle transaction (schéma à droite sur la figure 1.3).

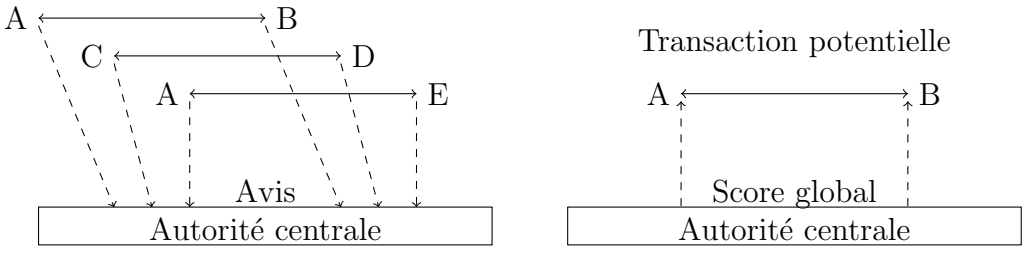


FIGURE 1.3 – Système de réputation centralisé

Dans [7], la confiance à base de réputation est utilisée pour améliorer la détection des intrusions dans les réseaux. Les travaux de thèse présentés dans [7] proposent

3. www.amazon.fr, accès 09/07/2015

d'aider un prestataire de service à produire et à partager, de manière anonyme, un profil de réputation de leurs clients à partir de leurs profils de comportement. Dans ce manuscrit, la réputation est une perception quantifiée, dépendant du contexte applicatif, donnée par un serveur et pour un client. Cette réputation est à prendre dans l'intervalle continu [-1,1] (cf. [33]). Puisque l'anonymat est un gage de la qualité du système de confiance à base de réputation, des systèmes de réputation centralisés sont proposés dans cette thèse. En effet, un unique tiers de confiance centralise les évaluations des réputations de chaque entité, et ensuite partage les évaluations anonymes au besoin (cf. [7]). Cependant, cette solution n'est pas retenue car une attaque du tiers de confiance entraine l'indisponibilité de l'ensemble du système.

Dans la littérature, les systèmes *décentralisés* ont été largement sollicités pour pallier les inconvénients des systèmes centralisés. Dans un système décentralisé, les avis sont échangés entre les participants après chaque transaction. Chaque participant collecte donc un ensemble d'avis dépendant de son expérience et ensuite calcule un score pour chaque participant qu'il a déjà rencontré. Lorsqu'un participant veut connaitre la réputation d'un participant appelé participant *cible*, il fait appel à tous les participants qui ont déjà interagi avec le participant *cible*.

Exemple 3 La figure 1.4 tirée de [31], montre le fonctionnement d'un système de réputation décentralisé. A et B sont des participants qui peuvent effectuer une transaction potentielle sachant qu'ils ont déjà un historique de transactions (des avis sont déjà échangés entre A et B). Pour décider d'effectuer cette transaction potentielle, A demande le score calculé de chaque participant (G et F) qui a déjà interagi avec B. B fait de même avec les participants (C, E et D).

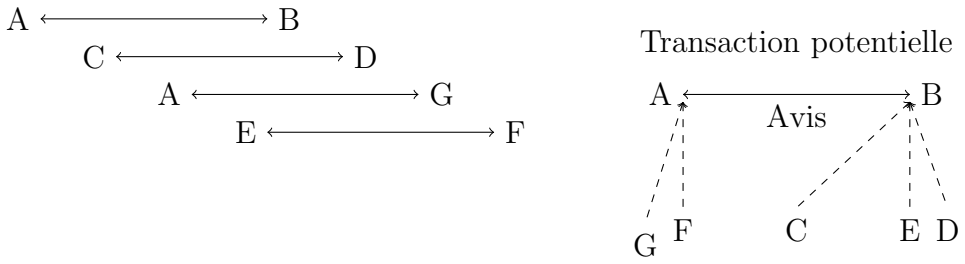


FIGURE 1.4 – Système de réputation décentralisé

Dans ce type de configuration, lorsqu'une entité n'est pas disponible, le système continue de fonctionner. L'inconvénient d'un système décentralisé est la perte de

l'anonymat. Dans [29], des modèles de calcul de la réputation sont décrits à partir de la confiance dans les contenus (mesurée par des notes ou *ratings* comme dans eBay ou Amazon) dans un contexte pair-à-pair. Les systèmes de réputation décentralisés sont par nature adaptés à ce type de contextes. Les modèles proposés prennent en compte la préservation de l'anonymat des notes. Il est rappelé dans [29] qu'un système dans lequel les informations concernant les notes (source, destination et valeur) sont connues de tous n'est pas optimal. La possibilité de rétorsion peut mener à une forme d'auto-censure. Dans les travaux décrits dans [29], les modèles répondent donc à la question suivante : « Comment mesurer la réputation d'une personne (par exemple un expert) à partir des recommandations de ses confrères, en cachant à tous, ce que chacun pense des autres ? »

Dans [12], un modèle de confiance utilisant un système de réputation décentralisé est présenté également. Abdul-Rahman et Hailes proposent un modèle de calcul de confiance basé sur l'expérience et la recommandation dans le contexte pair-à-pair. Chaque entité de la communauté stocke ses valeurs de confiance des entités avec lesquelles elle a eu des contacts (son expérience). Les recommandations des entités au sujet de la confiance en une autre entité permettent d'enrichir leurs propres expériences sur cet individu. Ces valeurs de confiance sont calculées à partir de l'historique complet des interactions.

Enfin, dans [39], différents modèles de réputation décentralisés sont proposés sur la base des interactions entre les agents. Chaque agent du système évalue et stocke la réputation des autres agents avec lesquels il a interagi. Il peut témoigner à propos d'un agent grâce à ses réputations stockées. Les principaux inconvénients de ces approches sont la difficulté à établir des liens solides entre les agents et le processus de recherche de témoins qui est parfois impossible. Un système de réputation hybride avec des caractéristiques centralisées (maintien de l'anonymat) et décentralisées (disponibilité en cas d'attaque) est proposé pour surmonter ces problèmes. Nous utilisons ce type de système dans le modèle de confiance que nous proposons dans la suite.

1.2.2.2 Confiance à base de réputation et contexte

Nous explorons ici la littérature consacrée à l'importance et au rôle du contexte dans les modèles de confiance à base de réputation. Nous nous intéressons particulièrement aux services en ligne.

Le nombre de transactions électroniques sur Internet (achat de biens, réservation des vols ou des chambres d'hôtel) est de plus en plus important. Cependant, il existe d'importantes menaces de sécurité sur Internet. Dans [40], le modèle de confiance et de réputation TRIMS est respectueux de la vie privée et permet de décider si un

contexte (ou domaine) est fiable. TRIMS utilise des éléments de sécurité pour évaluer la fiabilité d'un contexte. L'étude se penche sur les problèmes qui peuvent exister quand un contexte doit décider s'il échange des informations avec un autre contexte inconnu, afin de fournir un service à un de ses utilisateurs. Le modèle proposé permet la gestion de la confiance et de la réputation dans un scénario multi-contextes.

Dans [41], les auteurs étudient la confiance à base de réputation dans un service de messagerie, un contexte plus spécifique et populaire. Le spam interfère avec le courriel valide et alourdit la boite de réception des deux utilisateurs de la messagerie et des fournisseurs de services. Dans [41], un système de filtrage de spam dynamique basé sur la réputation du rapporteur (utilisateur) combiné à des techniques de filtrage de spam existantes est proposé. Le système comporte un composant de mise à jour de la confiance des rapporteurs, basé sur leur comportement. Le système proposé est évalué en utilisant les retours de réclamation réelle d'un grand nombre d'utilisateurs, et ses performances de filtrage de spam sont validées grâce à une collection de courriels réels sur plusieurs semaines. Cette étude montre comment un service de messagerie basique peut s'améliorer en utilisant la confiance à base de réputation.

Dans [42], l'auteur propose des travaux permettant aux fournisseurs de services sur Internet et aux utilisateurs de connaitre la confiance de l'un envers l'autre et réciproquement. Il présente un modèle de prédiction de la fiabilité du comportement de chacun sur Internet, basé sur le comportement passé. Un contexte est associé à un service sur Internet, une entité peut faire confiance de manière différente sur Internet. Tout comme dans [33], dans un contexte donné, la réputation est une perception subjective de la fiabilité d'une entité qui est cependant construite à partir d'informations décrivant le comportement passé de cette entité. La confiance est une fonction qui renvoie une valeur de l'intervalle continu [0,1], traduisant le degré de confiance. La relation de confiance n'étant pas statique, le temps est introduit dans l'étude de [42]. La mise à jour du degré de confiance à base de réputation est effectuée en fonction des nouvelles interactions. Le contexte de cet article est proche de notre contexte d'étude comme on le verra dans la suite. Le contexte est très important lorsqu'il s'agit de mettre en place des modèles de confiance à base de réputation. En effet, il permet d'affiner une métrique qui représente la confiance.

1.2.2.3 Calcul de la confiance à base de réputation et prise en compte de paramètres

Dans [16], Audun Josang propose une méthode originale, dite logique subjective pour modéliser des systèmes de réputation (cf. [43]). Il présente une méthode basée sur la logique de croyance subjective de propositions incertaines. Le composant de

base de cette logique est l'*opinion*. Une opinion est un quadruplet qui contient les éléments suivants :

- b - la croyance en la proposition ;
- d - l'incrédulité ou *disbelief* en la proposition ;
- u - l'incertitude de la proposition ;
- a - l'atomicité relative de la proposition.

Ces valeurs sont de type numérique et sont liées entre elles par la relation suivante : $b + d + u = 1$. La probabilité d'espérance est égale à $b+a\times u$. Le calcul de la confiance sur un contexte incertain est réalisé en raisonnant sur un chemin défini en terme de probabilité sur des événements incertains. Le système ne permet pas d'analyser et de faire des calculs sur la confiance si les chemins ne sont pas inter-connectés. Audun Josang énonce des métriques pour calculer la réputation dans le modèle de confiance.

Dans [10], un modèle général de confiance basé sur des modèles stochastiques de Web services est proposé. Dans un contexte de Web services et de transactions en ligne, les auteurs de [10] expliquent comment appliquer des méthodes statistiques pour construire un modèle de confiance.

Dans [44], les auteurs présentent et évaluent un modèle consacré au système de réputation distribué (ou décentralisé). Les auteurs proposent dans cet article une approche globale et considèrent un certain nombre de questions qui ont une incidence sur la confiance et la réputation comme l'ancienneté des notes, la valeur transactionnelle, la crédibilité des arbitres ou tiers de confiance, le nombre d'incidents malveillants, la complicité et les notes injustes. L'étude étend également son approche en considérant d'autres aspects relatifs à la caractéristique des fournisseurs de services, tels que l'existence de labels de confiance, les intermédiaires de paiement, les déclarations de confidentialité, les stratégies de sécurité et de confidentialité, la protection et l'assurance à l'achat ainsi que le règlement extra-judiciaire des différends. Ces paramètres doivent être pris en compte dans les modèles de type eBay ou Amazon.

L'état de l'art nous montre qu'il existe différents outils mathématiques pour calculer une métrique qui représente la confiance. Plusieurs paramètres peuvent être pris en compte également lorsqu'il s'agit de calculer la confiance.

1.2.3 Confiance et sciences sociales

Dans cette sous-section, les modèles de confiance de manière générale ainsi que certaines propriétés de la confiance sont présentés. Les modèles généraux de confiance

font appel à des aspects de la psychologie et de la sociologie (cf. [45]). Dans [46], les auteurs proposent une classification des différents types de confiance (*trust*) en sciences sociales. Dans le travail cité dernièrement, les auteurs identifient quatre éléments indispensables à analyser lors de la prise de décision basée sur la confiance : l'intégrité, l'expertise (capacité à donner des informations exactes), la bienfaisance (se donner la peine de faire un effort) et la prédictibilité. D'autres qualités inhérentes à la confiance sont présentées dans le domaine du *management* : la congruence (le fait de faire coïncider les actions avec les affirmations), la fiabilité, la franchise, la tolérance, et la sensibilité. Même la notion *d'intuition*, basée sur la similarité des profils des utilisateurs, est prise en compte pour déterminer la confiance quand on ne dispose en mémoire d'aucune action passée de l'utilisateur, dans un système multi-agent (cf. [47]). Il est possible de rapprocher l'*intuition* à la confiance *basique* de [11]. Nous aborderons la *réciprocité* qui est un composant important de la relation de confiance, avant de présenter le *modèle de confiance général fondateur* de Stephen Paul Marsh, formalisé dans sa thèse soutenue en 1994.

La *réciprocité* est l'échange mutuel d'actions (par exemple, comme une faveur ou une vengeance). La transitivité de la confiance est un phénomène commun intégré dans le raisonnement humain sur la confiance (cf. [16] et [48]). La réciprocité et la transitivité de la confiance sont aussi des concepts-clés de notre problématique (cf. [49], [50], [51], [52] et [53]). Dans un contexte spécifique, la transitivité de la confiance se manifeste souvent par l'intuition des êtres humains à se fier aux recommandations d'une entité digne de confiance B sur une autre entité recommandée C. Bien qu'intuitive, la transitivité de la confiance est un phénomène humain qui semble particulièrement complexe. Compte tenu de cette complexité, n'importe quel modèle mathématique simple aura un pauvre pouvoir prédictif. Dans [16], une interprétation de la transitivité de confiance radicalement différente de celles décrites dans la littérature jusqu'à présent est proposée. Plus précisément, elle considère les recommandations de l'entité digne de confiance B comme une preuve que l'entité requérante A utilisera comme arguments d'entrée dans les modèles de raisonnement conditionnel pour évaluer des hypothèses sur l'entité recommandée C. Le modèle proposé de la transitivité de confiance conditionnelle est basé sur le cadre de la logique subjective.

La probabilité subjective est le fondement du modèle proposé dans [12]. La valeur de la confiance est discrète et appartient à l'ensemble {-2, -1, 0, 1, 2} qui correspond respectivement à {*very untrustworthy, untrustworthy, neutral, trustworthy, verytrustworthy*} [4]. L'objectif des travaux décrits dans [12] est de fournir un modèle de confiance basé sur des propriétés des sciences sociaux. La confiance n'est pas une propriété

4. {*très indigne de confiance, indigne de confiance, neutre, digne de confiance, très digne de confiance*}

objective d'un agent mais une croyance subjective à propos d'agents. Les auteurs soulignent la différence entre la recommandation et la perception propre de celui qui fait confiance par la mise en place de ce qu'ils appellent « distance sémantique ».

L'étude de la confiance a progressé énormément ces dernières années, dans la mesure où l'objectif d'une formalisation plus unifiée de la notion est faisable. À cette fin, les auteurs de [54] ont commencé à examiner les concepts de *regret* et de *pardon* et leur relation avec la confiance. En outre, le regret et le pardon, lorsqu'ils sont alliés à la confiance, sont des outils très puissants dans le domaine de la sécurité *Ambient Intelligence (AMI)*, en particulier en *Human Computer Interaction (HCI)*, là où la compréhension humaine concrète est importante. Dans [54], les auteurs présentent les concepts de regret et de pardon, leur exploitation à partir de la psychologie sociale et introduisent une extension à la formalisation originelle de la confiance donnée dans [11] qui les prend en compte. La représentation de la confiance et la prise de risques associés ont fait l'objet de plusieurs interrogations dont les solutions sont apportées par les *travaux de Marsh* (cf. [11]). Dans sa thèse, Marsh propose un *modèle général de gestion de la confiance* dans le cadre de la coopération qui utilise la notion de réputation. Il définit trois sortes de réputations : basique, générale et contextuelle. Ces trois réputations sont représentées par des valeurs appartenant à l'intervalle [-1,1]. Dans ces travaux, il suppose que si la confiance aveugle existait, alors le fait de chercher à estimer la réputation d'une entité ne serait plus nécessaire. Il considère donc que la confiance aveugle n'existe pas et retire la valeur 1 du domaine des valeurs possibles pour les réputations. Il ne précise pas d'où provient l'information qui permet de calculer les réputations, ni comment les réputations sont initialisées. Il s'intéresse ensuite principalement aux processus de décision. Les réputations sont calculées par rapport à l'ensemble des interactions. Chaque décision est prise par seuillage : si la valeur de réputation contextuelle est supérieure au seuil de coopération, alors l'entité coopère sinon elle ne coopère pas. Marsh a proposé également plusieurs définitions de ce seuil de coopération en fonction des risques, de la compétence de la cible et de l'importance de la situation perçue par le bénéficiaire. Un modèle manipulant des réputations fondées sur des interactions directes entre bénéficiaires est proposé dans ce système. Ces réputation sont toutes subjectives, graduées et non transitives. Marsh est le premier à donner une formalisation du concept de confiance. Les fondements de son modèle sont très sociologiques. Son modèle reste relativement complexe et abstrait, il ne peut être utilisé dans le domaine du e-commerce aujourd'hui. Les processus de décision dans le modèle décrit dans [11] sont indépendants de l'humain. Par opposition dans [55], l'intervention de l'humain est discutée. Certains aspects des modèles de confiance comme son déploiement, son utilisation et *l'abus* sont examinés. Le calcul de la confiance formalise les processus de confiance chez l'homme afin de

permettre à des systèmes artificiels de mieux prendre des décisions ou de mieux donner des conseils. Cela est possible parce que la confiance est flexible, facile à comprendre, et relativement robuste. Depuis son introduction dans les années 90, elle a gagné en popularité en raison de ces caractéristiques. Cependant, ce qu'elle a souvent perdu, c'est l'intelligibilité. L'un des objectifs initiaux du raisonnement du calcul de la confiance était l'élément humain : la participation d'êtres humains dans le processus de prise de décision pour les outils, surtout au niveau de la compréhension des raisons de ces décisions. La prolifération des modèles de plus en plus complexes peut servir à augmenter la robustesse de la gestion de la confiance face à l'attaque, mais fait peu pour aider les humains à simplement comprendre ou, si nécessaire, intervenir lorsque les modèles de confiance échouent ou ne peuvent pas arriver à une décision raisonnable.

Résumé

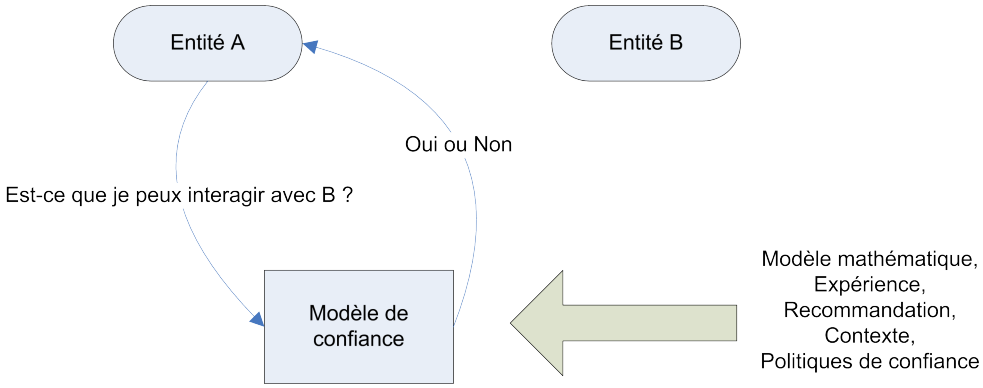


FIGURE 1.5 – Processus de confiance

Pour résumer, la production de confiance passe par la mise en place d'un modèle de confiance général. Les entités en interaction vont décider de faire confiance ou pas en fonction du contexte, de l'expérience et de la recommandation en se basant sur ce modèle, comme le montre le schéma de la figure 1.5. Le modèle de confiance est un modèle mathématique qui repose sur une architecture centralisée ou décentralisée et qui permet d'avoir une métrique qui représente la confiance qu'une entité a envers une autre entité. Différentes considérations sociales (réciprocité, intuition...) mais aussi des paramètres permettent d'affiner le modèle de confiance et de le rendre plus précis. Enfin, la prise en compte de politiques de confiance permet également d'avoir un modèle plus robuste et plus réel.

1.2.4 Contraintes et qualités du modèle à concevoir

La confiance est un domaine récent de recherche dont il existe plusieurs sous-domaines à explorer. Dans ce chapitre, nous tentons de répondre à la problématique de production de la confiance dans un ensemble de services réunis dans une application en examinant la question suivante : *comment un modèle de confiance hybride s'appuyant sur un modèle de réputation préservant la vie privée des utilisateurs et basé sur l'historique de comportement peut améliorer la sécurité de l'accès aux services en ligne en contrôlant l'accès aux fonctionnalités des dits services ?*

Dans la suite, nous allons décomposer cette problématique en différents objectifs que le modèle proposé doit atteindre. Bien que la production de confiance s'accompagne d'une prise de risque, l'étude ne quantifie pas cette prise de risque. La prise de risque associée à la confiance produite pourra être une suite de ce travail. Les points suivants sont les objectifs et contraintes du modèle que nous proposons.

Un modèle de confiance hybride est un modèle de confiance qui utilise un système de calcul de la confiance centralisé et décentralisé. La mise en place de ce type de modèle permet d'éviter les inconvénients des systèmes centralisés vus dans l'état de l'art. Le but de ce travail est donc de proposer un modèle de confiance qui ait les avantages des systèmes centralisés et décentralisés (système disponible même en cas d'attaque d'une entité). Comme nous l'avons vu dans la littérature, l'anonymat est un gage de qualité des modèles de confiance. *Un modèle de réputation préservant la vie privée* est nécessaire. Un service ne doit pas connaitre l'origine de la réputation d'un utilisateur. Dans [7], l'auteur propose le partage d'une réputation globale qui résulte de l'agrégation de réputations locales. Les réputations locales doivent être fiables. Le calcul des indices mesurant les réputations doit être irréversible.

L'étude du comportement de l'utilisateur dans le domaine du e-commerce est un sujet de recherche vaste. Le comportement de l'utilisateur est défini par un historique d'actions et d'échanges. Le comportement de l'utilisateur est souvent représenté par un indice de l'intervalle [-1,1[. Le bon comportement est représenté par 1 et le mauvais comportement par -1. L'objectif est de construire des profils utilisateurs simples et fiables. La confiance calculée se base sur le profil des utilisateurs ou des entités. Les profils doivent être mis à jour en temps réel et permettre l'identification d'un utilisateur dans 80% des cas. Nous nous sommes imposés cet

objectif.

L'accès aux niveaux de service contrôlé par des niveaux de confiance relève de la politique de l'organisme qui va mettre en place ce modèle. Des experts doivent analyser les criticités des fonctionnalités en fonction des services concernés. Ensuite, ils doivent les catégoriser. L'attribution des niveaux de confiance à chaque catégorie est enfin étudiée. Cette partie est un paramètre du modèle.

1.3 Exemple de modèle de confiance

Afin de faciliter l'utilisation des services en ligne et de minimiser les coûts transactionnels, l'entreprise avec laquelle nous avons travaillé a décidé d'intégrer le concept de confiance et de prise de risque associée dans les processus de décision d'accès à ses services. L'objectif à terme est de proposer un environnement propice à la création de confiance réciproque entre le système et ses utilisateurs. Dans cette section nous allons présenter le contexte et le cahier des charges de l'étude menée, ensuite proposer une modélisation du problème et enfin présenter les limites du modèle.

1.3.1 Contexte de l'étude

Notre contexte d'étude est un espace de vie numérique [5] qui offre un ensemble de services tels qu'un coffre-fort numérique, du commerce électronique ou encore des communautés virtuelles. Chaque service est constitué d'un ensemble de fonctionnalités. Une fonctionnalité est une action simple. Par exemple, dans un service d'e-commerce, *valider son panier* est une fonctionnalité. Cet espace de services comprend des fonctionnalités *critiques*, dont la mauvaise utilisation peut nuire au système ou au fournisseur de services. Par exemple, dans un service bancaire en ligne, la fonctionnalité *faire un virement à l'étranger* est une fonctionnalité *critique*. Ces fonctionnalités *critiques* ne doivent être accessibles que par un utilisateur autorisé, c'est-à-dire ayant un niveau de confiance suffisant.

Un niveau de confiance mesure la confiance accordée à un utilisateur en fonction d'un niveau de risque accepté par le fournisseur de service. Le niveau de confiance de chaque utilisateur est calculé en se basant sur le *comportement* de l'utilisateur mais aussi sur son identité, sa réputation et ses recommandations. Dans cette étude, nous

5. www.ebeeoffice.ca/ebee-home/public, accès 09/07/2015

définissons le *comportement* de l'utilisateur comme l'ensemble des fonctionnalités qu'il a utilisé lors de ces connexions sur l'espace de vie numérique.

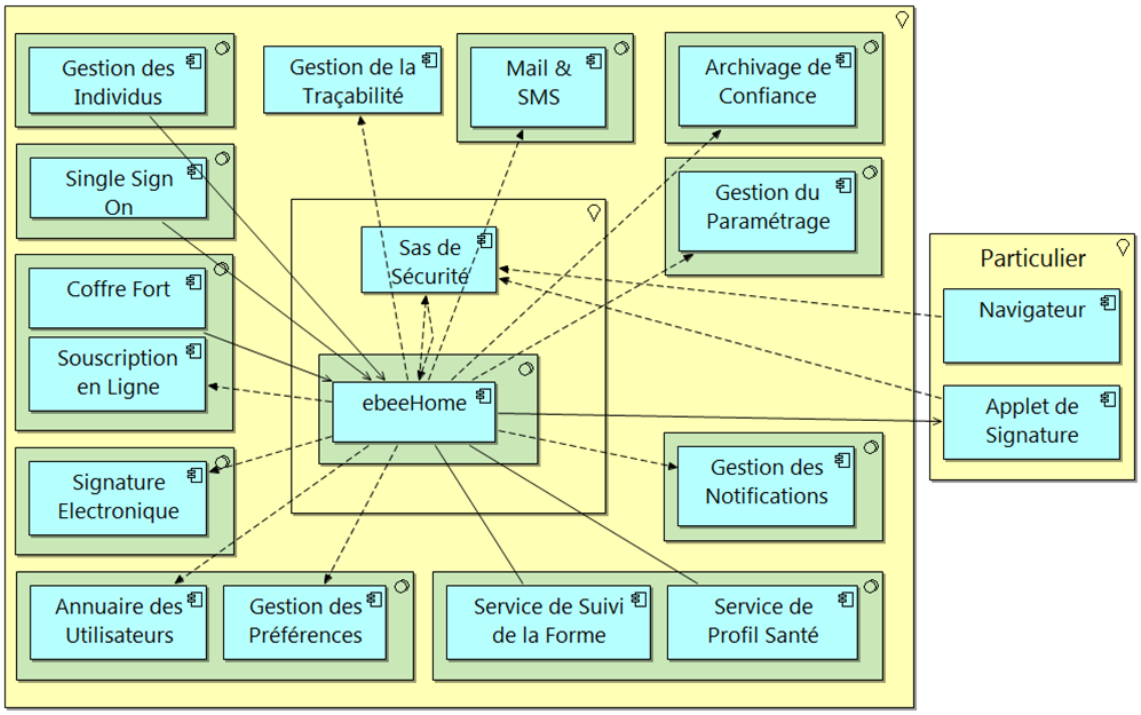


FIGURE 1.6 – eBeeHome, un espace de vie numérique

Cet espace, appelé eBeeHome, se présente comme un frontal. La figure 1.6 montre le schéma d'architecture de l'espace de vie numérique. Sur la figure 1.6, la gestion des individus permet d'ajouter, de modifier ou de supprimer un nouvel utilisateur après avoir paramétré le service avec ses droits d'accès. Avec le service Single Sign-On (SSO), les utilisateurs ne s'identifient qu'une seule fois pour accéder à l'intégralité des services proposés. Le service coffre-fort permet de stocker et de gérer des documents personnels de manière sécurisée. La souscription en ligne offre la capacité à l'utilisateur de s'enregistrer sur le service, moyennant différents facteurs (par le biais d'une entreprise, en payant le service...). La signature électronique permet à un utilisateur de signer un document à destination d'un tier. L'annuaire de service liste les utilisateurs et leurs relations pour échange d'informations ou documents. La gestion des préférences enregistre les préférences d'utilisation du service pour chaque utilisateur. La gestion des notifications permet d'indiquer à tout moment à un utilisateur un évènement le concernant (par ex : un nouveau document est disponible dans votre coffre de

la part de votre fournisseur d'électricité). Le service de suivi de forme et le service profil santé sont deux services pour le suivi de la santé de l'utilisateur. L'archivage de confiance permet de stocker à vie des documents dans le coffre. Les services sms et mail permettent l'envoi d'information à l'utilisateur par d'autres canaux.

Rappelons que l'originalité de la première partie de cette thèse se trouve d'abord dans l'étude du comportement (ensemble des fonctionnalités demandées) d'un utilisateur sur un espace de vie numérique donc dans un contexte multi-services. Notre travail devrait permettre de définir ce qu'est un accès sécurisé en fonction d'un niveau de confiance attendu. Cette confiance est à la limite entre une confiance assurée et une confiance décidée. Nous proposons un modèle de confiance inspiré de la littérature mais adapté à notre espace numérique (cf. [7]). L'étude devra permettre notamment de répondre à la question suivante : « Tel utilisateur doit-il avoir accès à telle fonctionnalité *critique* ? »

Le modèle proposé a pour vocation de rationaliser le risque encouru par l'ouverture d'un accès à une fonctionnalité d'un service pour un utilisateur donné. Pour cela, nous proposons un ensemble de processus pour la construction d'un niveau de confiance que le système peut fournir à un utilisateur émettant une requête d'accès à une fonctionnalité *critique*. Enfin, en fonction de la requête et en tenant compte du niveau de confiance, le système devra prendre la décision de donner ou non accès à cette fonctionnalité.

La confiance accordée à un utilisateur suit plusieurs stades de développement. Lorsque l'utilisateur utilise l'espace de vie numérique pour la première fois, le système ne *connait* pas l'utilisateur. Cependant, le système lui accorde une *confiance initiale décidée* ou une *confiance a priori* appelée dans la littérature confiance *basique* (cf. [11]). En effet, le système prend des risques en ouvrant l'accès à un ensemble de fonctionnalités dites *basiques* de chaque service. Il existe un niveau d'autorisation minimal pour accéder aux fonctionnalités *critiques* de chaque service. Des experts définiront les fonctionnalités dites *basiques* et celles dites *critiques*. Dans le groupe des fonctionnalités *critiques*, il existe différents niveaux de criticité. Ces niveaux sont dépendants du risque comme expliqué ci-dessus, mais aussi des types de données (sensibles ou personnelles) auxquelles la fonctionnalité va ouvrir l'accès. Les politiques d'entreprise définiront les niveaux de confiance requis pour accéder à ces fonctionnalités *critiques*. Ces politiques peuvent être « optimistes », c'est-à-dire qu'elles sont optimistes par rapport aux comportements des utilisateurs et donc les décisions devraient pousser les utilisateurs à améliorer leurs comportements, ou « pessimistes » soit les décisions prises sont très restrictives.

Le système observe l'utilisateur et doit calculer un niveau de confiance, ensuite il met à jour le niveau de confiance. Enfin, le niveau de confiance est réinitialisé lorsque

l'utilisateur ne s'est plus connecté au système depuis un certain temps, déterminé par le fournisseur. Par la suite, nous allons détailler comment ces phases sont étudiées après avoir défini les *données* de l'étude et formalisé les *définitions* importantes du modèle.

Le modèle de confiance repose sur la fouille du journal d'activités qui sont les *données* de l'étude. Lorsque l'utilisateur se connecte à l'espace de vie numérique, toutes les actions qu'il effectue sont tracées par une brique de traçabilité. Cette dernière a pour objet de collecter les *événements* et de les consolider. Le *journal* est un fichier qui contient l'ensemble des enregistrements séquentiels des événements affectant l'espace de vie numérique que nous appelons aussi *comportement*. Une trace est un comportement et doit comporter les éléments suivants : identifiant unique de la trace, date, digest comportant l'empreinte totale de la trace, auteur, fonctionnalité ou *événement*, service, description. On pourra ajouter, le lieu ainsi que la réponse lors de l'appel à l'événement. Les traces sont scellées régulièrement afin de garantir leur valeur de preuve qui pourra être utilisée par le service juridique. A tritre d'exemple, la table 1.1 présente les champs constituant la trace d'un événement de *connexion*.

Une trace peut comporter des informations complémentaires appelées méta-données de session.

Afin de faciliter la compréhension de la suite de ce chapitre, définissons les termes utilisés dans le modèle de confiance proposé.

Définition 1.3.1. ***Une session*** commence lorsque l'utilisateur se connecte au système et se termine dès qu'il se déconnecte du système ou que la session arrive à expiration.

Définition 1.3.2. ***L'historique intra-session*** définit l'ensemble des données de la session courante.

Définition 1.3.3. ***L'historique inter-session*** définit l'ensemble des historiques intra-session. Un historique intra-session est agrégé sous forme d'indice et résume une session.

Définition 1.3.4. ***Le comportement-session*** est extrait des *logs* consolidés par la brique de traçabilité. Cette notion désigne l'ensemble des actions effectuées par l'utilisateur sur le système, durant une session.

Définition 1.3.5. ***Les méta-données de session*** sont les méta-données de connexion à savoir : le mode (mobile ou poste de travail), la date et l'heure, le lieu et la fréquence de connexion sur une journée par exemple. La durée de la session

Nom de champs	Définition
service_SSO	Le service sollicité
event_CONNEXION	Le type d'événement à tracer par la brique de traçabilité
id_user	Identifiant unique de l'utilisateur contenu dans le jeton *Single Sign-On* (SSO)
id_trace	Identifiant unique de la trace
digest	Empreinte de la trace
descr	La fonctionnalité de connexion est une fonctionnalité *basique*
date_event	Date et heure à laquelle la connexion a été faite

TABLE 1.1 – La trace de *connexion* de la brique de traçabilité

ainsi que le mode de déconnexion (bouton déconnecté ou expiration de la page) sont aussi des méta-données session.

Définition 1.3.6. ***Une requête*** est une demande formulée par l'utilisateur pour accéder à une fonctionnalité d'un service donné. Une requête prend la forme d'un triplet (fonctionnalité, service, criticité).

Définition 1.3.7. ***Un niveau de service*** est l'ensemble des droits d'accès à des triplets (fonctionnalité, service, criticité) qu'un utilisateur possède. Les niveaux de service dépendent des niveaux de confiance.

Définition 1.3.8. ***Un niveau de confiance*** traduit le degré de confiance que le système peut accorder à un utilisateur. Ce niveau de confiance est associé à un niveau de service.

On discutera de la représentation d'un niveau de service et d'un niveau de confiance dans la suite.

Définition 1.3.9. ***L'indice d'identité*** donne le degré de certitude que les triplets (fonctionnalité, service, criticité) ont en l'identité de l'utilisateur. Cet indice appartient à l'intervalle continu [-1,1[. On exclut 1 de l'intervalle car on estime qu'on n'est jamais sûr de l'identité d'un utilisateur (cf. [11]).

Définition 1.3.10. ***L'indice de réputation*** traduit la réputation que les triplets (fonctionnalité, service, criticité) ont pour un utilisateur. Les réputations dépendent du contexte. Cet indice appartient à l'intervalle continu [-1,1[. On exclut 1 de l'intervalle pour les mêmes raisons que pour l'indice d'identité.

Définition 1.3.11. ***L'indice de comportement*** caractérise le comportement de l'utilisateur dans un service donné. Cet indice appartient à l'intervalle continu [-1,1[. On exclut 1 de l'intervalle pour les mêmes raisons que pour l'indice d'identité.

On note que les indices sont quantitatifs et continus. C'est un choix de modélisation inspiré de la littérature (cf. [11]).

Définition 1.3.12. ***Une confiance instantanée*** est une confiance qu'un service attribue à un utilisateur, au cours d'une session. Cette confiance est la confiance en cours de l'utilisateur (elle est calculée au fur et à mesure de l'interaction entre l'utilisateur et le système lors d'une session). L'agrégation des confiances instantanées sur une session est la confiance instantanée session.

Définition 1.3.13. ***La confiance continue*** est un historique des confiances instantanées session. Elle évalue la confiance que l'on peut accorder à un utilisateur depuis qu'il est connu du système, sans tenir compte des recommandations. Cette confiance est modulée par une fonction d'atténuation au cours du temps. Les confiances instantanées les plus anciennes sont moins prises en compte que les confiances instantanées les plus récentes.

Définition 1.3.14. ***La confiance globale*** est la confiance continue qu'un service attribue à l'utilisateur en prenant en compte les recommandations des autres services qui ont connu l'utilisateur. Cette prise en compte dépend de la confiance inter-services.

Définition 1.3.15. ***La confiance inter-services*** est la confiance qu'un service accorde à un autre service du système. Elle dépend de la politique de confiance mise en place entre les services du système.

Définition 1.3.16. ***L'indice d'utilisation*** mesure la fréquence d'utilisation d'une fonctionnalité. Cet indice est défini par le nombre d'appels à cette fonctionnalité sur le nombre de fonctionnalités appelées pendant un certain temps paramétrable. L'indice est défini dans l'intervalle [0,1]. L'indice est à 0 si la fonctionnalité n'est jamais utilisée et 1 si c'est la seule appelée pendant ce temps paramétrable.

Définition 1.3.17. ***L'indice de stress*** mesure la quantité de demandes d'accès à une fonctionnalité qui répond par la négation. Cet indice est défini par le nombre d'échecs sur le nombre de tentatives pendant un certain temps paramétrable. L'indice est défini dans l'intervalle [0,1]. L'indice est à 0 si la fonctionnalité est toujours accessible et 1 si toutes les sollicitations à la fonctionnalité ont une réponse négative.

Définition 1.3.18. ***L'indice d'annulation*** se base sur l'étude du *workflow*[6], et plus précisément sur la présence de fonctionnalités et de leurs inverses, pendant un certain temps paramétrable. Par exemple, *ajouter un document et supprimer un document, en deux secondes*. L'indice d'annulation est défini par le nombre de fonctionnalités annulées sur le nombre de fonctionnalités appelées pendant un certain temps. L'indice est défini dans l'intervalle [0,1]. L'indice est à 0 s'il n'y a aucune fonctionnalité annulée et 1 si toutes les fonctionnalités ont été annulées pendant ce temps paramétré.

1.3.2 Cahier des charges : les usages du modèle

En informatique, un cahier des charges fonctionnel est un document qui permet de formaliser les besoins d'un client. Il permet d'expliquer l'utilité du produit final. Les principales fonctions des services rendus ainsi que les contraintes y sont détaillées. Ici, nous présentons le cahier des charges pour permettre aux lecteurs de comprendre l'objectif et les limites du modèle de confiance.

Dans cette étude, il existe trois acteurs : l'utilisateur du service, le service et le fournisseur du service.

Définition 1.3.19. ***Un fournisseur de service*** est une entreprise qui propose l'accès à un service en ligne gratuitement ou moyennant un abonnement.

Définition 1.3.20. ***Un service*** est un programme informatique de type web permettant la communication et l'échange de données entre applications et systèmes hétérogènes dans des environnements distribués. Un service est un ensemble de fonctionnalités présentées sur Internet ou sur un intranet, par et pour des applications ou machines, sans intervention humaine, de manière synchrone ou asynchrone.

Définition 1.3.21. ***Un utilisateur du service*** est un internaute aussi appelé client qui accède aux fonctionnalités de ce service sans authentification ou avec une authentification plus ou moins forte.

6. On appelle un workflow, le flux des activités soit des fonctionnalités demandées par un utilisateur.

Nous présentons les cas d'utilisation en fonction de chaque acteur du système.
En tant que service, je veux pouvoir :

- Attribuer une *confiance a priori* à chaque nouvel utilisateur en utilisant des données non comportementales (données personnelles...) pour initialiser la relation de confiance. Dans [7], il définit des règles pour résoudre ce problème dit *démarrage à froid*, et met le niveau de confiance de tous les nouveaux clients à 0 (appelé la valeur *bootstrap*). Le niveau 0 correspond à de l'ignorance. Nous proposons de donner un minimum de confiance à chaque nouveau client pour lui montrer que le service lui fait confiance. Nous prenons un minimum de risque (ici on a une confiance décidée) et espérons que l'utilisateur tiendra compte de cette confiance accordée a priori. On fait appel ici à la notion de *réciprocité*.

- Faire confiance sans me perdre. La thèse de [7] ne prend pas en compte la criticité de la requête du client. Le serveur traite toutes les demandes de la même manière. Il attribue un niveau de service en fonction du niveau de réputation du client sans tenir compte du niveau de risque de la demande du client. Par exemple, *un prêt de 100 000 euros et de 1 000 euros ne peuvent pas être traités de la même manière, le banquier fera plus attention aux conditions de prêt et à la réputation du client pour un montant important*. Dans notre étude, la confiance doit être mise à jour en tenant compte du comportement de l'utilisateur mais aussi des criticités des fonctionnalités utilisées. Ce point est réalisable en satisfaisant les besoins suivants :

  - Analyser le comportement de l'utilisateur lors d'une session pour lui attribuer une confiance instantanée session.
  - Détecter un comportement suspect de l'utilisateur pour dégrader sa confiance instantanée session. Il est nécessaire d'assurer une réaction rapide à de mauvais comportements et une réaction plus mesurée pour les bons comportements (cf. [7]).
  - Utiliser l'historique de toutes les confiances instantanées d'un utilisateur pour lui donner une confiance continue. Dans la thèse de [7], la confiance continue est une composante de la réputation globale. En effet, l'ensemble des réputations locales (qui correspondent à l'ensemble des confiances instantanées session dans notre étude) est envoyé au *Global Reputation Analyser* (GRA), par les serveurs qui ont interagi avec le client. Le GRA analyse, entre autres, un historique de réputation de la part de chaque serveur ayant interagi avec le client, et prend en compte la *confidence* entre les serveurs.

  - Utiliser la confiance continue attribuée par un autre service (si l'utilisateur utilise un autre service du système) et sa propre confiance continue calculée pour lui donner une confiance globale. Le GRA de [7] gère cet aspect en utilisant une « matrice de *confidence* » des serveurs. La matrice donne le degré de confiance entre les serveurs. L'étude différencie bien *trust* et *confidence* (les serveurs utilisent une infrastructure de gestion de clés ou PKI).
  - Utiliser un système de recommandation d'utilisateur, entre les services pour donner une confiance globale. Un service *messagerie* pourra recommander un utilisateur qui s'est bien comporté à un autre service *coffre-fort*. Cette recommandation permettra de moduler la *confiance a priori* que le service *coffre-fort* devra attribuer pour initialiser la relation de confiance. Dans [7], la recommandation d'un client à un serveur n'est pas étudiée.

- Atténuer la réputation et moduler la confiance d'un utilisateur avec le temps. Si un utilisateur est enregistré depuis vingt ans, on peut négliger son comportement initial. On fait appel à la notion de pardon évoquée dans 1.2. De plus, des organismes comme la CNIL [7] chargés de la protection des données personnelles, réglementent la durée de conservation des données.

- Supprimer la confiance calculée lorsque l'utilisateur ne s'est pas connecté pendant un certain temps pour respecter les recommandations de la CNIL.

- Valoriser un utilisateur en lui donnant accès à des droits particuliers pour le fidéliser et lui montrer une certaine gratitude lorsque son niveau de confiance est satisfaisant. Le serveur donne accès à des niveaux de services supérieurs en fonction du comportement du client dans la thèse de [7]. Ces niveaux de service sont en fonction uniquement de la réputation globale du client. Ces attributions de niveau de service pourraient aussi être en fonction de la classe de l'utilisateur, comme le suggère l'auteur de cette étude de référence, Anirban Basu.

- Dévaloriser un utilisateur pour le « punir » et l'inciter à changer de comportement.

Le diagramme de cas d'utilisation de la figure 1.7 résume les cas d'utilisation du point de vue du service. Ce diagramme est général. Pour atténuer une réputation, il faut au préalable en construire une. Le service attribue une confiance basée sur la

7. www.cnil.fr/, accès 10/07/2015

réputation après l'avoir initialisée par une confiance a priori. Enfin, pour supprimer le confiance, il est nécessaire qu'elle existe.

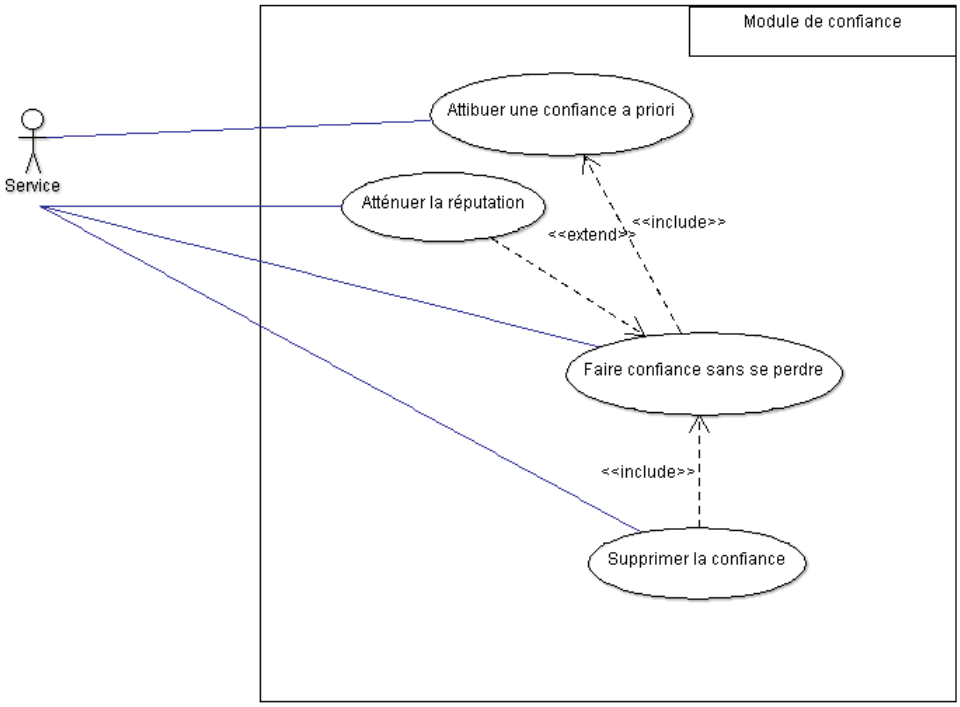


FIGURE 1.7 – Diagramme de cas d'utilisation du modèle de confiance

En tant qu'utilisateur d'un service, je veux pouvoir :

- Bénéficier des services d'un espace riche en toute sécurité et en toute confiance ;
- M'identifier et m'authentifier sur le système pour accéder à l'ensemble des services. L'utilisateur doit être reconnu du système comme étant le propriétaire légitime du compte. Un intrus qui possède les identifiants de l'utilisateur (vol ou prêt d'identifiants) doit être reconnu comme un intrus ;
- Avoir des autorisations *a priori* pour utiliser les fonctionnalités *basiques* de tout service ;
- Bénéficier de droits particuliers pour un service donné (être privilégié) lorsque mon comportement est satisfaisant ;

- Voir mon niveau de confiance et le comprendre.

En tant que fournisseur de service, je veux pouvoir :

- Attribuer des niveaux de criticité aux fonctionnalités de mes services pour relativiser l'impact du comportement de l'utilisateur sur système (sur une échelle absolue ou relative) ;
- Définir les niveaux de confiance que mon système propose ;
- Définir une politique de décision et de confiance avantageuse. Ces politiques doivent trouver un compromis entre la sécurité et la confiance.

Nous avons choisi de présenter les principaux cas d'utilisations du modèle de cette manière pour rester précis. La définition de ce cahier des charges général permet de passer à la partie conceptuelle de l'étude.

1.3.3 Conception

La partie conceptuelle étudie les données mais aussi les traitements à effectuer. Ici, on utilise des techniques de modélisation. L'idée est de permettre aux lecteurs d'avoir un aperçu du fonctionnement du modèle de confiance. L'étude conceptuelle (spécification, définition de l'architecture...) n'est pas complète car ce n'est pas l'objectif de ce manuscrit. Dans un premier temps, expliquons l'*interaction entre les acteurs* du modèle après les avoir énoncés. Puis dans un second temps, nous proposons une représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique. Avant d'expliquer notre modèle en utilisant une méthode d'analyse particulière, nous aborderons le modèle de référence que nous avons utilisé. Ce modèle est décrit dans [7].

L'étude [7] est au cœur de notre modèle de réputation sur lequel s'appuie notre modèle de confiance. La figure 1.8 montre une vue générale de l'*interaction entre les composants* du modèle de confiance proposé. Il existe plusieurs analyseurs ou calculateurs de confiance qui sont reliés et partagent des informations. Les services communiquent avec les analyseurs de confiance. Les utilisateurs interagissent avec les fonctionnalités des services. Les fonctionnalités font remonter une information qui prend en compte leurs criticités à leurs services respectifs. Une fonctionnalité peut être partagée par plusieurs services. Un utilisateur peut utiliser des fonctionnalités de plusieurs services différents. L'entreprise a en effet validé l'existence d'un nombre important de fonctionnalités partagées par les différents services. En proposant cette architecture, nous gardons l'anonymat permis par les systèmes centralisés. Les

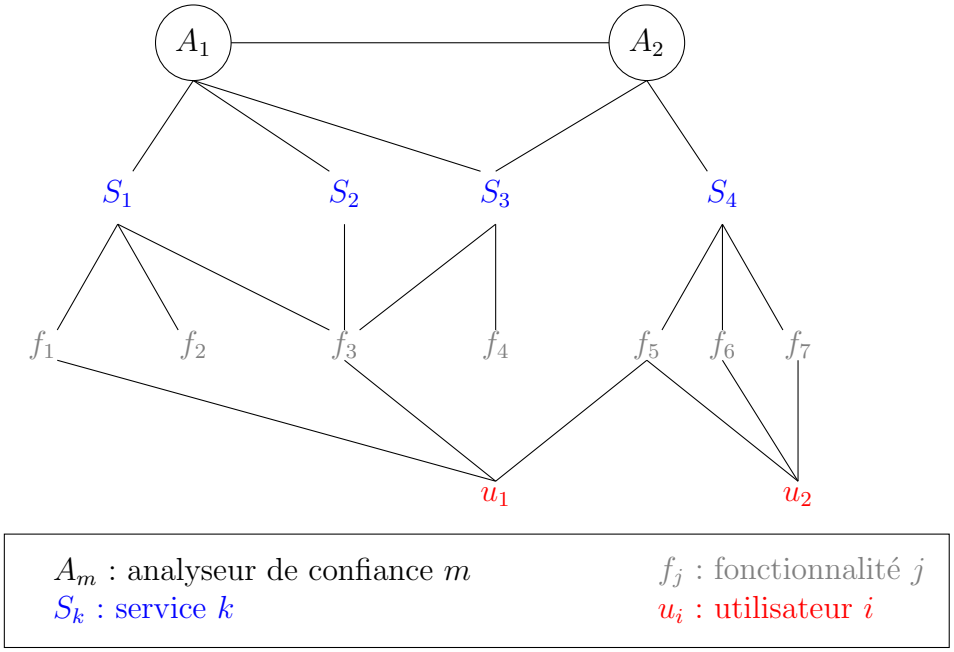


FIGURE 1.8 – L'interaction des acteurs du modèle de confiance

analyseurs de confiance savent tout et préservent la vie privée mais en attaquant un analyseur, le système continue de fonctionner. Il est possible d'ouvrir une discussion sur la redondance d'informations avec ce type d'architecture. Cependant en suggérant que chaque analyseur de confiance ne garde qu'un indice qui résume le comportement ou l'identité et que le calcul de ces indices soit irréversible, alors la question de la vie privée est résolue.

> **Exemple 4** Par exemple sur la figure 1.8, le service S_1 peut faire confiance à l'utilisateur u_1 pour utiliser la fonctionnalité réclamée f_1. Le service S_1 rendra compte de cette interaction à l'analyseur de confiance A_1.

Le *diagramme de séquence* de la figure 1.9 montre l'enchainement des actions lorsqu'il s'agit d'ouvrir l'accès à une fonctionnalité dans eBeeHome. Il décrit aussi l'interaction entre les acteurs et le système. Ce processus est valable sur la couche applicative. Lorsqu'un utilisateur clique sur une fonctionnalité, la fonctionnalité envoie à son service la requête et sa criticité. Ce service calcule la confiance instantanée de l'utilisateur qu'il envoie ensuite à un analyseur de confiance. Le service demande à l'analyseur de confiance ce qu'il sait de l'utilisateur émetteur de la requête, pour la

fonctionnalité réclamée. Cette analyseur calcule la confiance continue de ce service pour cette utilisateur et la stocke en mémoire. L'analyseur va regarder s'il a déjà en mémoire ou si d'autres analyseurs ont déjà en mémoire des confiances continues d'autres services pour cette utilisateur. Si c'est le cas, l'analyseur évalue une confiance globale pour l'utilisateur qui a envoyé la requête, à partir des confiances continues à disposition et de la matrice de *confidence* entre les services. Les composantes de cette matrice sont les confiances inter-services définies ci-dessus. La confiance globale de l'utilisateur est envoyée au service concerné par la requête. Cette confiance globale est classée dans son niveau de confiance équivalent. A l'aide de politiques de sécurité et d'experts, ce service possède un module capable de correspondre un niveau de confiance à un niveau de service. Si le niveau de service correspondant au niveau de confiance autorise l'accès à la fonctionnalité demandé alors la requête est acceptée. L'utilisateur accède à la fonctionnalité. Dans le cas contraire, la requête est rejetée. Un message d'erreur s'affiche.

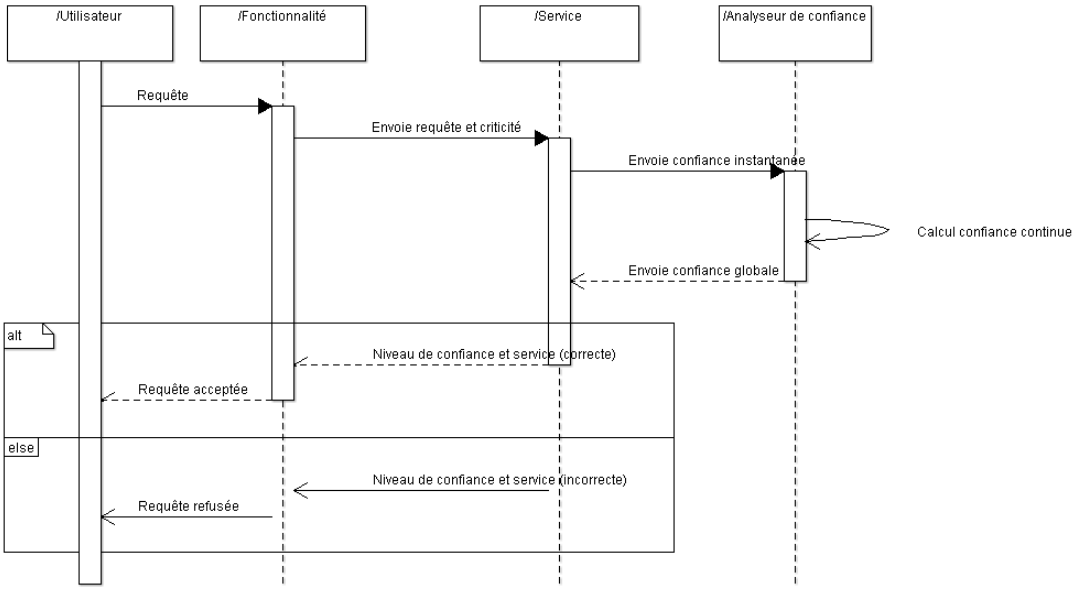


FIGURE 1.9 – Diagramme de séquence pour la confiance

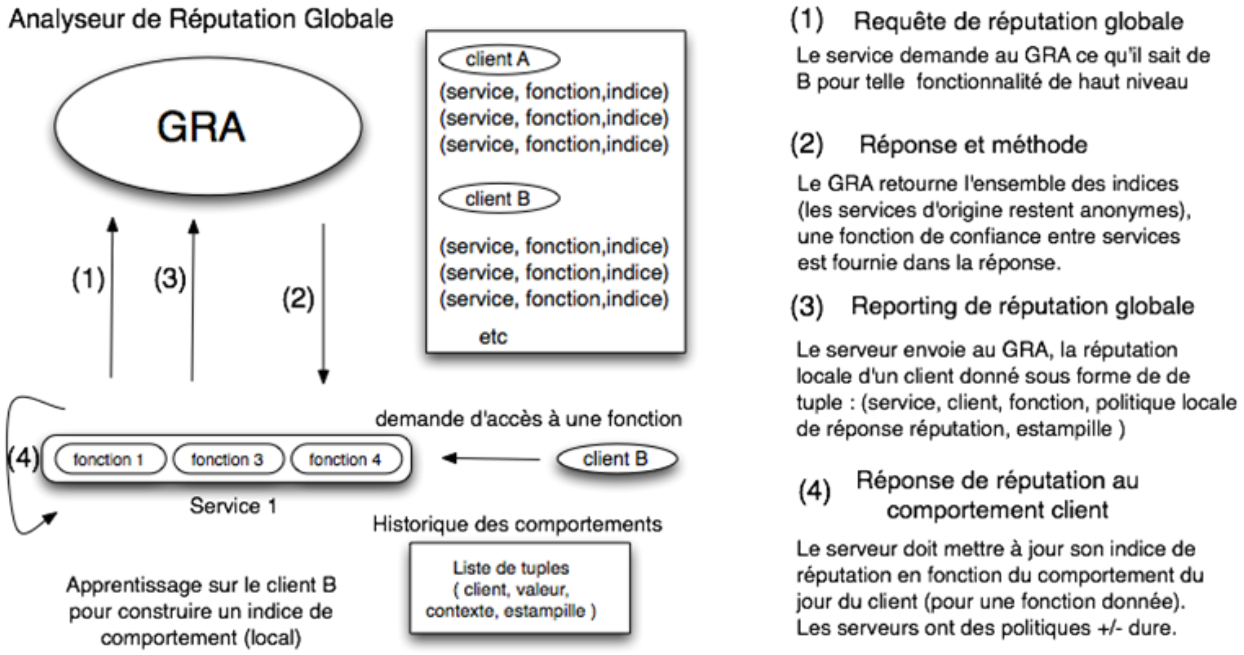


FIGURE 1.10 – Le fonctionnement du modèle de réputation d'Anirban Basu adapté à notre contexte

Une *adaptation du modèle de [7]* est proposée sur la figure 1.10. Dans ce modèle, le *Global Reputation Analyser* (GRA) mémorise les réputations envoyées par les serveurs et répond aux requêtes des serveurs lorsqu'il s'agit de demander la réputation d'un client. Le GRA est équivalent à notre analyseur de confiance. Le client est équivalent à un utilisateur dans notre modèle. Dans le modèle de [7], l'ensemble (service, fonction) est représenté par un serveur. Une *fonction* ou fonctionnalité de haut niveau de la figure 1.10 est une fonctionnalité du service dans eBeeHome. Lorsqu'un client B demande l'accès à une fonction appartenant à un service, le service demande au GRA ce qu'il sait de B pour telle fonctionnalité. Le GRA renvoie les indices des couples (service, fonction). Le serveur envoie au GRA la réputation locale du client B. Le système de réputation locale s'enrichit alors des comportements exclusivement sur cette fonctionnalité, et l'analyseur de confiance gère un système de recommandation basée lui aussi sur le couple (client, fonction).

Le *modèle que nous proposons* est structuré par module. Nous représentons ce modèle en utilisant la méthode d'analyse fonctionnelle descendante, *Structured Analysis and Design Technique* (SADT) (cf. [56]), introduite par Doug Ross en 1977. Cette méthode ne permet pas une représentation séquentielle du système mais elle a l'avantage de

permettre aux lecteurs d'avoir une structure claire hiérarchisée par niveau et une décomposition analytique de la complexité du système.

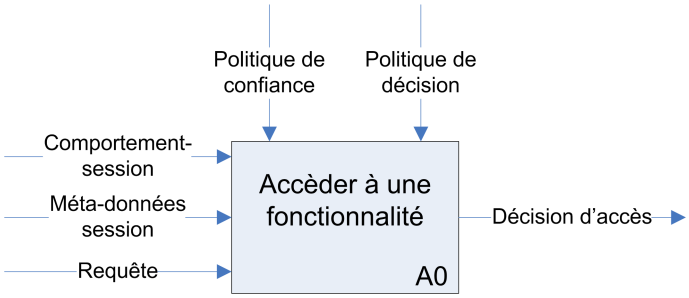


FIGURE 1.11 – Vue d'ensemble du modèle proposé

Dans le contexte de notre étude, les données d'entrée du système sont le comportement session (journal d'activités), les méta-données session (données personnelles...) et la requête d'accès à une fonctionnalité de l'utilisateur. Sur la figure 1.11, en sortie le système prend la décision d'accepter ou de refuser la requête de l'utilisateur en prenant en compte les paramètres du système, à savoir les politiques de confiance et de décision de l'entreprise.

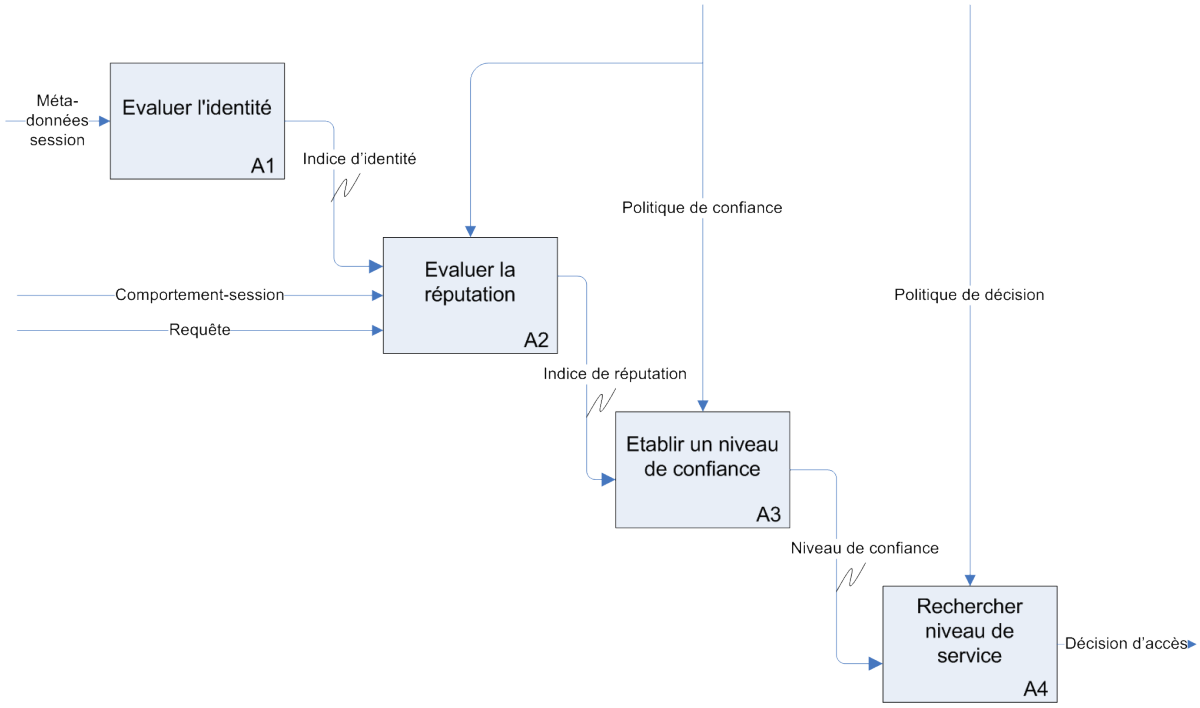


FIGURE 1.12 – Vue détaillée du modèle proposé

Nous proposons sur la figure 1.12, une vue détaillée de la fonction *Accéder à une fonctionnalité* de la figure 1.11. Pour accéder à une fonctionnalité, il est essentiel d'analyser quatre fonctions.

1. Construire un indice d'identité à l'utilisateur émetteur de la requête en utilisant les méta-données session (module A1).

2. Construire un indice de réputation à l'aide du journal d'activités, de la requête et de l'indice d'identité. Cette construction prend en compte les politiques de confiance de l'entreprise (module A2).

3. Évaluer un niveau de confiance à partir de l'indice de réputation et des politiques de confiance (module A3).

4. Définir le niveau de service associé au niveau de confiance afin de prendre la décision d'accepter ou de refuser une requête. Cette décision est basée sur les politiques de décision de l'entreprise (module A4).

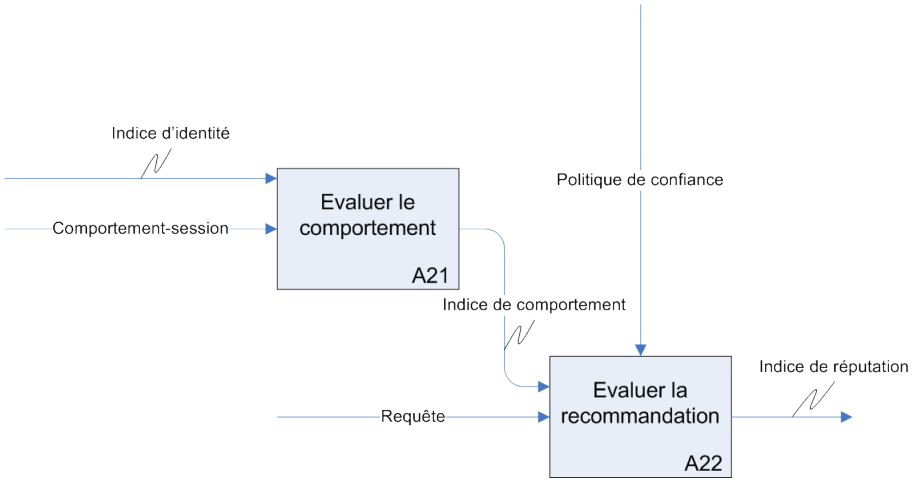


FIGURE 1.13 – Vue détaillée de l'évaluation de la réputation

La figure 1.13 montre plus précisément le fonctionnement de l'évaluation de la réputation. Sur cette figure, l'indice de réputation est issu de l'évaluation du comportement (étude du journal d'activités) ainsi que du système de recommandation entre les services décrit par la matrice de confiance inter-service. Nous avons vu les composants du modèle, l'interaction entre les différents composants ainsi que le fonctionnement général du modèle. Cela nous permet d'introduire l'évaluation des indices utilisés dans le modèle proposé.

Comme dans [7], nous proposons une *évaluation de l'indice* d'identité, de réputation et de comportement. Pour cela, il est nécessaire de définir le comportement session, les méta-données session ainsi que la requête sous forme de vecteur. Les méta-données session sont représentés par un tuple β :

$$\beta = \{id_user, nombre_session, lieu, date_event\}$$

Id_user est l'identifiant de l'utilisateur. Cet identifiant correspond à un profil enregistré dans le système lors de l'inscription de l'utilisateur sur eBeeHome. *Nombre_session* définit l'ancienneté de l'utilisateur dans le système, *lieu* est l'emplacement de l'utilisateur lorsqu'il se connecte au système. Enfin *date_event* représente la date et l'heure à laquelle la connexion a été faite. Calculé à chaque connexion, l'indice d'identité $Ind_{identite}$ est une fonction du tuple β définit par :

$$Ind_{identite} = \mathscr{F}_I(\beta)$$

Lors d'une requête, le service α reçoit un enregistrement τ et un ensemble d'enregistrement τ_i correspondant au comportement session de l'utilisateur (journal d'activités des fonctionnalités basiques) :

$$\tau_\alpha = \{objet_event, criticite, timestamp\}$$

Nous définissons *objet_event* comme la fonctionnalité demandée lors de la requête, *criticite* la criticité de la fonctionnalité et*timestamp* la date et l'heure à laquelle la requête a été faite.

$$\tau_{\alpha_i,i} = \{objet_event_i, timestamp_i\}$$

$Objet_event_i$ est la fonctionnalité basique utilisée lors de la navigation et $timestamp_i$ la date et l'heure à laquelle la fonctionnalité basique a été appelée.

L'indice de comportement $Ind_{comportement}$ est alors défini par la fonction $\mathscr{F}_C$. Cet indice dépend de trois indices énoncés dans la définition des terminologies utilisées : l'indice d'utilisation, de stress et d'annulation.

- L'indice d'utilisation prend un flux de comportement session τ_i et mesure la fréquence d'utilisation d'une fonctionnalité $objet_event_i$ pendant δ, un temps à paramétrer :
$$U^{\delta}_{objet_event_i} = \mathscr{F}_U(\tau_{\alpha_i,i})$$

- L'indice de stress prend un flux de comportement session τ_i et mesure la fréquence de refus à une demande d'accès à une fonctionnalité $objet_event_i$ pendant δ, un temps à paramétrer :

$$S^{\delta}_{objet_event_i} = \mathscr{F}_S(\tau_{\alpha_i,i})$$

- L'indice d'annulation prend un flux de comportement session τ_i et mesure la fréquence d'annulation d'une fonctionnalité $objet_event_i$ pendant δ, un temps à paramétrer :

$$A^{\delta}_{objet_event_i} = \mathscr{F}_A(\tau_{\alpha_i,i})$$

L'indice de comportement est fonction de l'ensemble des indices décrits ci-dessus et de l'indice d'identité :

$$Ind_{comportement} = \mathscr{F}_C(U^{\delta}_{objet_event_i}, S^{\delta}_{objet_event_i}, A^{\delta}_{objet_event_i}, Ind_{identite})$$

L'indice de réputation $Ind_{reputation}$ est fonction de l'indice de comportement, de la requête, de la politique de confiance ainsi que de la matrice de recommandation :

$$Ind_{reputation} = \mathscr{F}_R(Ind_{comportement}, \tau_\alpha, \mathscr{P}_c) \cup \mathscr{M}$$

$\mathscr{P}_c$ représente les politiques de confiance. $\mathscr{M}$ est la matrice de confiance entre les services.

Nous définissons le niveau de confiance par :

$$Confiance = \mathscr{F}_{conf}(Ind_{reputation}, \mathscr{P}_c)$$

Enfin, la décision prise par le système est un booléen représenté par σ, avec $\mathscr{P}_d$ la politique de confiance :

$$\sigma = \mathscr{F}(Confiance, \mathscr{P}_d)$$

1.3.4 Limites et extensions

Ce modèle, proposé suite à une étude de l'état de l'art, présente des limites conceptuelles que nous allons détailler ci-dessous.

La limite principale de ce modèle est l'évaluation des indices. Les indices sont définis. L'évaluation de la réputation est étudiée dans [7] ainsi que dans [16]. Il existe des évaluations possibles dans la littérature et dans le travail de référence [7] en particulier, mais l'étude reste générale sur ce point. Dans [7], les difficultés à proposer des fonctions de construction d'indice à partir du vecteur d'historique des comportements observés sont évoquées. Dans la suite de ce manuscrit, nous proposons de construire un profil utilisateur en utilisant des techniques de fouille de données. Il serait possible d'évaluer l'indice de comportement à partir de ce profil utilisateur.

Il est intéressant de discuter du temps de réponse d'un système basé sur ce modèle. L'utilisateur qui demande l'accès à une fonctionnalité doit avoir une réponse instantanée. De plus une fonctionnalité visible sur l'interface mais qui n'est pas accessible par l'utilisateur parce que son niveau de confiance ne le lui permet pas, pousse à la réflexion. Cette fonctionnalité pourrait être invisible et devenir visible dès que le niveau de confiance de l'utilisateur devient suffisant.

Dans notre modèle, nous avons choisi de travailler à l'échelle de la fonctionnalité. C'est une des originalités de ce modèle. Par contre, en choisissant ce niveau de granularité, une perte d'information est possible. En effet les logs, d'un niveau de granularité plus bas, sont consolidés dans la brique de traçabilité. Il est intéressant de savoir si le niveau de granularité a une influence sur la qualité et la robustesse du modèle.

La représentation des niveaux de confiance et de services n'a pas été abordée dans cette étude. Même si intuitivement on pourrait utiliser une représentation qualitative, il est nécessaire de travailler avec des experts du métier afin de définir le nombre adéquat de catégories pour les niveaux de confiance et de service.

L'enchainement des actions lorsque l'analyseur de réputation communique avec le service est une limite potentielle du modèle. En effet, il n'y a pas eu de réflexions sur d'autres schémas possibles. Nous avons adapté le travail [7] à notre contexte. L'enchainement des actions pour le modèle de réputation d'Anirban Basu est différent de celui du modèle qui a été proposé. En effet, notre proposition semblait plus adéquate à notre contexte. Cependant, l'étude n'a pas été poussée sur les avantages et les inconvénients des deux types de processus.

Le modèle reprend les fondements des modèles de confiance mais s'adapte à notre contexte. Cependant il reste assez général et perfectible. Étant en phase de conception, il est impossible de mesurer finalement la production de confiance obtenue grâce à ce modèle. Par contre la conceptualisation permet une implémentation afin de pouvoir mettre en place des mesures de confiance dans l'environnement eBeeHome. Ce sont des mesures de qualité soumises aux fournisseurs de service et aux utilisateurs des services. Enfin ces mesures de qualité peuvent être implicite (par exemple étude du temps de connexion sur eBeeHome) ou explicite (par exemple questionnaire de satisfaction en ligne).

Pour des raisons internes en entreprise à cette époque, le modèle conceptuel n'a été ni approfondi ni implémenté. Nous avons décidé de proposer un élément de confiance complémentaire au modèle de confiance : l'authentification implicite.

1.4 Authentification implicite

Les entreprises incitent leurs clients à accéder à leurs services via Internet. On admet que les services en ligne sont plus immédiats et plus confortables que l'accès à ces services via une agence physique qui implique un déplacement et souvent de l'attente. Néanmoins, l'accès à ces services pose des problèmes de sécurité. Certains services donnent accès à des données sensibles telles que les données bancaires pour lesquelles il est primordial d'authentifier les utilisateurs. Pourtant le nombre d'usurpations d'identité est en croissance permanente (cf. [57]). Nous pouvons distinguer deux paradigmes pour augmenter la sécurité des accès. Le *premier* consiste à durcir les protocoles d'accès en s'appuyant par exemple sur des *devices* externes pour transmettre des codes d'accès complémentaires au couple identifiant et mot de passe. Néanmoins ces processus nuisent au confort des services et à leur utilisabilité. Le nombre de transactions abandonnées avant la fin du processus augmente et les volumes

d'échanges diminuent. Le *second* paradigme consiste au contraire à simplifier les processus d'identification pour augmenter les volumes d'échanges. Nous pouvons citer à titre d'exemple le paiement en un clic (cf. [58] et [59]) ou l'utilisation des puces RFID pour les paiements sans contact. A l'intersection de ces deux paradigmes on trouve les moyens d'authentification implicite. En effet, un *moyen d'authentification implicite* a pour objet de renforcer la sécurité des services sans nuire à leurs utilisabilités.

Pour produire de la confiance dans un environnement multi-services, nous proposons de mettre en place une authentification implicite. Elle se présente comme une technologie permettant d'authentifier un utilisateur du système sans le solliciter. L'espace de vie numérique eBeeHome n'est accessible qu'avec une authentification forte (couple carte à puce et code PIN), ce qui est très contraignant pour les utilisateurs. En effet, cet espace de services comprend des fonctionnalités critiques qui ne doivent être accessibles que par un utilisateur authentifié et autorisé à y accéder. Pour faciliter l'accès à cet ensemble de services, une authentification faible (couple identifiant et mot de passe) sera disponible prochainement. L'authentification faible est moins contraignante mais elle est moins sécurisée et expose les fonctionnalités critiques à une utilisation malveillante. Le but de ce travail est de rendre cette authentification faible « plus forte » en effectuant une authentification implicite supplémentaire. En cas de perte ou de vol du moyen d'authentification, l'objectif de l'étude est de faire en sorte que le système puisse détecter la présence de l'intrus et suspendre l'accès à cet espace, en se basant sur son comportement. L'objectif de la mise en place de l'authentification implicite est aussi de rationaliser les prises de risque des fournisseurs de services en ligne et de produire de la confiance numérique.

Nous définissons d'abord l'authentification en passant par l'identification dans 1.4.1. Nous introduisons l'authentification implicite dans 1.4.2 et proposons un état de l'art sur ce sujet. Enfin, nous présentons comment le module d'authentification implicite est intégré dans l'espace de vie numérique eBeeHome.

1.4.1 Identification et authentification

> **Exemple 5** Sur la figure 1.14, un utilisateur anonyme est retrouvé parmi plusieurs utilisateurs d'une base de données. On parle d'identification. Dans le cas où le système vérifie que l'identité déclarée par Bob est bien celui de Bob, on parle d'authentification.

Beaucoup ont compris l'intérêt de mettre en évidence la différence entre l'identification et l'authentification. L'identification est l'action d'assimiler une personne ou une chose à une autre (cf.[2] et [60]). L'authentification est l'action de vérifier et de certifier l'identification faite au préalable (cf.[61] et [62]). Dans [63], les auteurs

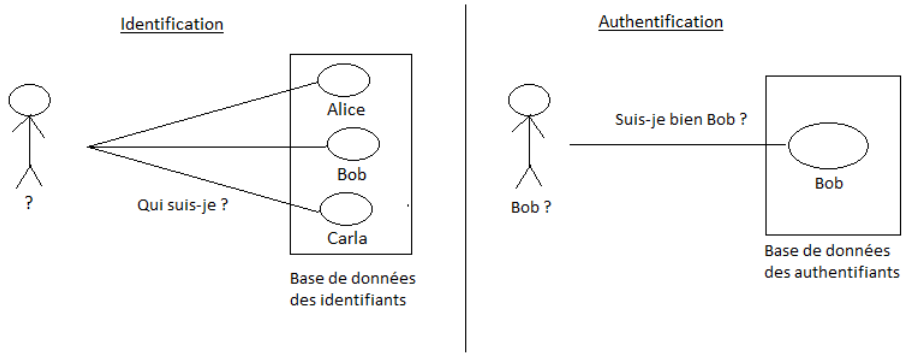


FIGURE 1.14 – Identification vs Authentification

expliquent que l'identification est une comparaison 1 à n (avec n le nombre d'éléments à comparer) et l'authentification est une comparaison 1 à 1 comme le montre la figure 1.14. L'article [64] donne une définition de l'identification et de l'authentification en comparant les résultats issus du procédé d'identification et d'authentification. En effet, l'authentification permet de donner une réponse fermée (oui ou non) et l'identification permet une réponse ouverte.

Un profil de l'utilisateur est construit à partir du comportement passé (cf. [7], [65], [66]) et un calcul de score est effectué dans les processus d'identification et d'authentification.

Exemple 6 Le processus d'identification est illustré sur la figure 1.15. Le profil d'Alice et de Bob est calculé par un algorithme d'apprentissage. Ce dernier est comparé à un profil anonyme « en cours » par un calcul de score. Dans cet exemple, le comportement en cours de l'utilisateur anonyme est plus proche de celui de Bob. L'utilisateur anonyme est identifié comme étant Bob à 80 %.

Exemple 7 Le processus d'authentification est illustré sur la figure 1.16. Alice est enregistrée dans le système et son profil utilisateur est connu du système. Une personne affirmant être Alice (couple identifiant et mot de passe appartenant à Alice) se connecte au système. Durant son utilisation du système, elle a un comportement en cours. Un score est calculé en comparant son comportement en cours au profil d'Alice enregistré dans la base de données du système. Alice est celle qu'elle prétend être à 30%. En fonction du seuil d'authentification et de la politique du système, Alice est authentifiée ou pas.

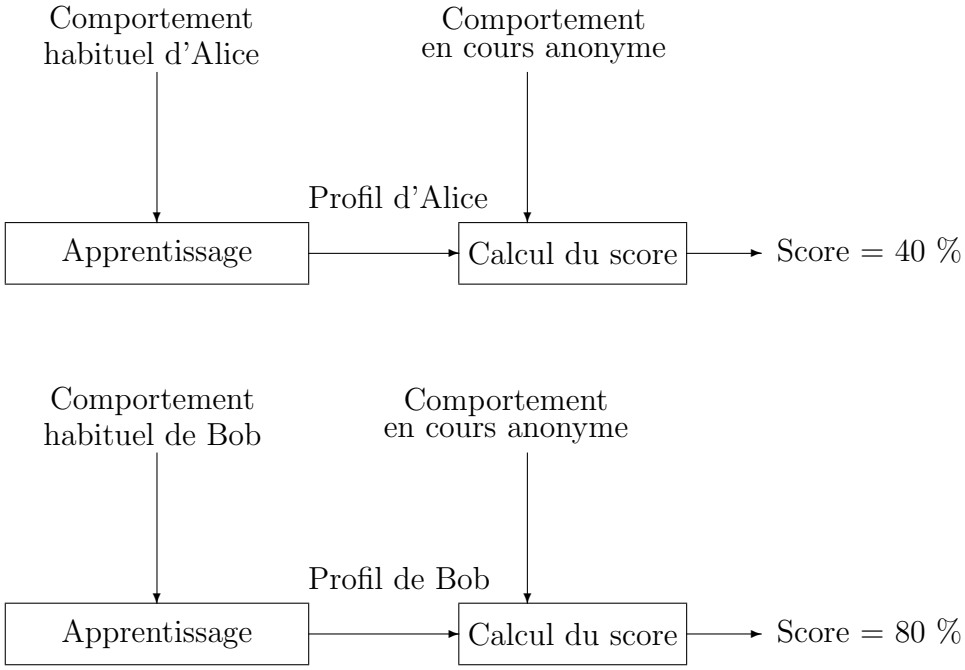


FIGURE 1.15 – Le processus d'identification

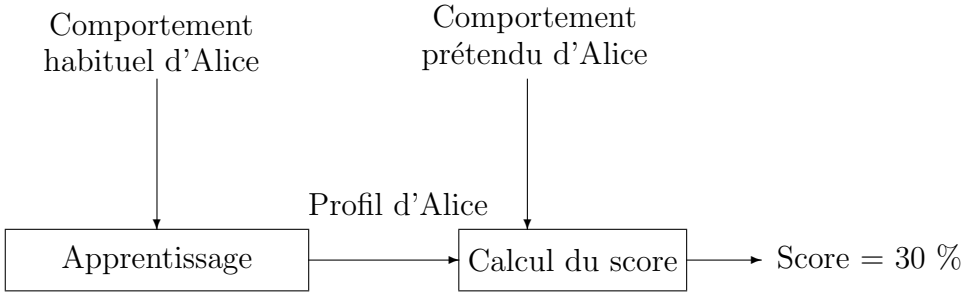


FIGURE 1.16 – Le processus d'authentification

Soit W_p^{Alice} le vecteur de profil d'Alice, W_c^{Alice} le vecteur de profil courant prétendu d'Alice, une fonction de distance ou de similarité *dist* va permettre de calculer le score. La fonction f permet d'authentifier. f est une fonction booléenne :

$$dist(W_p^{Alice}, W_c^{Alice}) = score \tag{1.1}$$

$$f(score, seuil) = \begin{cases} 1 & \text{si } score > seuil \\ 0 & \text{sinon.} \end{cases}$$

La variable *seuil* de la fonction d'authentification est à déterminer. Elle peut être fixée par le système ou par l'utilisateur.

> **Exemple 8** A partir de l'exemple de la figure 1.16, si dans eBeeHome le seuil est fixé à 40%, alors Alice ne sera pas authentifiée.

1.4.2 Définitions et concepts

Dans cette sous-section, une définition précise de l'authentification implicite s'appuyant sur l'état de l'art est donnée. La littérature est riche de travaux qui s'intéressent à l'authentification implicite à base d'études comportementales. Nous présentons quelques uns de ces travaux. Enfin, les questions de recherche liées à l'authentification implicite sont abordées.

1.4.2.1 Implicite vs explicite

Un moyen d'authentification est un processus permettant de s'assurer que l'identité déclarée lors de l'accès est bien celle de l'utilisateur (cf. [61]). Traditionnellement un utilisateur s'authentifie en donnant une preuve d'identité (cf. [67]). Ce processus est appelé authentification *explicite*. Par opposition, l'authentification *implicite* ne demande rien à l'utilisateur mais étudie son comportement, la trace de ces actions et valide ou non l'identité déclarée (cf. Tableau 1.2). Elle est transparente pour l'utilisateur.

> **Exemple 9** Par exemple sur le tableau 1.2, un mot de passe est entré par un utilisateur via le clavier de son ordinateur. Le mot de passe est un moyen d'authentification dit « explicite » car l'utilisateur est sollicité. Par contre, ce moyen d'authentification est menacé par les logiciels espions qui enregistrent les touches utilisées par un utilisateur sur son clavier.

Dans plusieurs articles (cf.[68], [69], [66], [70], [71], [72], [73] et [74]), l'authentification implicite est avant tout un moyen d'améliorer la sécurité du système mis en place. L'authentification implicite est définie plus précisément dans [68] par l'utilisation de données comportementales et biométriques de l'utilisateur par le système de manière

transparente dans le but de vérifier le propriétaire d'un appareil mobile. Le tableau 1.2 est repris de l'article [68] et montre les éléments biométriques et comportementaux utilisés pour faire de l'authentification implicite.

1.4.2.2 Authentification implicite basée sur le comportement des utilisateurs

Pour que le fournisseur de service puisse garantir la sécurité lorsque l'utilisateur accède à un service en ligne, l'utilisateur doit fournir au moins un des trois types d'éléments suivant : un élément qu'il connait (par exemple un mot de passe), un élément qu'il possède (par exemple une carte à puce) ou ce qu'il est (par exemple l'empreinte digitale). L'authentification faible (au moins un des éléments cités précédemment doit être fourni) et l'authentification forte (combinaison d'au moins deux éléments) ont déjà fait leur preuve et sont largement utilisées de nos jours. Nous rappelons que l'authentification implicite est un moyen d'authentification complémentaire aux moyens d'authentification classiques. L'authentification implicite, basée sur le comportement habituel de l'utilisateur, a montré son efficacité dans le domaine de la sécurité notamment sur les appareils mobiles et pourra être utilisée lors de l'accès à tout objet connecté. L'authentification implicite utilise ce que l'utilisateur est. On suppose que le comportement habituel de l'utilisateur est une caractéristique biométrique. Le comportement habituel satisfait les exigences d'un système biométrique classique. En effet, un système biométrique permet deux fonctions : la vérification et l'authentification. Il est possible de vérifier un comportement grâce à son caractère habituel. On émet l'hypothèse suivante : la routine est présente chez la plupart des utilisateurs et est unique. La routine est alors un critère qui permet de reconnaitre un utilisateur et de l'authentifier. Le caractère unique de la routine étant discutable, l'authentification à base de comportement reste un complément à d'autres mécanismes biométriques.

Dans l'étude [75], les auteurs introduisent les systèmes de reconnaissance biométrique. Ces systèmes sont basés sur une liste des caractéristiques biométriques physiologiques telles que l'acide désoxyribonucléique (ADN) et la voix, ou comportementales (cf. [76]). Toute caractéristique biométrique doit satisfaire un ensemble de propriétés selon [63]. Le système biométrique dans [75] utilise les données biométriques d'un individu et compare ces données à un *template* (profil de l'utilisateur) enregistré auparavant dans la base de données du système. Ce système a deux modes de fonctionnement qui dépendent du contexte de l'application : le mode d'identification et le mode d'authentification.

Les systèmes d'authentification implicite ont très vite été étudiés pour les téléphones mobiles. Dans [69] et [66] les auteurs étudient le comportement des utilisateurs en

se basant sur des variables propres aux smartphones tels que les appels, les sms, la navigation entre les applications, la localisation et l'heure de la journée. Les expérimentations ont été menées à partir des données de 50 utilisateurs sur une période de 12 jours. Les données ont été collectées grâce à une application installée par des utilisateurs volontaires. Les profils des utilisateurs sont construits à partir des fréquences d'événements positifs ou négatifs et de la localisation. Dans ce contexte, un événement positif est un événement cohérent avec des informations collectées en amont. A titre d'exemple, appeler un numéro qui se trouve dans le répertoire du téléphone est un événement positif. Les résultats de cette étude montrent qu'à partir d'une dizaine d'actions on peut détecter une utilisation frauduleuse d'un smartphone avec une précision de 95%.

L'article de [71] propose une architecture efficace qui respecte aussi la vie privée des utilisateurs dans un système d'authentification implicite. Dans [73], le système d'authentification implicite proposé dans le contexte des appareils mobiles est décentralisé. Chaque application authentifie l'utilisateur implicitement. L'étude de [73] permet d'avoir des résultats plus précis et améliore le système d'authentification implicite. Les systèmes proposés dans la littérature sont centralisés (cf. [69]). En effet, l'appareil mobile authentifie l'utilisateur implicitement.

1.4.2.3 Questions de recherche

L'objectif de cette étude est de mettre en place un système d'authentification implicite comme le montre la figure 1.17 afin de produire de la confiance numérique. Pour cela, il est nécessaire de :

- Construire un *profil utilisateur* à partir de son comportement passé par apprentissage. Dans notre étude, nous regardons à partir de quel moment le profil construit est fiable. Par contre, la mise à jour du profil ainsi que sa suppression ne sont pas étudiées ;
- Calculer un *score* en utilisant une fonction de comparaison (mesure de distance ou similarité) optimale et adaptée ;
- Authentifier un utilisateur en fonction de son profil. Le paramètre « seuil » doit être réglé en fonction d'un niveau de risque accepté par l'entreprise. Cependant les niveaux de risque ne sont pas traités dans ces travaux.

Exemple 10 La figure 1.17 illustre le modèle que nous proposons. Sur ce modèle, l'authentification implicite se base sur de l'identification. En

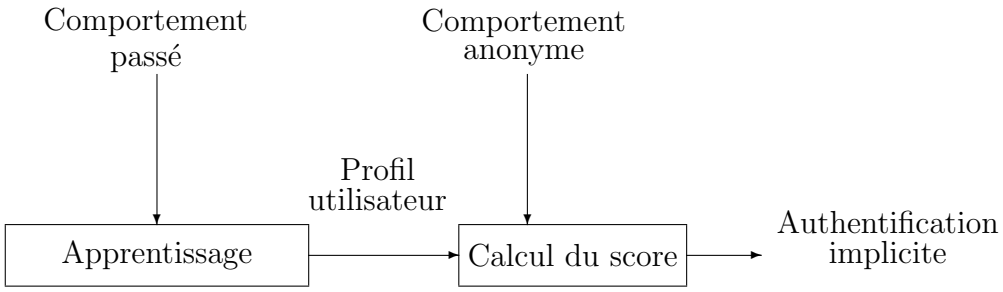


FIGURE 1.17 – Architecture de l'authentification implicite à base d'identification

effet, l'idée est de vérifier l'identité de l'utilisateur ayant un comportement anonyme pour ensuite l'authentifier.

1.4.3 Authentification implicite dans eBee

Expliquons le fonctionnement du module d'authentification implicite dans l'espace de vie numérique[8]. Une vue globale du fonctionnement du module d'authentification est présentée sur la figure 1.18. Les figures 1.18, 1.19 et 1.20 montrent en détails le fonctionnement du système.

Exemple 11 La figure 1.18 montre le fonctionnement général du module d'authentification. Lorsque l'utilisateur demande à accéder à une fonctionnalité critique, le système construit un profil de l'utilisateur, ensuite il regarde si le profil mémorisé est fiable. Si c'est le cas, l'utilisateur est authentifié implicitement et il accède à la fonctionnalité demandée. Si ce n'est pas le cas, selon la politique de refus de l'entreprise, un message d'alerte est envoyé ou bien le système arrête de fonctionner.

Exemple 12 Sur la figure 1.19, la phase d'apprentissage du profil est détaillée. Les données permettant de décrire un profil sont nettoyées. Elles sont ensuite analysées et filtrées. Les meilleurs descripteurs définissant un profil sont sélectionnés après une fouille de motifs fréquents.

Exemple 13 Sur la figure 1.20, le profil construit est comparé au comportement courant en utilisant un calcul de score. Les politiques de

8. www.ebeeoffice.ca/ebee-home/public, accès 09/07/2015

décision (choix du seuil) permettent ensuite d'authentifier implicitement l'utilisateur ou pas.

1.5 Résumé et discussion

Ce chapitre présente les premiers travaux de la thèse et introduit le deuxième sujet : l'authentification implicite.

Une définition de la confiance numérique est proposée à travers une analyse des précédents travaux (cf. [4] et [16]). Retenons, ici, les mécanismes de la confiance en informatique. Cette étude permet de mettre en évidence les points que nous estimons essentiels, dans le vaste champ de la confiance numérique :

- les politiques de confiance ;
- les modèles de confiance à base de réputation et de recommandation ;
- les modèles généraux de confiance dont les concepts sont empruntés à la sociologie et la psychologie.

Le modèle général de confiance numérique au cœur de l'application regroupant un ensemble de services en ligne, eBeeHome, est détaillé et constitue notre principale contribution. Ce modèle s'appuie sur le système de réputation décrit dans [7]. Le modèle n'est pas implémenté mais l'étude conceptuelle est établie.

Les concepts de l'authentification implicite, présentée comme un élément producteur de confiance numérique, sont également livrés dans ce chapitre. L'esquisse d'un modèle d'authentification implicite basé sur l'identification est décrite après avoir analysé l'état de l'art sur ce sujet. Trois étapes sont retenues lorsqu'il s'agit d'authentifier un utilisateur avec notre modèle : la construction du profil, la comparaison avec un comportement anonyme via un calcul de score et enfin l'authentification en fonction d'un seuil. Enfin le fonctionnement du module d'authentification implicite dans eBeeHome est décrit dans ce chapitre. La contribution se trouve dans la manière de traiter les questions qui surviennent à chaque étape de l'authentification implicite.

Caractéristique	*Méthode de capture*	*Implicite / Explicite*	*Menaces d'usurpation*	*Problèmes*
Code	Saisie clavier	Explicite	Enregistreurs de frappe, regarder au-dessus de l'épaule pour voler le code	Des mots de passe devinables sont le plus souvent utilisés.
Jeton d'authentification	Dispositif matériel	Principalement explicite, implicite possible	Aucun	Facilement volé ou perdu
Visage & Iris	Caméra	Les deux	Des photos de l'utilisateur légitime	Situation d'éclairage et maquillage
Parole	Microphone	Les deux	Enregistrement de la voix de l'utilisateur	Maladie, changement de voix, bruits de fond
Frappe sur clavier	Clavier	Implicite, explicite possible	Imiter la frappe (difficile)	Phase d'apprentissage long, fiabilité
Location	GPS, infrastructure	Implicite	Étrangers informés	Voyage, précision
Réseau	Protocole logiciel (e.g. WireShark)	Implicite	Étrangers informés	Précision

TABLE 1.2 – Comparaison de différents méthodes d'authentification

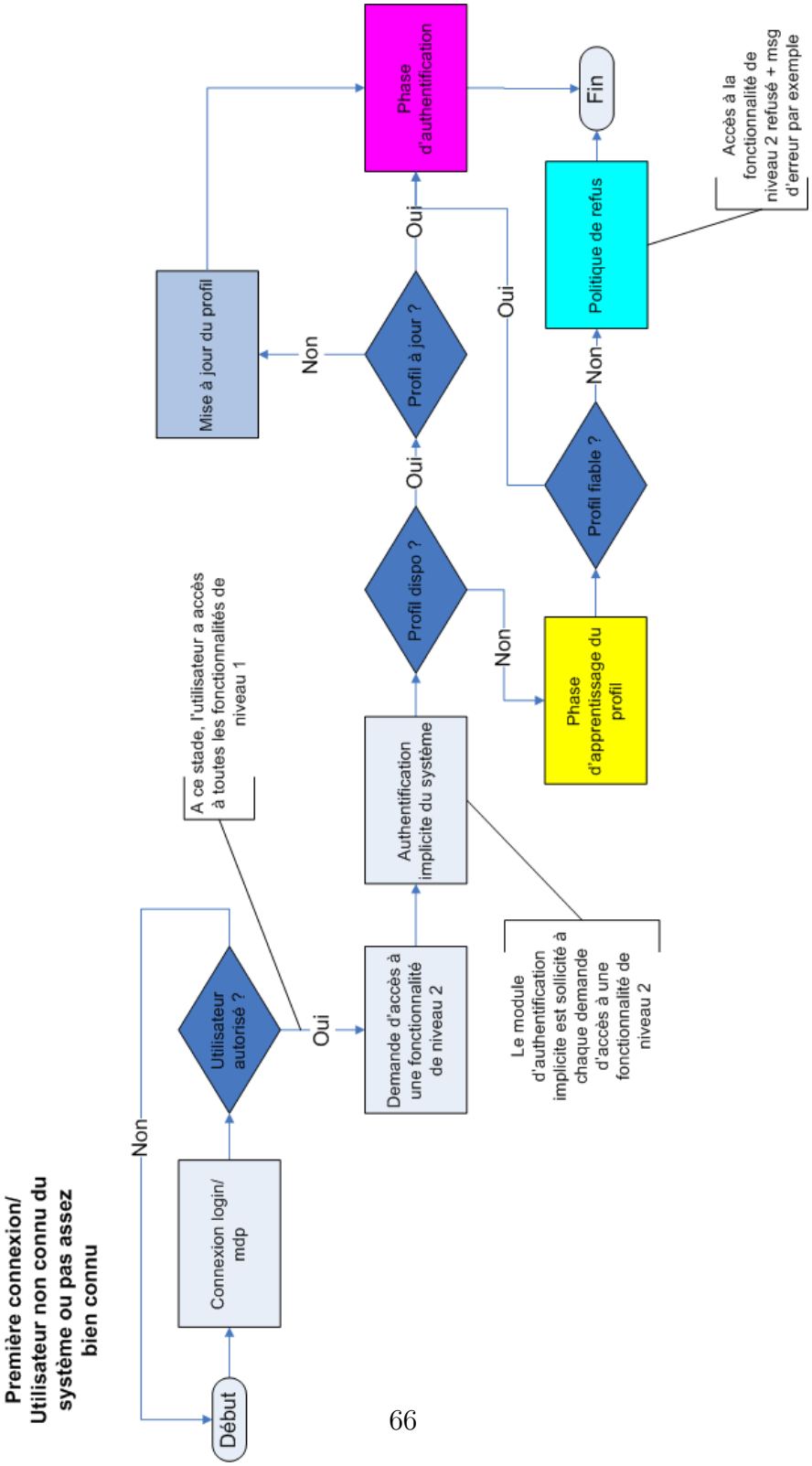


FIGURE 1.18 – Fonctionnement de l'authentification implicite dans eBeeHome : vue globale

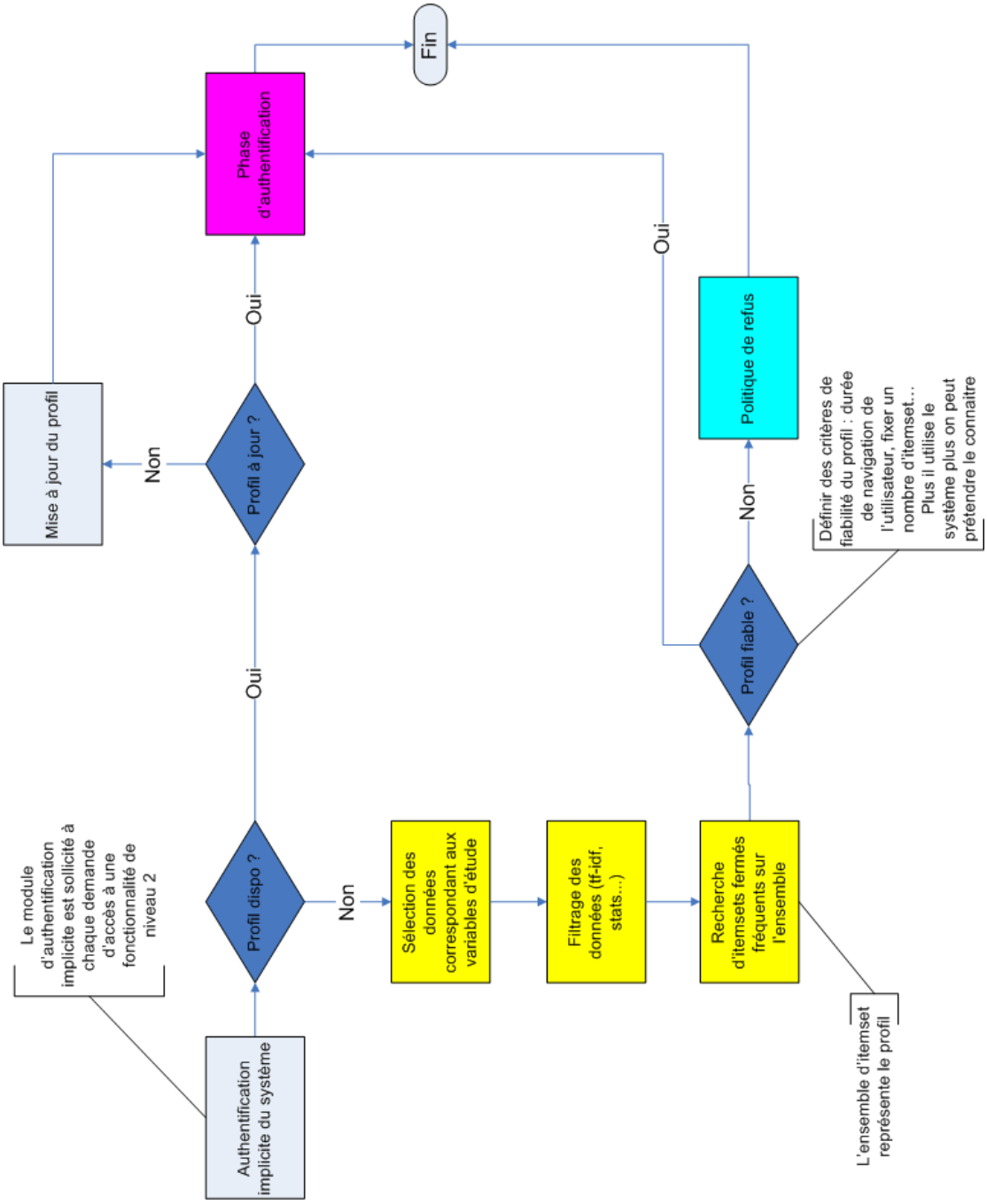


FIGURE 1.19 – Fonctionnement de l'authentification implicite dans eBeeHome : apprentissage du profil

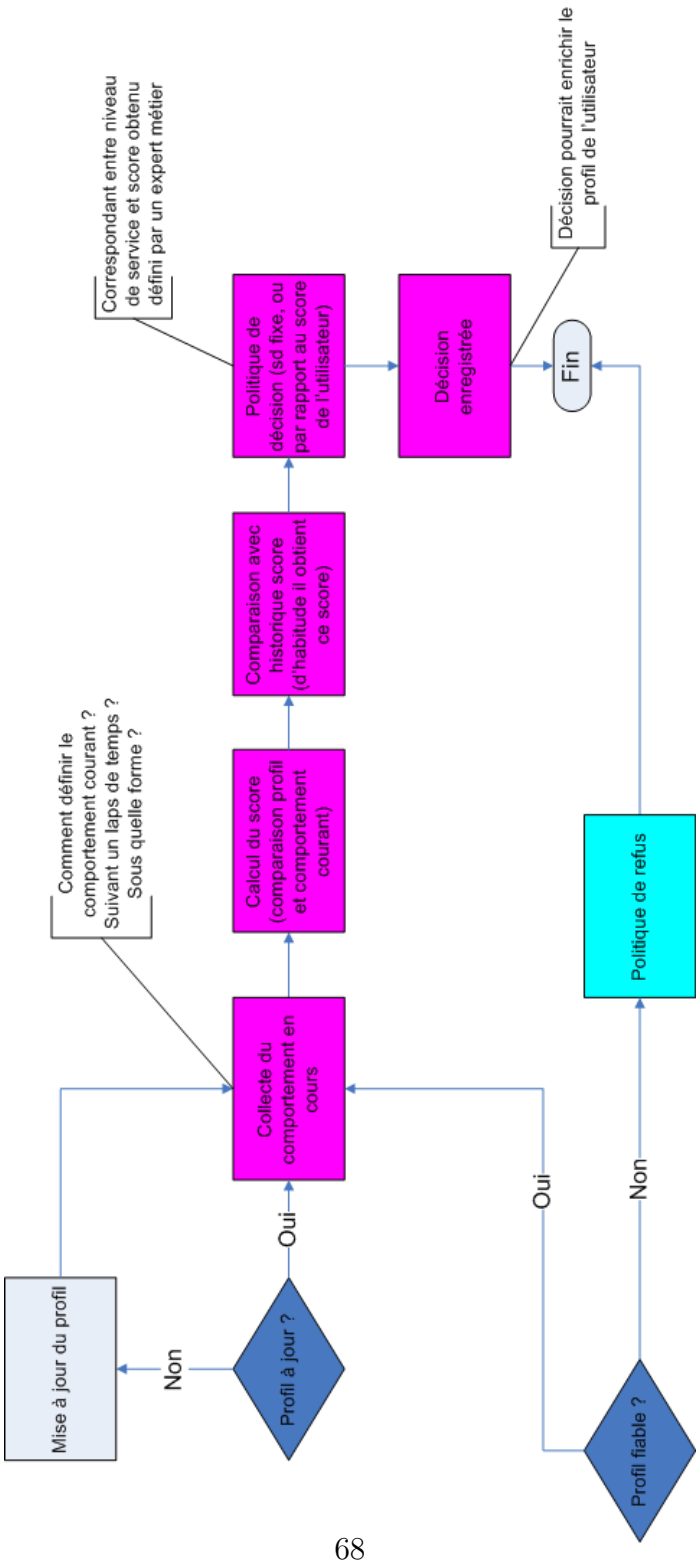


FIGURE 1.20 – Fonctionnement de l'authentification implicite dans eBeeHome : étape d'authentification

Chapitre 2

Itemsets fermés fréquents discriminants

La vie crée des motifs aléatoires. Ils sont chaotiques et emprunts d'une beauté que j'essaie de capturer lorsqu'ils passent, car qui sait s'ils reviendront un jour.

Margot Fonteyn, (1919-1991).

2.1 Introduction

Pour mettre en place l'authentification implicite, nous proposons un système qui s'appuie sur la classification à base de recherche de motifs fréquents. Dans ce chapitre, nous allons définir les notions et outils utilisés à travers la thèse pour répondre aux problématiques posées dans le chapitre précédent. Ces formalismes permettront une meilleure compréhension par la suite.

Les Hommes ont longtemps appris en observant les phénomènes répétitifs de leur environnement. Par exemple, les cycles lunaires sont à la base du calendrier musulman appelé calendrier hégirien. Selon le dictionnaire Larousse, le motif est « un dessin ou un ornement le plus souvent répété, sur un support quelconque ». Cette définition s'applique au domaine de l'art. Le motif est aussi défini dans le domaine de la musique comme un « ensemble de valeurs rythmiques pouvant se répéter indépendamment du contexte mélodique et harmonique », d'après le TLF [1]. Ces dernières années, les quantités de données collectées, dans divers domaines d'application, deviennent de plus en plus importantes. Ces quantités ont suscité le besoin d'analyse et d'interprétation afin d'en extraire des connaissances utiles. Dans ce contexte, la fouille de données propose des outils et des techniques nécessaires pour l'extraction de ces connaissances. La découverte de motifs est un des axes de recherche les plus importants en fouille de données et repose sur la mise en évidence d'informations récurrentes dans des données. En effet, un motif est une information, le plus souvent répétée et présentant un intérêt particulier, repérée dans une base de données (cf. [77]). Une base de données est une collection de données organisées. Le système de gestion de bases de données (SGBD) permet de structurer ces données, de les manipuler sous son contrôle et d'en garantir l'intégrité. Les motifs ont été introduits par [78] avec la définition des règles d'association. Les règles d'association sont des outils d'aide à la décision utilisés souvent en marketing. Lors de l'analyse du panier du client d'un supermarché, l'extraction des règles d'association permet de savoir comment organiser les rayons

1. Le Trésor de la Langue Française est un dictionnaire de la langue française.

dans les supermarchés. Par exemple, la règle d'association suivante : « pain et beurre sont des articles achetés ensemble dans 80% des cas », permet de mettre le pain et le beurre côte à côte dans les rayons de supermarché. Introduit par Rakesh Agrawal et Ramakrishnan Sikrant en 1994 (cf. [78]), l'algorithme Apriori, qui est à la base de nombreux algorithmes complexes d'extractions de règles d'association, est l'algorithme de référence lorsqu'il s'agit de générer des motifs de type *itemsets* fréquents. Un *itemset* fréquent est un ensemble non ordonné d'objets ou d'attributs, qui se répète dans la base de données. Cependant cet algorithme présente l'inconvénient de manière générale de générer un grand nombre d'*itemsets* fréquents. Il a été nécessaire de trouver des objets qui résument l'information d'une base de données de manière plus précise (cf. [79]) afin de remédier à l'inconvénient de l'algorithme Apriori. Des algorithmes, par exemple Charm (cf. [80]), qui permettent d'extraire ces types d'objets ont aussi été proposés dans la littérature.

Les motifs discriminants permettent de distinguer une classe ou une population d'une autre. L'extraction de motifs de ce type permet d'avoir une information non redondante. Les méthodes de sélection des motifs discriminants sont proposés dans ce chapitre également. Elles permettent de filtrer les ensembles d'objets obtenus par des algorithmes comme Charm afin de sélectionner les « meilleurs ». Les motifs discriminants doivent être assez fréquents pour représenter une population et assez discriminants pour distinguer une population d'une autre. Dans [77], l'auteur souligne que la notion d'intérêt ou de discrimination est propre à chaque problématique. Concernant la problématique expliquée dans [77], le motif discriminant est basé sur trois critères :

- la fréquence ;
- la discriminance ;
- la non redondance.

La figure 2.1 récapitule l'ensemble des types de motif que nous allons aborder dans ce chapitre. Dans 2.2, nous présentons les définitions générales sur les motifs. Dans la sections 2.3, nous nous intéressons à un type de motifs particulier : les *itemsets* fermés fréquents qui sont des ensembles d'objets non ordonnés fréquents à partir desquels il est possible de générer toute une base de données. Nous verrons dans 2.4 la définition des trois critères cités plus haut et leurs utilisations dans le choix des méthodes d'extraction des motifs discriminants. Enfin, une discussion est présentée sur les différents motifs définis, dans la section 2.5.

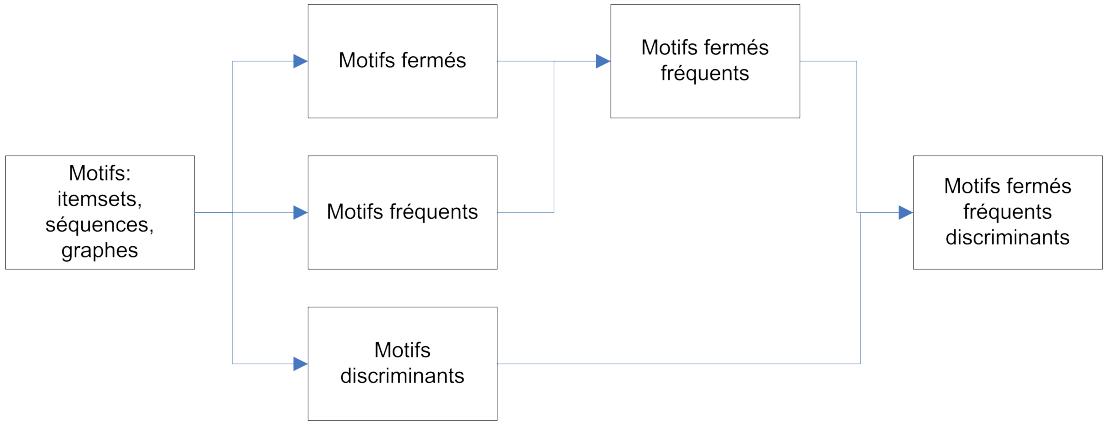


FIGURE 2.1 – Résumé des motifs utilisés

2.2 Définitions des notions de base

Les outils de recherche de motifs sont nombreux. Ils sont adaptés à des contextes divers tels que le domaine multidimensionnel (cf. [81]) ou spatial (cf. [82]) mais aussi à des données particulières. Les *itemsets* (cf. [78], [83], et [80]), les séquences (cf. [84], [85], et [86]), les arbres (cf. [87] et [88]) ou les graphes (cf. [89], [90] et [91]) sont des motifs fréquemment utilisés dans la fouille de données. Les propriétés particulières des motifs permettent de filtrer les motifs fermés (cf. [92] et [93]) ou maximaux (cf. [94]). Les motifs recherchés sont souvent très différents mais suivent le même principe. L'idée est de rechercher des éléments communs à un ensemble d'objets d'une base de données. Ces éléments sont alors représentatifs de la base de données, et sont utilisés pour résumer la base de données. Ce résumé est très intéressant lorsque la base de données étudiée est très grande mais aussi pour comprendre la tendance des données. Illustrons ce principe en utilisant le célèbre exemple du panier de la ménagère. Ce dernier a été étudié pour comprendre le comportement des clients lorsqu'ils font leurs courses dans les grandes surfaces. Les réponses à cette problématique permettent de mettre en place et d'améliorer les stratégies commerciales.

Exemple 14 L'analyse du panier de la ménagère est une application typique de l'extraction des règles d'association (cf. [95]) et peut être formulée sous la forme suivante : « Quels sont les produits qui sont fréquemment achetés ensemble par les clients ? » Sur la figure 2.2, une base de données d'achats des clients est présentée. Chaque ligne du tableau correspond à un ensemble d'articles achetés par un client. Le motif

Clients	Achats
Awa	pain, céréale, lait
Bob	pain, œufs, lait, sucre
Carl	pain, beurre, lait
Didier	œufs, sucre

FIGURE 2.2 – Base de données d'achats

{pain,lait} est retrouvé dans 75% des transactions. La base de données de la figure 2.2 est un ensemble de transactions, qu'on appelle aussi base transactionnelle ou base de transactions. Des définitions formelles seront proposées dans la suite.

Donnons des définitions formelles aux notions essentielles pour la suite du manuscrit en nous basant sur les définitions proposées dans [77] et [96]. Les définitions suivantes sont valables pour tous les types de motifs (séquences, graphes, *itemsets*...). Par analogie avec la notion d'attribut en Analyse Formelle des Concepts [2] (cf. [79]), la notion de motif est présentée via la définition de l'environnement d'extraction 2.2.2 proposée dans [96]. L'environnement d'extraction est un cadre aux définitions qui vont suivre.

Définition 2.2.1. ***Relation de support***
Un objet $O \in \mathscr{O}$ supporte un motif $M \in \mathscr{M}$ (ou M est supporté par O) s'il existe un couple (O, M) tel que $(O, M) \in \mathscr{R}$, la relation de support.

La relation de support 2.2.1 permet d'identifier quels motifs apparaissent dans quels objets.

Définition 2.2.2. ***Environnement d'extraction***
Le triplet $K = \{\mathscr{O}, \mathscr{M}, \mathscr{R}\}$ est appelé un environnement d'extraction, avec $\mathscr{O} = \{O_1, O_2, ..., O_n\}$ un ensemble de n objets, $\mathscr{M} = \{M_1, M_2, ..., M_p\}$ un ensemble de p motifs et $\mathscr{R}$ une relation de support entre $\mathscr{O}$ et $\mathscr{M}$ tel que $\mathscr{R} \subseteq \mathscr{O} \times \mathscr{M}$.

Exemple 15 Pour illustrer les définitions 2.2.2 et 2.2.1, le tableau 2.3 montre une base de données $\mathscr{D}$ composée de six objets tel que $\mathscr{D} =$

2. L'Analyse Formelle des Concepts est une branche de la théorie des treillis pouvant servir comme formalisme de base pour structurer les grands volumes de données. Elle étudie la structure induite par une relation binaire sur un couple d'ensembles.

	M_1	M_2	M_3	M_4
O_1		X		X
O_2	X	X	X	
O_3	X			X
O_4	X		X	X
O_5		X		
O_6	X			X

FIGURE 2.3 – Un environnement d'environnement $\{\mathscr{D}, \mathscr{M}, \mathscr{R}\}$

$\{O_1, O_2, O_3, O_4, O_5, O_6\}$ et un ensemble $\mathscr{M}$ de quatre motifs tel que $\mathscr{M} = \{M_1, M_2, M_3, M_4\}$. L'environnement d'extraction donnée par le tableau 2.3 indique le fait qu'un motif est supporté ou non par les objets de $\mathscr{D}$. Par exemple, l'objet O_1 supporte les motifs M_2 et M_4. Par contre l'objet O_6 ne supporte pas le motifM_2 mais supporte le motif M_4.

L'exemple 15 implique une définition du support d'un motif dans une base de données énoncée dans la définition 2.2.3.

Définition 2.2.3. ***Support d'un motif***

Soit $K = \{\mathscr{O}, \mathscr{M}, \mathscr{R}\}$ un environnement d'extraction. Le support du motif $M \in \mathscr{M}$ dans un ensemble d'objets $\mathscr{O}$, noté $Support_{\mathscr{O}}$(M), est défini comme le nombre d'objets dans $\mathscr{O}$ qui supportent M :

$$Support_{\mathscr{O}}(M) = |\{O \in \mathscr{O} | (O, M) \in \mathscr{R}\}|$$

Exemple 16 Dans le tableau 2.3, le support du motif M_2 est $Support_{\mathscr{D}}(M_2) = |\{O_1, O_2, O_5\}| = 3$ et le support du motif M_4 est $Support_{\mathscr{D}}(M_4) = |\{O_1, O_3, O_4, O_6\}| = 4$.

La notion de fréquence est naturellement introduite suite à la définition du support d'un motif. En effet la fréquence est couramment utilisée lorsqu'il s'agit de la proportion d'objets qui supportent un motif. Plus la fréquence d'un motif est élevée, plus ce motif est représentatif de la base de données.

Définition 2.2.4. ***Fréquence d'un motif***

Soit $K = \{\mathscr{O}, \mathscr{M}, \mathscr{R}\}$ un environnement d'extraction. La fréquence d'un motif $M \in \mathscr{M}$ dans $\mathscr{O}$, notée $Frequence_{\mathscr{O}}$(M), désigne le rapport entre le support du motif $M \in \mathscr{M}$ dans un ensemble d'objets $\mathscr{O}$ et le nombre total d'objets dans $\mathscr{O}$:

$$Frequence_{\mathscr{O}}(M) = \frac{Support_{\mathscr{O}}(M)}{|O|}$$

Exemple 17 Dans le tableau 2.3, la fréquence du motif M_2 est $Frequence_{\mathscr{D}}(M_2) = \frac{3}{6} = 50\%$ et la fréquence du motif M_4 est $Frequence_{\mathscr{D}}(M_4) = \frac{4}{6} \approx 66\%$.

Extraire tous les motifs à partir d'une base de données entraine une explosion combinatoire. Un seuil de support minimum limite le nombre de motifs extraits. L'ensemble des définitions précédentes ainsi que ce seuil de support minimum fondent la définition d'un motif fréquent.

Définition 2.2.5. ***Motif fréquent***
Soit $K = \{\mathscr{O}, \mathscr{M}, \mathscr{R}\}$ un environnement d'extraction et un entier θ tel que $0 < \theta < |\mathscr{O}|$. Un motif $M \in \mathscr{M}$ est un motif fréquent dans $\mathscr{O}$ si et seulement si $Support_{\mathscr{O}}(\mathrm{M}) \geq \theta$. Dans ce cas, θ est appelé le **seuil de support minimum**.

Exemple 18 Soit un seuil de support minimum $\theta = 3$. Dans le tableau 2.3, le motif M_4 est fréquent car $Support_{\mathscr{D}}(M_4) = 4 \geq \theta$.

La définition d'un motif fréquent est liée au support du motif mais il est possible de définir un motif fréquent en s'appuyant sur la fréquence d'un motif.

Dans la section 2.3, nous allons étudier un type de motif fréquent particuliers : les *itemsets* fermés fréquents.

2.3 *Itemsets* fermés fréquents

L'extraction des *itemsets* est utile pour l'analyse du panier de la ménagère. Elle consiste par exemple à déterminer les articles (appelés aussi les *items*) qui sont « souvent » achetés simultanément. Ce sont des *itemsets* fréquents. Ainsi un *itemset* est un ensemble d'*items* dépourvu de relation d'ordre. Une relation d'ordre dans un ensemble est une relation binaire dans cet ensemble qui permet de comparer ses éléments entre eux de manière cohérente. Un ensemble muni d'une relation d'ordre est un ensemble ordonné. Un *itemset* est donc un ensemble qui n'est pas ordonné.

Dans cette section, nous nous intéressons aux *itemsets* fermés fréquents. Pour définir un *itemset* fermé fréquent, nous allons donner les principaux concepts qui ont été formalisés dans [78] et utilisés dans [97] avant d'aborder les algorithmes d'extraction de ces objets et en particulier l'algorithme Charm (cf. [80]) .

2.3.1 Formalisation

Introduisons les définitions formelles à partir de l'exemple précédent.

Exemple 19 Le pain et le lait sont deux *items* et le motif {pain,lait} est un *itemset* de taille 2 et de fréquence 75 %. Pour un support de $\theta = 40\%$, l'*itemset* {pain,lait} est fréquent.

Le cadre classique du problème d'extraction des *itemsets* fermés fréquents est décrit comme suit : soit $\mathscr{I} = \{x_1, x_2, ..., x_n\}$ un ensemble d'*items*. Soit $\mathscr{D}$ un ensemble de transactions où chaque transaction est un ensemble d'*items*, identifiée de façon unique par un identifiant *tid*.

Définition 2.3.1. ***Un item*** est tout attribut, objet, article appartenant à un ensemble fini d'éléments distincts $\mathscr{I} = \{x_1, x_2, ..., x_n\}$.

Définition 2.3.2. ***Un itemset*** est défini comme tout sous-ensemble d'*items* de $\mathscr{I}$. Un *itemset* constitué de k *items* est appelé un k-*itemset*.

Définition 2.3.3. ***Un itemset fréquent***
Étant donné un seuil θ, appelé support minimum, un *itemset* X est dit fréquent (relativement à θ), si son support dépasse un seuil fixé *a priori* appelé support minimum et noté θ (cf. définition 2.2.5).

Définition 2.3.4. ***Une transaction*** est un *itemset* identifié par un identifiant unique appelé *tid*.

Définition 2.3.5. ***Une base de transaction*** ou une base transactionnelle $\mathscr{D}$ est un ensemble de couples formés d'un identificateur de transaction *tid* et de la transaction proprement dite.

Item	Produit
a	pain
b	œufs
c	beurre
d	sucre
e	céréale
f	lait

FIGURE 2.4 – Liste des articles achetés

Tid	Transaction
1	a, e, f
2	a, b, f, d
3	a, c, f
4	b, d

FIGURE 2.5 – Base de transaction du panier de la ménagère

Exemple 20 Le tableau de la figure 2.4 présente l'ensemble des articles achetés qui sont des *items*. A partir du tableau de la figure 2.2, construisons une base des transactions représentée par la figure 2.5. Chaque client est identifié par un numéro unique *tid*. L'ensemble des articles achetés par un client est une transaction. L'ensemble $\{a, e, f\}$ de la figure 2.5 est un *itemset* noté aussi *aef*, appelé 3-*itemset*.

Nous allons introduire des notions de la théorie des ordres afin de définir l'opérateur de fermeture. Ces notions sont extraits de [79].

Définition 2.3.6. ***Relation binaire***
Une relation binaire $\mathscr{R}$ entre deux ensembles $\mathscr{M}$ et $\mathscr{N}$ est définie par le produit cartésien $\mathscr{M} \times \mathscr{N}$ et consiste en une paire (m, n) avec $m \in \mathscr{M}$ et $n \in \mathscr{N}$. Lorsque $(m, n) \in \mathscr{R}$, on note $m\mathscr{R}n$. Si $\mathscr{M} = \mathscr{N}$ alors $\mathscr{R}$ est une relation binaire dans l'ensemble $\mathscr{M}$ (ou dans l'ensemble $\mathscr{N}$).

Définition 2.3.7. ***Relation d'ordre***
Une relation binaire $\mathscr{R}$ dans un ensemble $\mathscr{M}$ est une relation d'ordre si elle satisfait les conditions suivantes pour tout élément $x, y, z \in \mathscr{M}$:

- (réflexivité) $x\mathscr{R}x$;
- (antisymétrie) $x\mathscr{R}y$ et $x \neq y \Rightarrow$ pas $y\mathscr{R}x$;
- (transitivité) $x\mathscr{R}y$ et $y\mathscr{R}z \Rightarrow x\mathscr{R}z$.

Le symbole $\leq$ est souvent utilisé pour une relation d'ordre sur l'ensemble $\mathscr{M}$. L'exemple trivial d'un ensemble ordonné est l'ensemble des nombres réels R avec la relation $\leq$. Étant donné l'ensemble $\{1, 5, 5.2, 4, 2\}$, il est possible d'écrire $1 \leq 2 \leq 4 \leq 5 \leq 5.2$ Dans le cas d'un ordre total, chaque paire d'éléments peut être comparé. Ici $\leq$ est un ordre total. Il existe des cas pour lesquels tous les éléments ne sont pas comparables. On parle d'ordre partiel.

Définition 2.3.8. ***Ordre sur les items***

L'ensemble des *items* $\mathscr{I}$ étant fini, on choisit une application $f : \mathscr{I} \longmapsto \{1, ..., |\mathscr{I}|\}$ qui associe à chaque *item* $x \in \mathscr{I}$ un entier naturel. On a alors un ordre total noté $\ll < \gg$ sur les *items* de $\mathscr{I}$. On notera l'ensemble $\mathscr{I}$ muni de l'ordre $\ll < \gg$ entre *items* de $\mathscr{I}$ par $(\mathscr{I}, <)$.

Définition 2.3.9. ***Treillis***

Un ensemble ordonné $(\mathscr{T}, \leq)$ est un treillis si toute paire d'éléments de $\mathscr{T}$ possède une borne inférieure et une borne supérieure. On désignera par la suite la borne inférieure de la paire (x, y) par $(x \wedge y)$ et la borne supérieure par $(x \vee y)$. On notera la borne inférieure de $\mathscr{T}$ par $\perp$ et la borne supérieure par $\top$.

Définition 2.3.10. ***Treillis*** $(P(\mathscr{I}), \subseteq)$

L'ensemble $P(\mathscr{I})$ des parties d'un ensemble $\mathscr{I}$ muni de l'inclusion $\subseteq$ est un treillis. Les opérations binaires $\wedge$ et $\vee$ sont respectivement $\cap$ et $\cup$. Ce treillis admet une borne inférieure $\perp = \emptyset$ et une borne supérieure $\top = \mathscr{I}$.

> **Exemple 21** La figure 2.6 représente le treillis des *itemsets* pour $\mathscr{I} = \{a, b, c, d\}$. On retrouve au niveau 3 du treillis, tous les *itemsets* de taille 3, les 3-*itemsets*.

> **Exemple 22** Reprenons la figure 2.6 pour construire la figure 2.7. Sur celle-ci, le nombre entre parenthèse à côté de chaque *itemset* représente le support de cet *itemset* dans une base de transaction donnée. Supposons un seuil minimum de 3, alors les *itemsets* en gris sur la figure sont les *itemsets* fréquents.

Définition 2.3.11. ***Opérateur de fermeture***

Soit $\mathscr{S}$, un ensemble et ϕ une fonction tel que, $\phi : P(\mathscr{S}) \longrightarrow P(\mathscr{S})$. $P(\mathscr{S})$ est l'ensemble de tous les sous-ensembles de $\mathscr{S}$, incluant l'ensemble vide et $\mathscr{S}$ lui-même. ϕ est dit opérateur de fermeture sur $\mathscr{S}$, si pour tout $A, B \subset \mathscr{S}$, il est :

- extensive : $A \subset \phi(A)$;
- monotone : $A \subset B$ implique $\phi(A) \subset \phi(B)$;
- idempotente : $\phi(\phi(A)) = \phi(A)$.

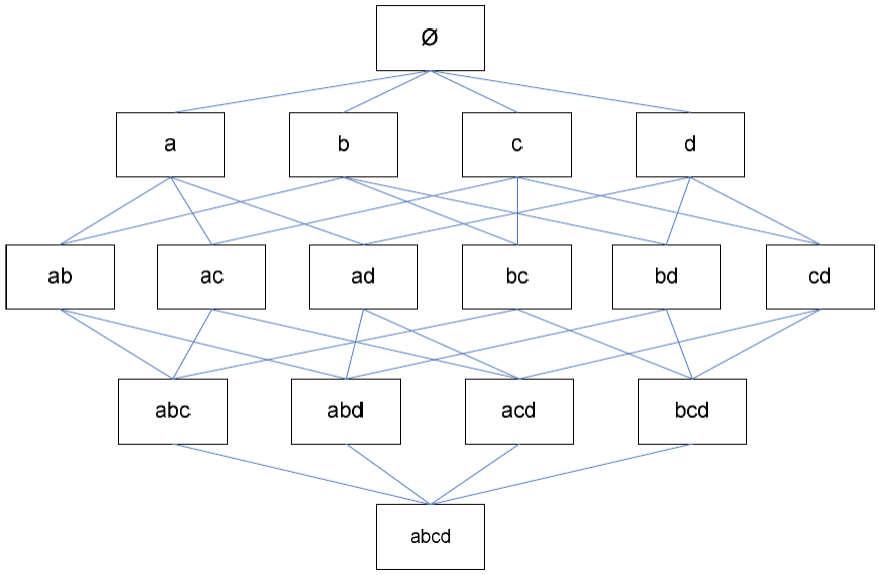


FIGURE 2.6 – Treillis des *itemsets*

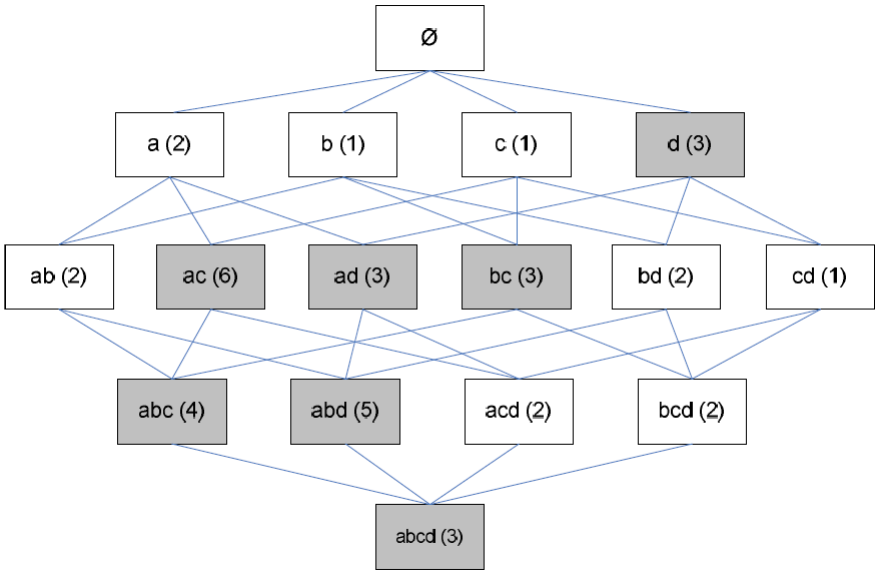


FIGURE 2.7 – Treillis des *itemsets* fréquents

Définition 2.3.12. ***Un superset*** est un *itemset* défini par rapport à un autre *itemset*. $\mathscr{I} = \{x_1, x_2, ..., x_n\}$ est un *superset* de $\mathscr{J} = \{x_1, x_2, ..., x_m\}$, si $Cardinal(\mathscr{J}) < Cardinal(\mathscr{I})$ et que $\mathscr{J} \subset \mathscr{I}$ c.à.d. on retrouve dans $\mathscr{I}$ tous les items de $\mathscr{J}$.

> **Exemple 23** Prenons un exemple simple : {abc} est un *superset* de {ab}.

La représentation basée sur les *itemsets* fermés est une représentation exacte. Elle se base sur la notion des ensembles fermés et tire son origine de la théorie des treillis et plus précisément des travaux autour de l'Analyse Formelle des Concepts (cf. [79]).

Définition 2.3.13. ***Un itemset fermé fréquent***

Un *itemset* fréquent est dit fermé si aucun de ses *supersets* n'a de support identique. Autrement dit, tous ses *supersets* ont un support strictement plus faible.

> **Exemple 24** Sur la figure 2.8, les *itemsets* en bleu sont des *itemsets* fermés fréquents. Par exemple, {ac} est fermé fréquent car aucun de ses *supersets* n'a de support égal à 6 : $Support_{\mathscr{D}}(\{abc\}) = 4$ et $Support_{\mathscr{D}}(\{acd\}) = 2$.

Il est possible de définir un *itemset* fermé en définissant un système de fermeture.

A partir de l'exemple de la figure 2.9 et pour un support minimum $\theta = 1$, le processus d'extraction des *itemsets* fréquents génèrerait $2^{100} - 1$ *itemsets* fréquents (soit à peu près 10^{30}). Dans [98], Pasquier et son équipe proposent une alternative à l'extraction de tous les *itemsets* fréquents. L'idée consiste à extraire un sous-ensemble d'*itemsets* fréquents fermés qui constituent « un ensemble générateur non redondant minimal pour tous les *itemsets* fréquents et leurs supports ». Ces *itemsets* fermés sont des représentations condensées de tous les *itemsets* fréquents. Ils résument l'information. A partir des fermés, il est possible de déduire le support de n'importe quel *itemset* fréquent sans recourir au parcours de la base de transactions.

Plusieurs algorithmes ont été proposés pour l'extraction d'*itemsets* fermés fréquents. Citons-en quelques uns : Close (cf. [99], [100] et [98]), Charm (cf. [80]), Pascal (cf. [101] et [102]), Titanic (cf. [103] et [104]) et Closet (cf. [1]). Nous verrons dans ce qui suit le fonctionnement des algorithmes Close et Closet avant de décrire un algorithme en particulier, l'algorithme Charm proposé par Zaki et *al.* en 2002. Notons que des améliorations de Charm, tels que LCM (cf. [105]) et DCI-CLOSED (cf. [106]), ont été aussi proposées.

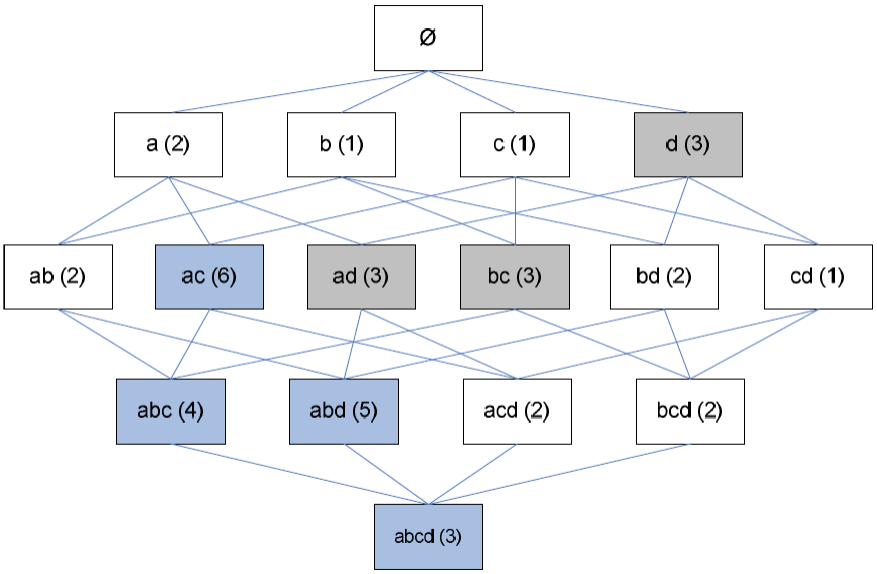


FIGURE 2.8 – Treillis des *itemsets* fermés fréquents

Tid	Transaction
1	$x_1, x_2, ..., x_{100}$
2	$x_1, x_2, ..., x_{50}$

FIGURE 2.9 – Un exemple de base de transaction inspiré de [1]

2.3.2 Extraction d'*itemsets* fermés fréquents

Dans cette sous-section, nous présentons quelques algorithmes retenus parmi les nombreux algorithmes d'extraction d'*itemsets* fermés fréquents de la littérature. Il est possible de classer les algorithmes d'extraction d'*itemsets* fermés fréquents selon la technique adoptée lors de l'exploration de l'espace de recherche. Dans [107], les auteurs rappellent deux catégories d'algorithme : « Tester-et-générer » et « Diviser-pour-régner ».

- « Tester-et-générer » : l'espace de recherche est parcouru par niveau par les algorithmes. Un ensemble de candidats de taille k est généré, à chaque niveau k. Cet ensemble de candidats est, généralement, élagué par la conjonction d'une

métrique statistique comme le support et des heuristiques basées essentiellement sur les propriétés structurelles des *itemsets* fermés.

- « Diviser-pour-régner » : le contexte d'extraction est divisé en des sous-contextes par les algorithmes. Ensuite le processus de découverte des *itemsets* fermés récursivement sur ces sous-contextes est appliqué par ces derniers. Ce processus de découverte repose sur un élagage du contexte basé essentiellement sur l'imposition d'une métrique statistique et d'heuristiques introduites.

L'algorithme Close est un algorithme de type « Tester-et-générer » tandis que Closet est de type « Diviser-pour-régner ». Charm appartient aux deux catégories. L'algorithme Charm est de type hybride. Nous allons les décrire dans la suite.

Close

L'algorithme itératif Close a été proposé par Pasquier et son équipe (cf. [99], [100] et [98]) pour extraire les *itemsets* fermés fréquents. Il utilise un mécanisme de fermeture basé sur l'Analyse Formelle des Concepts (AFC). Rappelons que l'AFC est une branche de la théorie des treillis. Elle étudie la structure induite par une relation binaire sur un couple d'ensembles, comme par exemple les items et les transactions. Ainsi, le treillis des *itemsets* peut être perçu comme un treillis complet dit treillis de Galois ou treillis de concepts (cf. [108]). L'algorithme Close est basé sur la notion d'*itemset* générateurs des *itemsets* fermés. Ces *itemsets* générateurs sont utilisés afin de construire un ensemble d'*itemsets* fermés candidats (*itemsets* fermés potentiellement fréquents) qui sont les fermetures des générateurs.

Définition 2.3.14. Itemsets *générateurs d'un* itemset *fermé*
Un *itemset* générateur g d'un *itemset* fermé $f = \gamma(f)$ est un *itemset* minimal (selon la relation d'inclusion) dont la fermeture par l'opérateur γ est l'*itemset* fermé $\gamma(g) = f$. L'ensemble G_f des générateurs d'un *itemset* fermé f est :

$$G_f = \{g \subseteq \mathscr{I} | \gamma(g) = f \wedge \nexists g\prime \subset g, \gamma(g\prime) = f\}.$$

Durant chaque itération, l'algorithme Close considère un ensemble de k-*itemsets* générateurs. Il construit un ensemble d'*itemsets* fermés candidats qui sont les fermetures de ces k-générateurs et il détermine ensuite parmi ces candidats les *itemsets* fermés fréquents selon le seuil minimal de support. Finalement, il crée les (k+1)-générateurs qui seront utilisés lors de l'itération suivante afin de construire l'ensemble d'*itemset* fermés candidats qui sont les fermetures des (k+1)-générateurs. Un balayage du contexte d'extraction est nécessaire durant chaque itération, afin de déterminer les fermetures des k-générateurs et calculer leurs supports (cf. [109]).

Closet

Closet, de type « Diviser-pour-régner », a été proposé en 2000 par Jian Pei et son équipe (cf. [1]). Dans la littérature, des améliorations de Closet ont été proposées (cf. [110]). Cet algorithme permet l'extraction d'*itemsets* fermés fréquents en se basant sur trois techniques en particuliers :

- extraction d'*itemsets* fermés sans générer les candidats en utilisant une structure de données compacte FP-tree (*frequent pattern tree*) ;
- identification des *itemsets* fermés fréquents rapidement en développant une technique de compression d'un chemin préfixe unique ;
- exploration d'un mécanisme de projection à base de partition.

Avec cet algorithme, la génération d'un nombre important de candidats par les algorithmes de type « Tester-et-générer » est évitée. La structure de données avancée, FP-tree, utilisée par Closet permet de compresser la base de données pour arriver à achever le processus de fouille de données. L'idée derrière l'utilisation de cette structure de données compactes(cf. []), est que lorsque plusieurs transactions se partagent un item, alors elles peuvent être fusionnées en prenant soin d'enregistrer le nombre d'occurrences des items. Le processus d'extraction des *itemsets* fermés se fait en deux étapes successives par Closet :

1. Les items des transactions sont ordonnés par support décroissant. Ensuite, le FP-tree est construit. Premièrement, le nœud racine est créé et étiqueté par « root ». Pour chaque transaction du contexte, les items sont traités et une branche est créée pour chaque transaction. Dans chaque nœud du FP-tree, il y a un compteur qui garde trace du nombre d'apparitions de l'*item* dans le contexte d'extraction. Spécifiquement dans le cas où une transaction présente un préfixe commun avec une branche du FP-tree, alors le compteur de chaque nœud appartenant à ce préfixe est incrémenté de 1 et une sous-branche va être créée contenant le reste des items de la transaction.

2. Au lieu d'une exploration en largeur des *itemsets* fermés candidats, il effectue une partition de l'espace de recherche pour effectuer ensuite une exploration en profondeur d'abord. Ainsi, il commence par considérer les 1-*itemsets* fréquents, et examine seulement leur sous-contexte conditionnel. Un sous-contexte conditionnel ne contient que les items qui co-occurrent avec le 1-*itemset* en question. Le FP-Tree conditionnel associé est construit et le processus se poursuit récursivement.

Exemple 25 Par exemple, la figure 2.10 illustre un contexte d'extraction trié et la structure FP-tree associée pour un support minimum, minsup=2.

	B	C	E	A
1		×		×
2	×	×	×	
3	×	×	×	×
4	×		×	
5	×	×	×	×

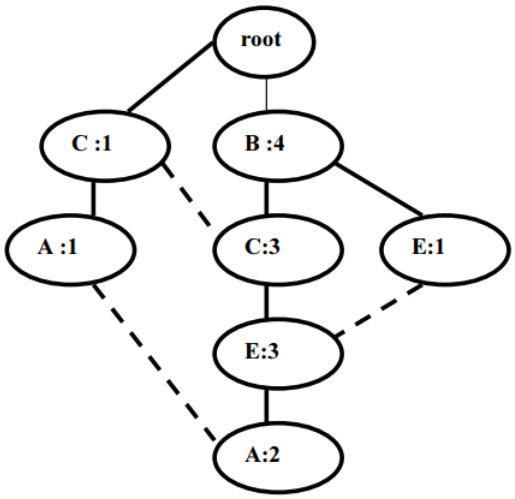


FIGURE 2.10 – Résumé des motifs utilisés

Cette description de l'algorithme Closet ainsi que l'exemple de la figure 2.10 est issue de [107].

Dans [107], les performances de l'algorithme Closet ont été évaluées comparativement à l'algorithme Charm. Les résultats de ces expérimentations ont montré que l'algorithme Closet est plus performant que l'algorithme CHARM sur les contextes épars et denses. Les auteurs de l'algorithme Closet ne présentent pas une technique de gestion du buffer, d'après [107]. Dans ce cas, toute l'information nécessaire pour le déroulement du processus de découverte des *itemsets* fermés ne peut tenir en mémoire centrale. Pour les bases de données faiblement corrélées et denses, l'arbre représentant la trace d'exécution a tendance à augmenter.

Charm

L'algorithme Charm (Close Association Rules Mining), proposé par Zaki et al., a retenu notre attention en particulier parce que dans [107], l'étude comparative d'algorithmes d'extraction d'*itemsets* fermés fréquents les plus populaires, révèle les bonnes performances de Charm. Ceci est une raison qui justifie notre choix entre autres. L'algorithme Charm comme tous les algorithmes d'extraction d'*itemsets* fermés fréquents, constitué en deux étapes permet d'abord d'extraire les *itemsets* fermés fréquents et ensuite les règles d'association. Nous nous focalisons essentiellement sur l'étape de découverte des *itemsets* fermés fréquents. Charm est un algorithme qui utilise les stratégies des deux catégories citées plus haut. En effet il n'énumère que les *itemsets* fermés fréquents donc il se garde de générer un nombre élevé de candidats. Charm explore en profondeur l'espace de recherche. Le parcours en profondeur rappelle la stratégie « Diviser-pour-régner ». Charm ne divise pas le contexte d'extraction en sous-contextes, évitant ainsi une saturation de la mémoire comme dans la technique « Diviser-pour-régner ». Charm explore simultanément l'espace des *itemsets* fermés et celui des transactions. Zaki et *al.* (cf. [80]) utilise une structure d'arbre particulière appelée *Itemset-Tidset Tree*. Chaque nœud de l'arbre est composé du couple suivant : un *itemset* fermé fréquent candidat et l'ensemble des transactions auxquelles il appartient qui est l'ensemble des *tid*, appelé *tidset*. En parcourant l'arbre à partir de la racine de gauche à droite et en profondeur, Charm détermine tous les *itemsets* fermés fréquents en considérant un par un les 1-*itemsets* fréquents triés par ordre lexicographique. Notons que Charm utilise une représentation verticale, appelée *diffset*, pour accélérer le calcul des supports. En effet, la représentation de données *diffset* permet un stockage incrémental des *tidsets*. En revanche, l'inconvénient majeur de l'algorithme Charm réside dans le fait qu'il est exigeant en espace mémoire. En effet, le fait de stocker les *itemsets* et leurs *tidsets* ne fait qu'accroitre la quantité d'espace mémoire utilisée (cf. [107]).

2.4 Outils de discrimination

Les *itemsets* fermés fréquents extraits permettent de mettre en place la classification d'objets appelés sessions [3]. Afin d'optimiser le processus de classification, il est nécessaire que chaque classe soit définie par des *itemsets* fermés fréquents discriminants.

Des algorithmes comme Charm ou Closet permettent de générer un ensemble d'*itemsets* fermés fréquents, parfois trop volumineux. Pour récupérer les *itemsets*

3. Rappelons qu'une session est un ensemble de pages visités de taille fixe (cf. chapitre 1).

fermés fréquents les plus pertinents et discriminants pour une problématique donnée, il est nécessaire d'utiliser certains outils statistiques. Pour déterminer le caractère discriminant des motifs, l'auteur utilise les critères suivants dans [77] : la fréquence, la discriminance, et la non redondance. Nous adaptons les définitions données dans [77] à notre problématique. Dans notre étude, la fréquence est définie par le nombre de sessions dans chaque classe de la base de données qui supportent un *itemset* fermé fréquent en fonction du nombre total de sessions dans ces classes. Les *itemsets* fermés fréquents les plus récurrents dans chaque classe de la base de données sont les plus intéressants car ces *itemsets* fermés fréquents décrivent mieux une classe donnée que les *itemsets* fermés fréquents avec une faible fréquence. Dans [77], la discriminance est définie par le taux d'émergence dans chaque classe de la base de données. En effet, une valeur statistique appelée taux d'émergence, en anglais *growth rate*, caractérise un motif discriminant. Cette valeur se base sur la fréquence des motifs et donc peut être appliquée à tout type de motifs. Dans notre cas, cette mesure calcule un rapport sur la fréquence d'un *itemset* fermé fréquent dans une classe par rapport à une autre. Les *itemsets* fermés fréquents les plus discriminants dans une classe, ont une forte valeur de *growth rate* pour cette classe, correspondant aux motifs qui sont plus fréquents dans une classe par rapport à une autre. Enfin la redondance est caractérisée par le fait que deux *itemsets* fermés fréquents soient supportés par un nombre élevé de sessions communes. Réduire cette redondance dans les résultats permet d'améliorer l'analyse des motifs. Plus la redondance entre deux motifs est faible, plus ils sont susceptibles de donner une information différente sur la base.

Dans notre étude, nous utilisons des outils basés sur le critère de fréquence, de discriminance [4] et de non-redondance.

Les outils statistiques comme le test de χ^2 permettent de voir le degré de corrélation entre les *itemsets* fermés fréquents. Souvent utilisé en recherche d'information et en particulier dans la fouille de textes, le $tf \times idf$ (de l'anglais *Term Frequency-Inverse Document Frequency*) est une méthode de pondération. Cette mesure statistique permet d'évaluer l'importance d'un terme contenu dans un document, relativement à une collection ou un corpus. Cette mesure peut être utilisée pour définir et collecter des *itemsets* fermés fréquents discriminants. La stabilité est une mesure de qualité permettant aussi d'extraire des *itemsets* fermés fréquents discriminants pour notre étude. Dans la suite, rappelons la définition du test de χ^2 et le $tf \times idf$ avant d'introduire la stabilité des *itemsets* fermés.

4. Nous n'utiliserons pas le taux d'émergence dans notre étude mais l'idée de discriminance est conservée lors du choix des outils pour trouver les motifs discriminants

2.4.1 Outils statistiques

Le test du χ^2 est un test en statistiques, permettant de mesurer le degré de corrélation entre différents événements (attributs, motifs...). Le test compare la distribution réelle de leur occurrence avec la distribution attendue, sous l'hypothèse d'une distribution normale et d'une indépendance totale. Le test de χ^2 est utilisé pour mesurer la solidité des règles d'association (cf. [111]), dans le cadre de la généralisation des règles d'associations aux règles de corrélation. Dans notre étude, nous proposons de l'utiliser pour extraire les *itemsets* fermés fréquents les plus discriminants. Plus spécifiquement, pour la sélection des *itemsets* fermés fréquents les plus discriminants, on utilise χ^2 pour tester si l'occurrence d'un *itemset* fermé fréquent spécifique et l'occurrence d'une classe spécifique sont indépendants. Pour chaque *itemset* fermé fréquent, le calcul de χ^2, qui donne un score, est réalisé.Ensuite les scores calculés de chaque *itemset* fermé fréquent sont ordonnés de manière décroissante. Les scores élevés de χ^2 indiquent que l'occurrence de l'*itemset* fermé fréquent et de la classe sont dépendants. Si elles sont dépendantes, on garde l'*itemset* fermé fréquent comme descripteur pour la classification. Cette *itemset* fermé fréquent est discriminant. En utilisant la méthode du χ^2, il est nécessaire de ne sélectionner qu'un nombre précis de descripteurs qui ont un score supérieur à un seuil permettant d'avoir des résultats statistiquement significatifs.

En statistiques, un résultat est dit statistiquement significatif lorsqu'il est improbable qu'il puisse être obtenu par un simple hasard. Habituellement, on utilise un seuil de signification (une probabilité notée α) de 0,05 (seuil de référence en psychologie, les sciences biomédicales utilisent un seuil plus restrictif à 0,01 voire 0,001), ce qui signifie que le résultat observé a moins de 5 % de chances d'être obtenu par hasard. Il est donc jugé statistiquement significatif. Le seuil observé (*p-value*, en anglais) peut être calculé et il correspond à la probabilité que le résultat obtenu soit le fruit du hasard. Cependant, de nouvelles recherches montrent qu'un test statistiquement significatif ne correspond à une évidence forte que pour une *p-value* de 0,5 % ou même 0,1 %, c'est-à-dire une chance sur mille d'être dû au hasard. Le test du χ^2 reste cependant une mesure difficile à calculer pour chaque *itemset* fréquent, car elle demande beaucoup de calculs. De plus la précision du χ^2 se dégrade lorsque les valeurs attendues sont faibles (cf. [111]).

Il existe d'autres outils pour sélectionner des *itemsets* fermés fréquents discriminants (cf. [112]), mais notre choix s'est porté sur le test de χ^2, le $tf \times idf$ et la mesure de la stabilité. En effet, nous balayons un large spectre en empruntant ces mesures à différents domaines : les statistiques, la recherche d'information et l'Analyse Formelle des Concepts.

Le $tf \times idf$ est une mesure, utilisée en recherche d'information, introduite dans [113] en 1989. Cette mesure est basée sur un système de pondération et permet d'estimer l'importance d'un terme contenu dans un document, relativement à une collection ou un corpus. Cette mesure est basée sur la fréquence. Le poids augmente proportionnellement au nombre d'occurrences du mot dans le document. Il varie également en fonction de la fréquence du mot dans le corpus. Si une requête contient le terme T, un document a d'autant plus de chances d'y répondre qu'il contient ce terme : la fréquence du terme au sein du document (TF ou *term frequency*) est grande. Néanmoins, si le terme T est lui-même très fréquent au sein du corpus, c'est-à-dire qu'il est présent dans de nombreux documents (par exemple les articles définis - le, la, les), il est en fait peu discriminant. C'est pourquoi le schéma propose d'augmenter la pertinence d'un terme en fonction de sa rareté au sein du corpus (fréquence du terme dans le corpus IDF ou *inverse document frequency* élevée). Ainsi, la présence d'un terme rare de la requête dans le contenu d'un document fait croître le « score » de ce dernier.

2.4.2 Stabilité des *itemsets* fermés fréquents

Plusieurs mesures sont utilisées afin de trouver des motifs intéressants en fonction des problématiques. En général, ces mesures calculent des scores qui sont ensuite classés par ordre décroissants. Les motifs les mieux classés sont sélectionnés pour devenir des descripteurs dans un processus de classification. La mesure de stabilité introduite par [114] est une mesure de qualité pour des motifs (ou plus précisément des concepts formels) issus de l'Analyse Formelle des Concepts (cf. [115]). Cette mesure est adaptée aux *itemsets* fermés. La stabilité d'un *itemset* fermé X calcule le nombre relatif de sous-ensembles de l'ensemble d'objets $\mathcal{O}$, qui peut être supprimé de la base de données en préservant X. La stabilité est une mesure fiable lorsqu'il s'agit de base de données d'une mauvaise qualité (cf. [116]), mais pour les larges treillis, il est nécessaire d'utiliser des estimations (cf. [115]). Elle se calcule de la manière suivante :

$$Stab(X) = \frac{|\{Y \subseteq O |\ X\ est\ ferme\ dans\ (Y, M, I \cap Y \times M)\}|}{2^{|O|}} \tag{2.1}$$

Exemple 26 Sur l'exemple 2.11, considérons le motif $\{M_4\}$. Le nombre de sous-ensembles de l'ensemble d'objets $\mathcal{O}$, qui peut être supprimé de la base de données en préservant $\{M_4\}$ est $2^4 = 16$. La stabilité de $\{M_4\}$ est :

$$Stab(\{M_4\}) = \frac{16 - (1 + 4 + 3)}{16} = 0,5$$

	M_1	M_2	M_3	M_4	M_5
O_1	X			X	
O_2		X		X	
O_3			X	X	
O_4					X

FIGURE 2.11 – Une base de données $\{\mathscr{O}, \mathscr{M}, \mathscr{R}\}$

Si l'ensemble d'objets $\mathscr{O}$ est réduit à $\{O_1\}$ ou $\{O_2\}$ ou $\{O_3\}$ ou $\{O_4\}$ alors le motif $\{M_4\}$ n'existe plus. Par exemple, si $\mathscr{O} = \{O_1\}$ alors $\{M_4\}$ n'est pas fermé tandis que si $\mathscr{O} = \{O_4\}$ alors le motif ne supporte plus aucun objet. De plus $\{M_4\}$ n'est pas un motif dans l'ensemble vide et dans les ensembles suivants : $\{O_1, O_4\}$, $\{O_2, O_4\}$, et $\{O_3, O_4\}$.

2.5 Résumé et discussion

Dans ce chapitre, nous avons défini les outils nous permettant d'introduire la suite de notre étude. Les bases de l'Analyse Formelle de Concepts ont été établies afin d'introduire l'*itemset* fermé fréquent. Ensuite, nous avons présenté des algorithmes d'extractions d'*itemsets* fermés fréquents de la littérature. Nous en avons retenu trois, l'algorithme Close (cf. [99]), l'algorithme Closet (cf. [1]) et l'algorithme Charm (cf. [80]). Les avantages et les inconvénients ont été précisés. Pour la suite, nous avons décidé de choisir l'algorithme Charm pour extraire ces objets de notre base de données. Afin de ne travailler qu'avec les *itemsets* fermés fréquents les plus pertinents, nous utilisons des outils statistiques afin de réduire le nombre des résultats. Le test de χ^2 ainsi que le $tf \times idf$ sont définis dans ce chapitre. La mesure de la stabilité des *itemsets* fermés fréquents est enfin exposée.

Des questions peuvent être soulevées sur le choix des outils pour répondre à la problématique. Concernant l'objet « *itemsets* fermés fréquents », la littérature offre une palette d'outils qui pourraient être adéquats à notre problématique. Après avoir essayé plusieurs algorithmes d'extraction des *itemsets* fermés fréquents, Charm nous donne entière satisfaction. Ce choix est effectivement ouvert au débat d'autant plus qu'il existe des améliorations de Charm (cf. [105] et [106]). Enfin les outils statistiques, pour l'extraction des *itemsets* fermés fréquents discriminants à partir de l'ensemble des *itemsets* fermés fréquents, est assez intuitive. En effet le $tf \times idf$ est un outil populaire basé sur la fréquence, et le test de χ^2 permet de définir l'indépendance entre deux événements. Enfin, utilisant les objets de l'Analyse Formelle des Concepts,

il semble cohérent de se tourner vers des outils propres au domaine tel que la mesure de la stabilité.

Chapitre 3

Classification

Sommaire

1.1 Introduction 20

1.2 Définitions 24

1.2.1 Confiance à base de politiques 25

1.2.2 Réputation et recommandation 27

1.2.3 Confiance et sciences sociales 32

1.2.4 Contraintes et qualités du modèle à concevoir 36

1.3 Exemple de modèle de confiance 37

1.3.1 Contexte de l'étude 37

1.3.2 Cahier des charges : les usages du modèle 43

1.3.3 Conception 47

1.3.4 Limites et extensions 55

1.4 Authentification implicite 56

1.4.1 Identification et authentification 57

1.4.2 Définitions et concepts 60

1.4.3 Authentification implicite dans eBee 63

1.5 Résumé et discussion 64

The capacity of digital data storage worldwide has doubled every nine months for at least a decade, at twice the rate predicted by Moore's Law for the growth of computing power during the same period.

Fayyad et al, 2002.

3.1 Introduction

La classification permet de mettre en place un système organisé et hiérarchisé de catégorisation d'objets (par exemple : la classification périodique des éléments du chimiste russe Dmitri Mendeleïev construite en 1869). Elle apporte un éclairage sur la nature de la connaissance.

En fouille de données, la classification est un domaine de recherche très vaste qui englobe différents cas d'utilisation et qui répond à des besoins divers. Des informations, qui permettent de prendre des décisions importantes sont cachées dans les bases de données riches. La classification permet d'analyser ce type de bases de données afin d'en extraire des modèles qui les décrivent ou qui prédisent leur comportement à venir. D'ailleurs la classification permet de mieux comprendre les grandes bases de données. Dans le processus d'extraction de connaissance présenté dans l'introduction de ce manuscrit, la classification intervient dans l'étape de fouille de données. En effet, dans cette étape, des modèles de prédiction ou des modèles descriptifs peuvent être utilisés. Les modèles prédictifs analysent les événements présents et passés pour prédire des événements futurs. Les modèles descriptifs identifient comment certaines variables dépendent des autres. Par exemple, la classification et la régression [1] sont des modèles prédictifs tandis que le partitionnement ou *clustering* [2] en anglais est un modèle descriptif, même s'il peut être utilisé pour faire de la prédiction [3] (cf. figure 3.1). Plus précisément, la classification prédit l'appartenance d'un objet à des classes catégorielles. Rappelons le fonctionnement de la classification de manière générale. La base de données sur laquelle l'étude est faite est partagée en deux sous-ensembles : une base d'apprentissage (en anglais *training database*) et une base de test (en anglais *test database*). Le processus de classification se décline en deux étapes. Durant la

1. Le modèle de prédiction a une sortie numérique lorsqu'il s'agit d'une régression tandis qu'il a une sortie catégorielle lorsqu'il s'agit d'une classification.
2. Les événements similaires sont regroupés dans une partition ou *cluster* lors du *clustering*.
3. http ://www.saedsayad.com/, accès 19/10/2015.

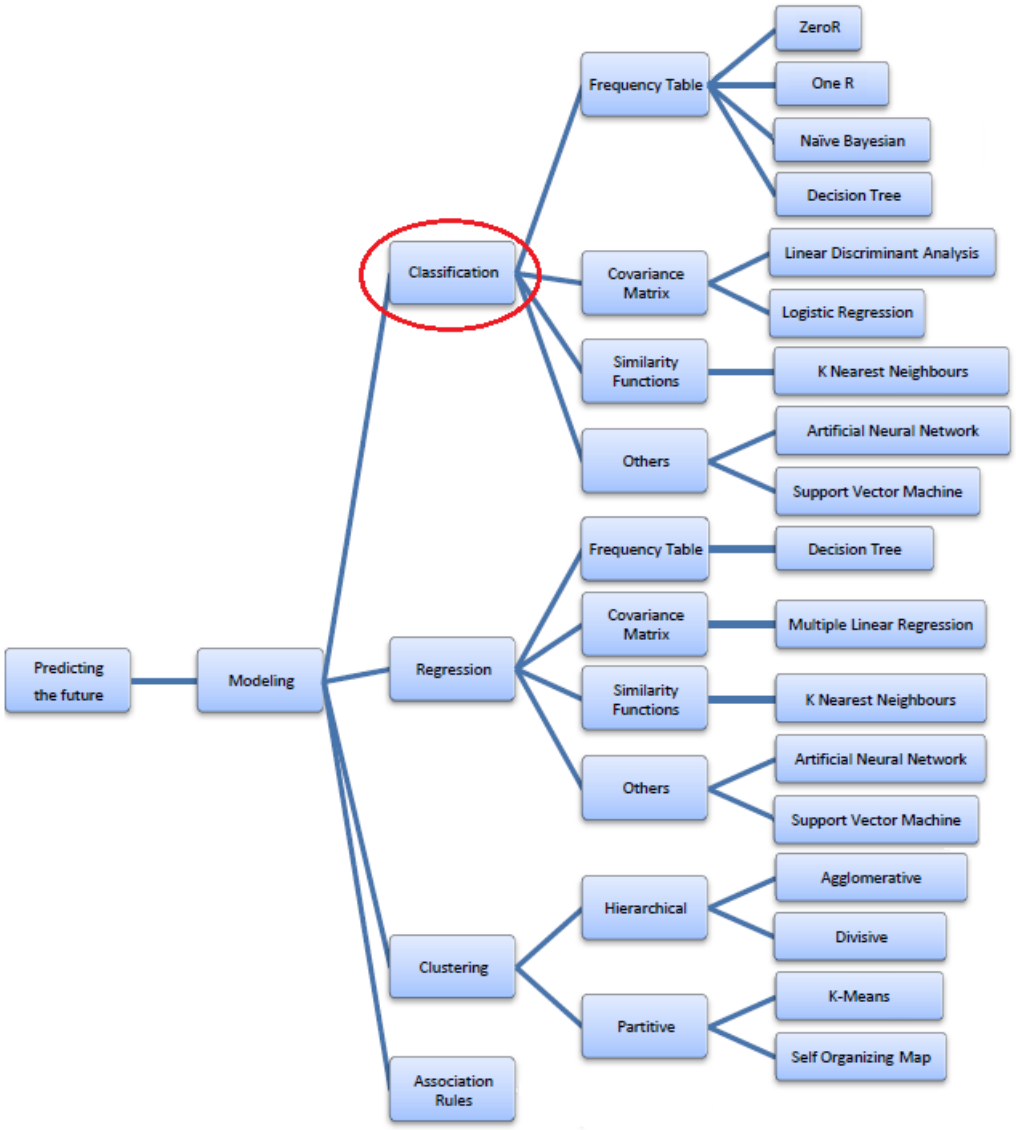
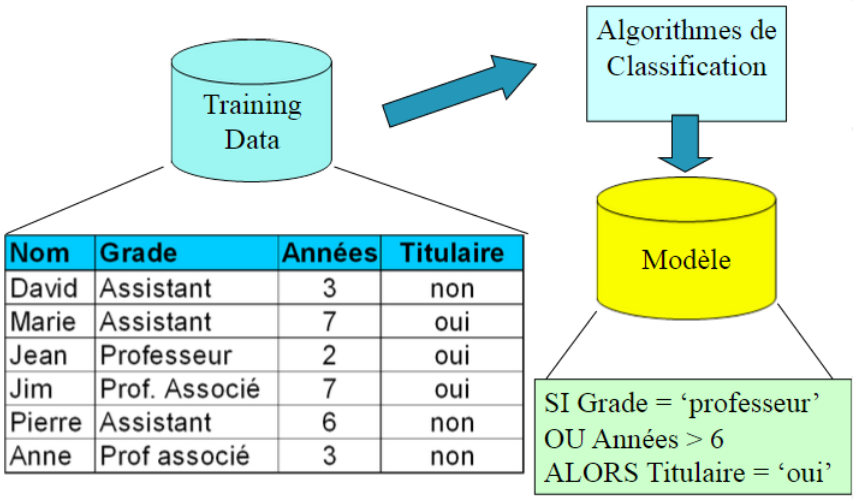


FIGURE 3.1 – Hiérarchisation des modèles de prédictions.

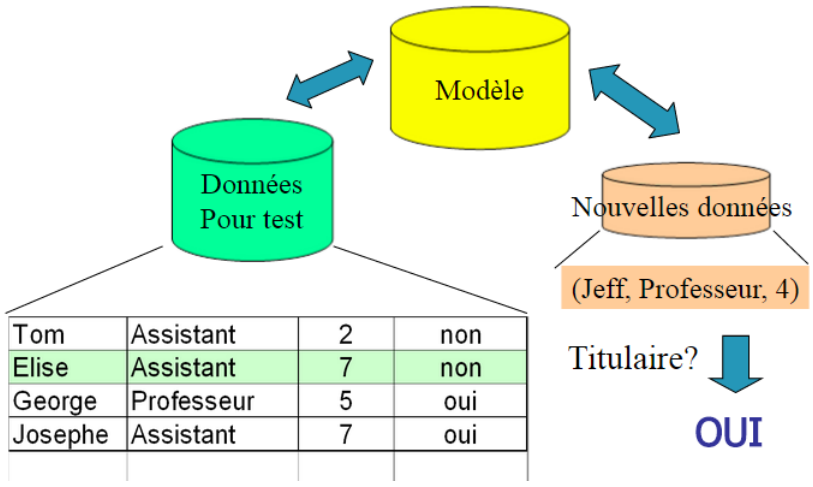
phase d'apprentissage, un modèle de classification (arbres de décision, règles...) est construit à partir d'une base de données dite base de données d'apprentissage. Dans cette base, l'association données-classes est connue. Chaque exemple ou tuple ou encore observation est supposé appartenir à une classe. Durant la phase de test, le modèle construit est utilisé pour classer les données de test. Une mesure de précision

estime le pourcentage de tuples correctement classifiés par le modèle construit. Les schémas des figures 3.2 et 3.3 illustrent le fonctionnement de la classification.



Nom	Grade	Années	Titulaire
David	Assistant	3	non
Marie	Assistant	7	oui
Jean	Professeur	2	oui
Jim	Prof. Associé	7	oui
Pierre	Assistant	6	non
Anne	Prof associé	3	non

FIGURE 3.2 – Le processus de classification : construction du modèle



Tom	Assistant	2	non
Elise	Assistant	7	non
George	Professeur	5	oui
Josephe	Assistant	7	oui

FIGURE 3.3 – Le processus de classification : phase de test

Exemple 27 Sur la figure 3.2, le modèle, construit à partir de la base d'apprentissage est une règle : « si l'individu est un professeur ou a plus

de six ans d'expériences alors il est titulaire ≫. Durant la phase de test sur la figure 3.3, Jeff qui est professeur et a quatre ans d'expériences est classé comme titulaire. Deux classes existent dans cette exemple : ≪ oui ≫ ou ≪ non ≫.

Les méthodes de classification proposées par les chercheurs en statistique, en apprentissage automatique (en anglais *machine learning*) ou en reconnaissance de modèle (en anglais *pattern recognition*) sont diverses et variées (cf. [117] et [118]). De nombreuses méthodes de classification sont conçues pour traiter plusieurs types d'objets et peuvent être appliquées dans différents contextes (classification d'images, de textes...). Ces approches requièrent souvent des connaissances *a priori* sur les données comme par exemple le nombre des classes. Citons des méthodes de classification basiques : les arbres de décisions (cf. [119]), les classifieurs à base de règles d'association comme CBA (cf. [120]) ou CMAR (cf. [121]), ou encore les classifieurs naïfs bayésiens basés sur le théorème de Bayes.

Dans notre travail, nous utilisons la classification naïve bayésienne et la classification à base d'*itemsets* fermés fréquents discriminants. Dans cette section, nous présentons les travaux effectués lors de cette thèse. Ces travaux ont déjà été présentés lors d'une conférence (cf. [122]). Le contexte de l'étude est l'authentification implicite. La classification à base d'*itemsets* fermés fréquents discrminants proposée a pour objectif d'authentifier un utilisateur au sein de l'espace de vie numérique eBeeHome. En effet elle permet d'identifier l'utilisateur propriétaire d'un comportement. L'authentification implicite se base sur cette identification au préalable. Plus précisément, nous avons besoin de classer les sessions des utilisateurs d'eBeeHome.

Nous allons présenter les motivations sous-jacentes à ces travaux dans 3.2. Ensuite, nous présenterons des variantes de la classification naïve bayésienne adaptées à notre problématique dans 3.3. La classification à base d'*itemsets* fermés fréquents discriminants est ensuite exposée dans la section 3.4. Un résumé du chapitre est donné dans 3.5.

3.2 Motivations : limites des approches existantes

Rappelons que la problématique de l'étude est de prédire l'utilisateur propriétaire d'une session afin de l'authentifier implicitement. Les utilisateurs sont les classes du problème de classification. Nous avons autant de classes que d'utilisateurs. La classification est une solution naturelle lorsqu'il s'agit d'extraire de la connaissance à partir de données. L'utilisation de la classification à base d'*itemsets* fermés fréquents discriminants pour répondre à notre problématique est une approche proposée après

avoir exploré plusieurs méthodes de classification populaires. Il existe plusieurs méthodes de classifications dans la littérature. Ici, nous présentons quelques méthodes et leurs limites : la classification à base de règles (cf. [120] et [121]), la classification à base d'arbres de décision (cf. [119]) et la classification naïve bayésienne. Intuitivement, ces trois méthodes de classification sont les plus adaptées à notre problématique et leurs performances sont avérées dans la littérature. Ce sont des méthodes d'apprentissage supervisé c'est à dire que le nombre de classes est connu. La contribution de nos travaux par rapport à ces méthodes est abordée dans cette section également.

Classification à base d'arbres de décision

Méthodes d'apprentissage populaires, les arbres de décisions sont introduits en 1986 dans [119]. Par exemple, pour classer les sessions dans des classes utilisateurs, il est nécessaire de construire un arbre de décision par utilisateur. Pour déterminer à quel(s) utilisateur(s) appartient une nouvelle session, on utilise l'arbre de décision de chaque utilisateur auquel on soumet la session à classer. Chaque arbre prend une décision (répond « Oui » ou « Non »). Concrètement, chaque nœud d'un arbre de décision contient un test (un SI...ALORS) et les feuilles ont les valeurs « Oui » ou « Non ». Chaque test regarde la valeur d'un attribut de chaque classe. En effet, on suppose qu'une classe est un ensemble d'attributs. Dans le cas de notre étude, chaque attribut peut être une page web visitée ou une fonctionnalité utilisée dans eBeeHome, et la valeur sera par exemple 0 ou 1 selon que cette page web appartient ou non à la session ou que la fonctionnalité a été utilisée ou non durant la session. Pour construire l'arbre de décision, il faut trouver quel attribut tester à chaque nœud. Pour résumer, un arbre de décision représente la suite de questions à poser pour pouvoir classifier un nouvel exemple, ou une nouvelle session. Le processus est récursif. Pour déterminer quel attribut tester à chaque étape, un calcul statistique, qui détermine dans quelle mesure cet attribut sépare bien les exemples Oui/Non, est utilisé. Le but est d'obtenir une classification en posant le moins de questions possibles. Il existe de nombreuses variantes pour construire des arbres de décision.

La classification à base d'arbres de décision est une méthode non paramétrique. Elle ne postule aucune hypothèse *a priori* sur la distribution des données et est résistante aux données atypiques. Le modèle de prédiction est non linéaire (cf. [123]). Par contre dans cet article, les auteurs soulignent l'inconvénient majeur de cette méthode qui est son incapacité à détecter les combinaisons de variables. Ceci est dû au principe de construction pas à pas de l'arbre. Le second reproche fait par les auteurs réside dans la nécessité de disposer d'un échantillon d'apprentissage de grande taille. Il y est souligné que « l'arbre certes peut reproduire approximativement toutes formes de

frontières, mais au prix d'une fragmentation rapide des données, avec le danger de produire des feuilles avec très peu d'individus. Corollaire à cela, les arbres sont en général instables ; les bornes de discrétisation notamment dans les parties basses de l'arbre sont entachées d'une forte variabilité. Ainsi, certains chercheurs préconisent de procéder à la discrétisation préalable des variables avant la construction de l'arbre (cf. [124]) ». Enfin un arbre de décision peut être lu et interprété directement. En effet, il est possible de le traduire en base de règles sans perte d'information.

Classification à base de règles d'association

Le classifieur à base de règles d'association utilise un ensemble de règles de type SI...ALORS pour la classification. Une règle est exprimée de la manière suivante :

SI condition ALORS conclusion

Exemple 28

SI $age = jeune$ ET $etudiant = oui$ ALORS $achat_ordinateur = oui$.

L'ensemble des conditions représente l'ensemble des attributs et l'ensemble des conclusions représente l'ensemble des classes. Sur l'exemple 28, les attributs sont *age* et *etudiant*, la classe est représentée par *achat_ordinateur*.

Les règles sont générées à partir d'un arbre de décision ou directement de la base d'apprentissage en utilisant l'algorithme d'apprentissage de règles de classifications, par exemple RIPPER introduit dans [125]. Lors de la classification à base de règles, une évaluation de la qualité des règles générées est effectuée au préalable. Un élagage des règles non pertinentes est réalisé à l'aide de mesure statistique comme FOIL (*First Order Inductive Learner*) (cf. [126]).

Les résultats sont souvent meilleurs qu'avec les arbres de décision d'après la littérature. Les inconvénients de cette méthode se situent au niveau de la taille de l'ensemble d'apprentissage. En effet, les résultats issus de cette méthode se dégradent lorsque la base d'apprentissage augmente mais aussi lorsque les données sont de mauvaise qualité (bruits).

Classification naïve bayésienne

Appartenant à la famille des classifieurs linéaires, la classification naïve bayésienne est une classification bayésienne probabiliste. Elle est basée sur le théorème de Bayes issu des travaux du révérend Thomas Bayes au $XVIII^e$ siècle. Elle prédit plusieurs hypothèses en les pondérant par leurs probabilités. Étant donné un objet O, la méthode

consiste à calculer la probabilité d'appartenance de O à chaque classe, puis choisir celle qui maximise cette valeur. Dans le cas de la classification naïve bayésienne, les attributs sont indépendants (ce qui explique l'adjectif « naïve »). Par exemple, un animal sera considéré avec un bon degré de confiance comme un canard s'il a des ailes, un bec et qu'il cancane comme un canard, en ignorant totalement la possibilité que ces caractéristiques puissent être corrélées. La caractéristique principale de la classification naïve bayésienne est qu'elle émet une hypothèse forte (indépendance des attributs) *a priori* inadaptée aux cas pratiques. Même s'il s'avère que les méthodes bayésiennes se révèlent insolubles, elles peuvent être considérées comme un point de comparaison pour mesurer la correction d'autres méthodes.

Le modèle naïf se comporte d'autant mieux que le nombre de classes parmi lesquelles choisir est faible. Ceci est dû en bonne partie au fait que le degré de certitude avec lequel il prend sa décision peut se permettre d'être médiocre tout en restant suffisant, par un effet de seuil. D'autre part, plus la répartition de la covariance des caractéristiques est distribuée équitablement entre les différentes classes, plus leurs contributions respectives à une mauvaise classification auront tendance à s'annuler dans l'estimation finale de la probabilité d'appartenance à une classe donnée : en d'autres termes, le modèle naïf est relativement peu sensible au bruit dès lors que la contribution de quelques caractéristiques à une classe « pèse » suffisamment lourd (cf. []). La classification naïve bayésienne est adaptée à des échantillons de taille réduite.

Contributions

Au vu de l'état de l'art des techniques de classifications, il a été nécessaire de trouver une technique qui permet d'avoir de bons résultats sur une base d'apprentissage importante et bruitée. Cette technique doit prendre en compte la dépendance des attributs. La classification naïve bayésienne a d'abord été retenue. Cette méthode va nous servir de référence à notre étude. Cependant elle n'est efficace que pour des échantillons de taille réduite. Une adaptation de la classification naïve bayésien au contexte de l'étude a été proposée, ainsi que des variantes. La classification naïve bayésienne et le test de χ^2 sont combinés et appliqués à la classification de sessions d'internautes. La classification naïve bayésienne à base d'*itemsets* fermés fréquents discriminants est proposée naturellement également.

Dans la suite, nous présentons la classification à base d'*itemsets* fermés fréquents discriminants. Cette approche s'inspire de la classification à base de règles d'association. Cependant elle est plus efficace car la classification à base de règles d'association nécessite de générer un nombre important de règles dans le cas des grandes bases de données. La classification à base d'*itemsets* fermés est traitée dans [127] pour classer

des données audio. Les travaux décrits dans [127] présentent de bonnes performances de classification à base d'*itemsets* fermés par rapport à la classification à base d'arbres de décision. D'ailleurs, dans [2], la classification à base d'*itemsets* fréquents fournit de meilleurs résultats que la classification à base d'arbres de décision. La classification à base d'*itemsets* fermés fréquents discriminants est réalisée sur des données de navigation sur Internet. Notre contexte d'application reste original.

3.3 Classification naïve bayésienne

Comme dit plus haut, nous avons proposé plusieurs versions de la classification naïve bayésienne, afin d'améliorer les résultats obtenus en l'adaptant à nos données. Nous avons travaillé sur une base de données constituée de logs de navigation d'internautes. Les internautes visitent des pages web sur Internet. Nous adaptons notre modèle à ce contexte qui est semblable au contexte de navigation sur eBeeHome.

Formalisme

Une session est un ensemble de sites internet visités par un utilisateur donné u_i, $i \in [1, n]$ et n est le nombre d'utilisateurs. L'ensemble des sessions de u_i, noté $\mathcal{S}_{u_i} = \{S_{i,1}, S_{i,2}, ..., S_{i,n_i}\}$ où n_i est le nombre de sessions de l'utilisateur u_i, est construit à partir des données brutes des journaux de navigation. Les sessions ont une taille fixe. Nous appelons $\mathcal{S} = \bigcup_i \mathcal{S}_{u_i}$, l'ensemble des sessions de tous les utilisateurs de la base de données. Nous appelons X_{u_i} l'ensemble des sites visités au moins une fois par l'utilisateur u_i et nous appelons $X = \bigcup_i X_{u_i}$ l'ensemble des sites visités par tous les utilisateurs. L'ordre des sites visités n'est pas pris en compte.

Exemple 29 Sur la figure 3.4, une base de données de deux utilisateurs y est présentée. L'ensemble X des sites visités par tous les utilisateurs est $\{a, b, c, d, e, f, g\}$. $\mathcal{S} = \mathcal{S}_{Marion} \cup \mathcal{S}_{Simon}$. Chaque utilisateur a 4 sessions. On écrira $abef$ pour désigner l'ensemble $\{a, b, e, f\}$. $\mathcal{S}_{Marion} = \{abef, aceg, aefg, bcfg\}$ et

$\mathcal{S}_{Simon} = \{bdeg, abcf, bdef, bcdf\}$. Dans cet exemple, la taille de la session est fixée à 4.

3.3.1 Adaptation simple de la classification naïve bayésienne

Le nombre d'utilisateurs est le nombre de classes soit n. Soit D la base d'apprentissage composée des sessions $\mathcal{S}_{u_i}$ des utilisateurs u_i. On définit une session anonyme

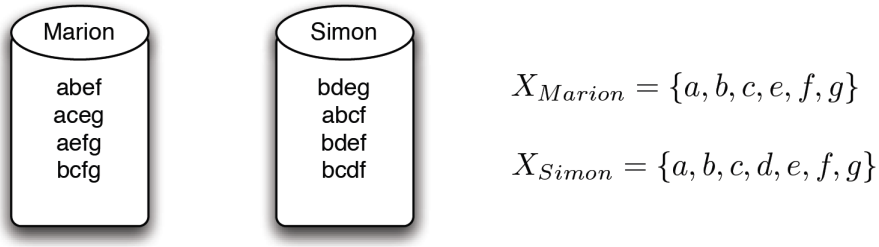


FIGURE 3.4 – Exemple de cadre d'étude

$S_{anonyme,j} = (s_1, s_2, ..., s_t)$, où t est la taille de la session et j est l'indice de la session de l'utilisateur *anonyme*, que l'on note S dans la suite pour simplifier. Le classifieur naïf bayésien prédit qu'une session appartient à un utilisateur u_i si et seulement si $P(u_i|S)$, la probabilité que l'hypothèse se réalise sachant la session anonyme S est maximisée.

$$P(u_i|S) > P(u_j|S)$$

pour

$$1 \leq j \leq n, i \neq j$$

D'après le théorème de Bayes :

$$P(u_i|S) = \frac{P(S|u_i)P(u_i)}{P(S)}$$

$P(S)$ étant constant pour toutes les classes, il est nécessaire de maximiser le terme $P(S|u_i)P(u_i)$. On suppose que $P(u_1) = P(u_2) = ... = P(u_n)$. $P(u_i)$ est une constante et est estimée :

$$P(u_i) = \frac{|u_{i,D}|}{|D|}$$

avec $|u_{i,D}|$, le nombre de sessions de l'utilisateur u_i dans la base d'apprentissage D et $|D|$, le nombre de sessions total de la base d'apprentissage D. Il est alors nécessaire de maximiser le terme $P(S|u_i)$. L'hypothèse naïve de l'indépendance conditionnelle des classes permet d'écrire :

$$P(S|u_i) = P(s_1|u_i) \times P(s_2|u_i) \times ... \times P(s_t|u_i)$$

On estime les probabilités $P(s_1|u_i), P(s_2|u_i), ..., P(s_t|u_i)$ à partir des données de la base d'apprentissage. s_k, avec $k \in [1, t]$ est un site visité. Comme s_k n'est pas une

donnée qui a une valeur continue, $P(s_k|u_i)$ est défini comme le nombre de sessions de l'utilisateur u_i contenant le site visité s_k divisé par le nombre de sessions total de l'utilisateur u_i.
Finalement, pour prédire l'utilisateur à qui appartient une session anonyme S, $P(S|u_i)$ est calculée pour chaque utilisateur. L'utilisateur prédit est celui pour lequel le terme $P(S|u_i)$ est maximisé.

Correctif de Laplace

Si un site s_k n'est pas connu dans la base d'apprentissage alors $P(s_k|u_i)$ est estimée en utilisant le correctif de Laplace. En utilisant le correctif de Laplace, on évite d'avoir des probabilités nulles. L'idée derrière ce correctif est simple. On suppose que la base d'apprentissage est tellement grande qu'en ajoutant 1 au nombre de sessions de l'utilisateur et au nombre de sessions total de l'utilisateur, la différence des probabilités estimées est négligeable.

> **Exemple 30** Supposons qu'on ait trois sites dans la base d'apprentissage (*msn.fr* avec un support de 800, *facebook.com* avec un support de 50 et *seneweb.com* avec un support de 150) et que dans la base de test, il y a *univ-bp.fr*, en utilisant le correctif de Laplace on ajoute une instance à chaque support. On aura alors *msn.fr* avec un support de 801, *facebook.com* avec un support de 51 et *seneweb.com* avec un support de 151 et *univ-bp.fr* avec un support de 1. Le site *univ-bp.fr* aura une probabilité de ($\frac{1}{1004} = 0.001$).

3.3.2 Classification naïve bayésienne à base d'*itemsets* fermés fréquents discriminants

Dans cette version, nous appelons un pattern, *itemset* fermé fréquent discriminant. Un pattern est un ensemble de sites visités non ordonnés de taille variable. Soit D_p la base d'apprentissage composée des patterns $P_{i,j}$ des utilisateurs u_i, avec $j \in [1, h]$ et h est le nombre de pattern de l'utilisateur. Un utilisateur peut avoir plusieurs patterns le décrivant.

On rappelle que $P(X)$ est constant pour toutes les classes, ainsi on suppose que $P(u_1) = P(u_2) = ... = P(u_n)$. $P(u_i)$ est une constante et est estimée :

$$P(u_i) = \frac{|u_{i,D_p}|}{|D_p|}$$

avec $|u_{i,D_p}|$, le nombre de patterns de l'utilisateur u_i dans la base d'apprentissage D_p et $|D_p|$, le nombre de patterns total de la base d'apprentissage D_p.

Le classifieur naïf bayésien prédit alors qu'une session anonyme S appartient à un utilisateur u_i si et seulement si $P(s|u_i)$ est maximisée.

$$P(s|u_i) = P((s_1, s_2, ..., s_t)|u_i)$$

On propose d'estimer la probabilité $P((s_1, s_2, ..., s_t)|u_i)$ à partir des données de la base d'apprentissage, de différentes manières. On présente dans la suite les variantes de la classification naïve bayésienne proposées.

Version 1

Décrivons la première version proposée. $P((s_1, s_2, ..., s_t)|u_i)$ est défini comme le nombre de patterns de l'utilisateur u_i contenant $(s_1, s_2, ..., s_t)$ divisé par le nombre de patterns total de l'utilisateur u_i. Le nombre de patterns de l'utilisateur u_i contenant $(s_1, s_2, ..., s_t)$ est estimé par :

$$\sum_{j=1}^{J} \frac{|S \cap P_{i,j}|}{min(t, |P_{i,j}|)}$$

avec $J = |u_{i,D_p}|$, le nombre de patterns de l'utilisateur u_i dans la base d'apprentissage D_p, t la taille de la session S, $|P_{i,j}|$ la taille du pattern j de l'utilisateur u_i.

Montrons que le calcul d'estimation proposé permet de vérifier que la somme des probabilités est bien égale à 1 :

$$\sum_{i=1}^{n} P((s_1, s_2, ..., s_t)|u_i) = 1$$

Soit :

$$\sum_{i=1}^{n} \frac{\sum_{j=1}^{J} \frac{|S \cap P_{i,j}|}{min(t,|P_{i,j}|)}}{|u_{i,D_p}|} = 1$$

Or

$$\sum_{i=1}^{n} \frac{\sum_{j=1}^{J} \frac{|S \cap P_{i,j}|}{min(t,|P_{i,j}|)}}{|u_{i,D_p}|} \leq \sum_{i=1}^{n} \frac{J}{|u_{i,D_p}|} = n$$

Supposons que :

$$\sum_{i=1}^{m} \frac{\sum_{j=1}^{J} \frac{|S \cap P_{i,j}|}{min(t,|P_{i,j}|)}}{|u_{i,D_p}|} = 1$$

Alors :

$$\frac{\sum_{j=1}^{J} \frac{|S \cap P_{i,j}|}{min(t,|P_{i,j}|)}}{|u_{i,D_p}|} \leq 1$$

$$\sum_{j=1}^{J} \frac{|S \cap P_{i,j}|}{min(t,|P_{i,j}|)} \leq |u_{i,D_p}|$$

$$\frac{|S \cap P_{i,j}|}{min(S,|P_{i,j}|)} \leq |u_{i,D_p}|$$

$$|S \cap P_{i,j}| \leq |u_{i,D_p}| * min(t,|P_{i,j}|)$$

Version 2

Rappelons que $P((s_1, s_2, ..., s_t)|u_i)$ est défini comme le nombre de patterns de l'utilisateur u_i contenu dans $(s_1, s_2, ..., s_t)$ divisé par le nombre de patterns total de l'utilisateur u_i. Le nombre de patterns de l'utilisateur u_i contenu dans $(s_1, s_2, ..., s_t)$ est un support estimé pour chaque utilisateur u_i par :

$$Supp_i = \sum_{j=1}^{J} Supp_{i,j}$$

tel que si

$$|S \cap P_{i,j}| = |P_{i,j}|$$

alors

$$Supp_{i,j} = 1$$

sinon

$$Supp_{i,j} = 0$$

avec $J = |u_{i,D_p}|$, le nombre de patterns de l'utilisateur u_i dans la base d'apprentissage D_p, t le nombre de sites visités de la session S, $|P_{i,j}|$ la taille du pattern j de l'utilisateur u_i.

Version 3

Pour améliorer les résultats des versions précédentes, un système de filtre inspiré de [128] est proposé. L'idée est de calculer, pour une session test, l'ensemble des probabilités pour chaque utilisateur. Le filtre ne garde que les utilisateurs qui ont une probabilité supérieure à une certaine valeur. Ensuite on reclasse (on recalcule les probabilités) la session test en question parmi les utilisateurs filtrés. Il est possible de répéter le processus jusqu'à l'obtention d'un seul utilisateur filtré qui sera l'utilisateur à qui appartient la session test en question. Ce processus itératif a pour objectif d'améliorer les résultats mais il demande un temps de calcul important.

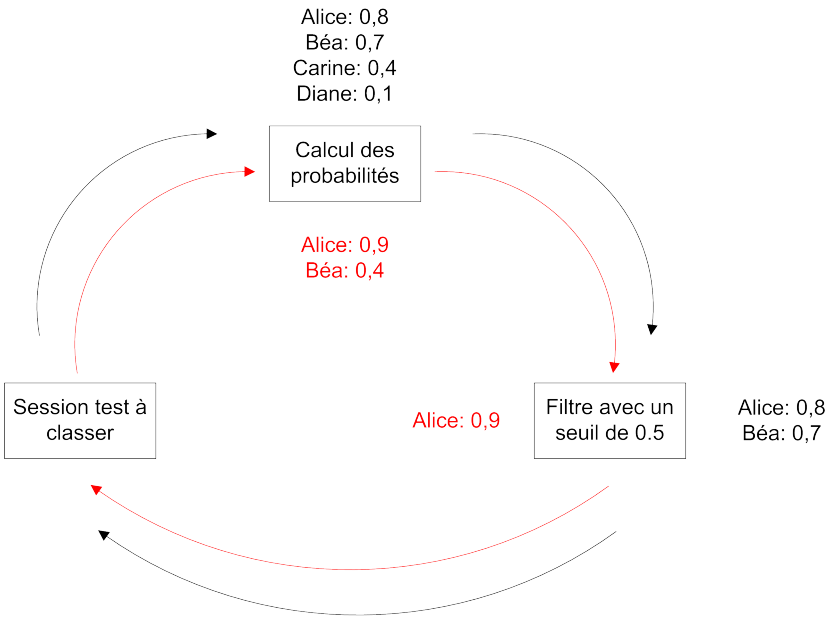


FIGURE 3.5 – Le processus itératif du modèle

Exemple 31 Sur la figure 3.5, les flèches noires schématisent la première itération. La session test est à classer parmi les utilisateurs de la base d'apprentissage à savoir Alice, Béa, Carine et Diane. Avec le modèle naif bayésien, le calcul de probabilité est effectué pour chaque classe. Rappelons que les classes sont les utilisateurs. Ensuite, en fonction d'un seuil de 0,5, un filtre des utilisateurs permet de garder Alice et Béa. La session test est à classer de nouveau mais cette fois parmi Alice et Béa. La deuxième itération est représentée par les flèches rouges. On répète le processus

jusqu'à l'obtention d'une classe. La session test appartient donc à la dernière classe restante.

Version 4

Dans cette version, une sélection des motifs discriminants est d'abord réalisée avec le test de χ^2. Les descripteurs caractéristiques de la base d'apprentissage sont ensuite utilisés dans la classification naïve bayésienne. A partir d'un code en Java et disponible sur Github [129], nous avons adapté la méthode de classification de texte par Bayes à notre étude. La méthode consiste à utiliser la classification naïve bayésienne sur une base d'apprentissage épurée à l'aide d'algorithmes de sélection. Le processus de sélection est mis en place avant le processus d'apprentissage. La sélection par le test de χ^2 permet de réduire la quantité de données d'apprentissage, et améliore la vitesse de la phase d'apprentissage. La base d'apprentissage, constituée de 2/3 des données est réduit à une base constituée que de descripteurs qui satisfont le test d'indépendance de χ^2. Cet ensemble réduit est utilisé dans l'algorithme de classification naïve bayésienne.

3.4 Classification à base d'*itemsets* fermés fréquents discriminants

Pour utiliser la classification à base d'*itemsets* fermés fréquents discriminants, il est nécessaire de suivre plusieurs étapes. L'extraction des *itemsets* fermés fréquents est la première étape. Dans le chapitre 2, nous avons expliqué le processus d'extraction des *itemsets* fermés fréquents en utilisant notamment Charm (cf. [80]). Ensuite, il est nécessaire de sélectionner les meilleurs descripteurs discriminants de la base d'apprentissage. Dans cette étude, nous proposons trois nouvelles heuristiques de sélection des descripteurs adéquats à notre problématique. A partir des descripteurs, des profils utilisateurs sont construits. Chaque classe est caractérisée par un ensemble d'*itemsets* fermés fréquents discriminants. Pour la phase de classification, nous avons utilisé des mesures de distance afin de calculer la distance entre les profils des utilisateurs et un profil anonyme. Notons qu'un profil anonyme est construit à partir de sessions de la base de test. Enfin le score obtenu du calcul de distance permet de classer un profil anonyme. Dans cette section, nous présentons les étapes du modèle de classification proposé.

Formalisme

Rappelons que $\mathcal{S} = \bigcup_i \mathcal{S}_{u_i}$ est l'ensemble des sessions de tous les utilisateurs de la base de données. X_{u_i} est l'ensemble des sites visités au moins une fois par l'utilisateur u_i et $X = \bigcup_i X_{u_i}$ est l'ensemble des sites visités par tous les utilisateurs.

Définition 3.4.1. ***k-pattern***
Soient X un ensemble de sites visités et $\mathcal{S}$ un ensemble de sessions de X. Un sous-ensemble P de X est appelé un k-pattern, où k correspond à la taille de P. Une session S dans $\mathcal{S}$ contient un k-pattern si $P \subseteq S$.

Nous définissons le support d'un pattern P comme le pourcentage de sessions dans S qui contiennent P (par extension, nous donnons le support d'un pattern dans l'ensemble des sessions d'un utilisateur donné u_i) :

$$support_{\mathcal{S}}(P) = \frac{||\{S \in \mathcal{S} \mid P \subseteq S\}||}{||\mathcal{S}||}$$

$$support_{\mathcal{S}_{u_i}}(P) = \frac{||\{S \in \mathcal{S}_{u_i} \mid P \subseteq S\}||}{||\mathcal{S}_{u_i}||}$$

Pour un utilisateur donné, la force relative d'un pattern est équivalente au lift [4] dans le contexte des règles d'association (c'est-à-dire, le support du pattern propre à cet utilisateur divisé par le support du pattern sur tous les utilisateurs). Plus formellement :

$$lift_{\mathcal{S}_{u_i}\mathcal{S}}(P) = \frac{support_{\mathcal{S}_{u_i}}(P)}{support_{\mathcal{S}}(P)}$$

Le *support* mesure la force d'un pattern dans une description comportementale d'un utilisateur donné. Le *support relatif* atténue la mesure du support en considérant le support du pattern sur l'ensemble des sessions. Plus le support global du pattern est élevé, moins il est caractéristique d'un utilisateur.

Exemple 32 Soit $P_1 = \{abe\}$ et $P_2 = \{af\}$ deux patterns. Nous avons :

$$support_{\mathcal{S}}(P_1) = 1/8 \qquad support_{\mathcal{S}}(P_2) = 3/8$$

$$support_{\mathcal{S}_{Marion}}(P_1) = 1/4 \qquad support_{\mathcal{S}_{Marion}}(P_2) = 1/2$$

4. Le lift mesure l'intérêt d'une règle d'association, soit la dépendance statistique de deux ensembles (cf. [130]).

$$lift_{\mathcal{S}_{Marion}\mathcal{S}}(P_1) = (1/4)/(1/8) = 2$$
$$lift_{\mathcal{S}_{Marion}\mathcal{S}}(P_2) = (1/2)/(3/8) = 4/3$$

Nous utilisons aussi dans la suite de ce chapitre le $tf \times idf$, une statistique numérique introduite dans le chapitre 2. La valeur du $tf \times idf$ augmente proportionnellement au nombre de fois que le mot apparaît dans le document, mais il est compensé par la fréquence d'apparition de ce mot dans tout le texte (cf. [113]). Dans notre contexte, un mot devient un pattern, un document devient un ensemble de sessions $\mathcal{S}_{u_i}$ d'un utilisateur donné et la structure devient l'ensemble global $\mathcal{S}$ de toutes les sessions.

Définition 3.4.2. $tf \times idf$

Soient P un pattern et U un ensemble d'utilisateurs avec $U_P \subseteq U$ tel que $\forall u_i \in U_P$, $support_{\mathcal{S}_{u_i}}(P) \neq 0$. Soit $\mathcal{S}_{u_i}$ un ensemble de sessions d'un utilisateur donné u_i et $\mathcal{S}$ l'ensemble de toutes les sessions. Le *normalized term frequency*, noté $tf(P)$, est égal au $support_{\mathcal{S}_{u_i}}(P)$ et l'*inverse document frequency*, noté $idf(P)$, est égal au $\log(||U||/||U_P||)$. Nous avons :

$$tf \times idf(P) = support_{\mathcal{S}_{u_i}}(P) \times \log\Big(\frac{||U||}{||U_P||}\Big)$$

Exemple 33 Considérons l'ensemble des sessions de Marion et de Simon. Nous avons deux utilisateurs, alors $||U|| = 2$. Soit le patterns $P_1 = aeg$ et $P_2 = bf$, alors $||U_{aeg}|| = 1$ car le pattern aeg apparait seulement dans les sessions de Marion et $||U_{bf}|| = 2$ car le pattern apparait dans les deux ensembles de sessions. Nous avons :

- $tf \times idf_{Mar}(aeg) = support_{S_{Mar}}(aeg) \times log\ 2 = 1/2$
- $tf \times idf_{Mar}(bf) = support_{S_{Mar}}(bf) \times log\ 1 = 0$

Un concept de contexte formel $\mathbb{K} = (S, X, I)$ où S est un ensemble d'objets, X un ensemble d'attributs et I une relation entre S et X, est établi.

Exemple 34 Sur la figure 3.6, le treillis des patterns fermés construit à partir de l'ensemble des sessions de Marion est représenté. Chaque ensemble est un ensemble fermé. L'ensemble aeg est un 3-pattern fermé et f un 1-pattern fermé.

Dans la suite de l'étude, nous allons nous concentrer sur trois points essentiels afin de mettre en place un système d'authentification implicite :

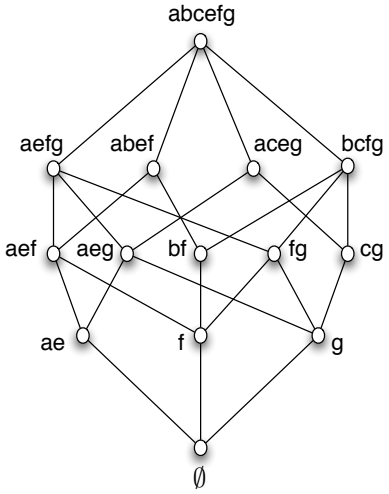


FIGURE 3.6 – Treillis des fermés

1. *la sélection des descripteurs* : un ensemble de descripteurs est calculé pour chaque utilisateur. Dans la suite de l'étude, un pattern est un *itemset* fermé fréquent de taille variable.

2. *le calcul du profil utilisateur* : un vecteur de profil commun est défini et nous calculons ses composantes pour chaque utilisateur.

3. *l'étape d'identification* : l'identification d'un profil anonyme est possible grâce à ces profils et l'utilisation de mesures de distance. Nous calculons la précision du modèle proposé en utilisant la matrice de confusion construite au préalable.

3.4.1 Sélection des patterns propres

L'étape de sélection des meilleurs descripteurs pour la classification est abordée dans cette section. En effet, la première et la plus importante étape de notre modèle, appelée aussi *sélection des patterns propres*, est de calculer un ensemble de patterns propres pour chaque utilisateur u_i. Cet ensemble de patterns est noté $\mathcal{P}_{u_i} = \{P_{i,1}, P_{i,2}, ..., P_{i,p}\}$. Notre étude s'appuie sur une étude menée en 2010 et décrite dans [2]. L'auteur estime que $p = 10$ est la valeur à partir de laquelle les résultats restent stables. Nous choisissons de suivre cette recommandation. Dans [2], les 10 plus fréquents 1-patterns sont sélectionnés pour chaque utilisateur. L'objectif de notre étude est de montrer qu'il pourrait être beaucoup plus efficace de sélectionner des k-patterns fermés. Cependant, le nombre de patterns fermés peut être très élevé, il est donc nécessaire de les

sélectionner, au travers de trois heuristiques (H_1, H_2 et H_3). Pour chaque heuristique, les patterns fermés sont calculés grâce à l'algorithme Charm (cf. [80]) provenant de la plate-forme Coron (cf. [131]). Seuls les patterns fermés de taille inférieure ou égale à 7 sont sélectionnés. Au sein des heuristiques H_2 et H_3, les k-patterns fermés sont mis à l'écart des sites internet n'étant pas caractéristiques de l'utilisateur (via un filtrage par $tf \times idf$). De cette façon, nous appelons *pattern fermé filtré* un pattern fermé restreint au sites présents dans la liste des 40 sites ayant une valeur de $tf \times idf$ maximale pour un utilisateur donné. Les trois heuristiques et le modèle décrit dans [2] sont décrits ici. p est le nombre de patterns fermés sélectionnés.

1. H_1 : p k-patterns fermés ayant un $tf \times idf$ maximal ;
2. H_2 : p k-patterns fermés filtrés avec un support maximal et maximums par inclusion d'ensembles ;
3. H_3 : p k-patterns fermés filtrés avec un $tf \times idf$ maximal et minimums par inclusion d'ensembles ;
4. p 1-pattern avec un support maximal (cf. [2]).

L'heuristique H_1 est un processus simple basé sur la mesure du $tf \times idf$pour sélectionner les meilleurs patterns propres (cf. algorithme 1). Les processus H_2 et H_3 nous permettent de comparer les mesures de $tf \times idf$ et de support. Les algorithmes 2 et 3 décrivent les processus de H_2 et H_3 qui sont très similaires. Notons que la liste des patterns propres sélectionnés par ces trois heuristiques dépend de l'ordre de la liste des patterns fermés en entrée, même si cette liste est triée en fonction des valeurs de support ou de $tf \times idf$.

Algorithme 1 : H_1 : p $k-patterns$ fermés de $tf \times idf$ maximal.

Entrées : $\mathcal{C}_{u_i}$: l'ensemble des *itemsets* fermés de Charm de l'utilisateur u_i ;
p : le nombre de patterns retournés ;
Résultat : $\mathcal{P}_{u_i}$: l'ensemble des patterns propres de l'utilisateur u_i ;

```
1  début
2      Calculer le tf × idf pour chaque pattern de Charm ;
3      Trier les patterns par ordre décroissant de valeurs de tf × idf ;
4      Retourner les p premiers patterns ;
```

Algorithme 2 : H_2 : p k-patterns fermés filtrés avec un support maximal et maximums par inclusion d'ensembles

Entrées :
$\mathcal{C}_{u_i}$: l'ensemble des itemsets fermés de Charm de l'utilisateur u_i ;
(par ordre décroissant du support) ;
p : le nombre de patterns retournés.
Résultat : $\mathcal{P}_{u_i}$: l'ensemble des p patterns propres de l'utilisateur u_i ;

```
1  début
2      P_{u_i} ← ∅ ;
3      tant que (|P_{u_i}| < p) faire
4          Sélectionner C ∈ C_{u_i} ;
5          si (∀P ∈ P_{u_i}, C ⊈ P) alors
6              pour (P' ∈ P_{u_i}) faire
7                  si (P' ⊂ C) alors
8                      P_{u_i} ← P_{u_i} \ {P'} ;
9              P_{u_i} ← P_{u_i} ∪ C ;
10     Retourner P_{u_i} ;
```

Algorithme 3 : H_3 : p $k-patterns$ fermés filtrés avec un $tf \times idf$ maximal et minimums par inclusion ensembliste.

Entrées :
$\mathcal{C}_{u_i}$: l'ensemble des itemsets fermés de Charm de l'utilisateur u_i ;
(par ordre décroissant du $tf \times idf$) ;
p : le nombre de patterns retournés.
Résultat : $\mathcal{P}_{u_i}$: l'ensemble des p patterns propres de l'utilisateur u_i ;

```
1  début
2      P_{u_i} ← ∅ ;
3      tant que (|P_{u_i}| < p) faire
4          Sélectionner C ∈ C_{u_i} ;
5          si (∀P ∈ P_{u_i}, P ⊈ C) alors
6              pour (P' ∈ P_{u_i}) faire
7                  si (C ⊂ P') alors
8                      P_{u_i} ← P_{u_i} \ {P'} ;
9              P_{u_i} ← P_{u_i} ∪ C ;
10     Retourner P_{u_i} ;
```

Exemple 35 Par exemple, fixons la valeur de p à 4. Attribuons à Marion ces patterns propres à l'aide des quatre approches énoncées précédemment. Les patterns fermés de Marion sont présentés sur la figure 3.6. Nous avons alors :

1. H_1 : 4 k-patterns fermés ayant un $tf \times idf$ maximal. On peut donner les premiers patterns fermés et leurs valeurs de $tf \times idf$ correspondant :
   $(ae, 3/4), (aeg, 1/2), (aef, 1/2), (fg, 1/2), (cg, 1/2), ...$ Comme $p = 4$, On peut choisir $\mathcal{P}_{Marion} = \{ae, aef, aeg, fg\}$.
2. H_2 : 4 k-patterns fermés filtrés avec un support maximal et maximums par inclusion d'ensembles. L'exécution de l'algorithme 2 permet d'avoir $\mathcal{P}_{Marion} = \{aef, bf, fg, cg\}$.
3. H_3 : 4 k-patterns fermés filtrés avec un $tf \times idf$ maximal et minimums par inclusion d'ensembles. L'algorithme 3 permet d'avoir seulement 3 $k - patterns$. $\mathcal{P}_{Marion} = \{a, f, g\}$. En effet, les autres $k - patterns$ fermés ne sont pas minimum par l'inclusion.
4. 4 1-pattern avec un support maximal (cf. [2]). Les patterns propres appartiennent à l'ensemble $\{a, b, c, e, f, g\}$. La fonction de support est maximal pour chaque P dans $\{a, e, f, g\}$. Alors on peut dire $\mathcal{P}_{Marion} = \{a, e, f, g\}$.

Pour résumer, sur la figure 3.7, nous situons le processus de sélection des meilleurs patterns propres par les algorithmes H_1, H_2 ou H_3 après l'extraction des *itemsets* fermés fréquents par Charm.

FIGURE 3.7 – Sélection des patterns propres

3.4.2 Calcul des profils utilisateurs

Nous définissons et nous notons $\mathcal{P}_{all} = \bigcup_i \mathcal{P}_{u_i}$ l'ensemble des patterns propres de tous les utilisateurs. L'ensemble $\mathcal{P}_{all}$ nous permet de définir un espace commun au

sein duquel chaque utilisateur peut être plongé. Plus formellement, $\mathcal{P}_{all}$ définit un espace vectoriel $\mathcal{V}$ de taille $all = ||\mathcal{P}_{all}||$ où un utilisateur donné u_i est représenté comme un vecteur $V_{u_i} = \langle m_{i,1}, m_{i,2}, ..., m_{i,all} \rangle$. La figure 3.8 représente ce vecteur.

FIGURE 3.8 – L'ensemble des patterns propres de tous les utilisateurs $\mathcal{P}_{all} = \bigcup_i \mathcal{P}_{u_i}$ définit un vecteur $\mathcal{V}$.

La seconde étape de notre modèle, appelée *calcul du profil utilisateur*, consiste à calculer, pour chaque utilisateur u_i, une valeur numérique pour chaque composante $m_{i,j}$ du vecteur V_{u_i}, où i est l'utilisateur courant, $j \in [1, all]$ est un pattern de cet utilisateur et m représente une mesure donnée. Nous comparons deux mesures proposées dans [2] : le *support* et le *lift*.

$$m_{i,j} = support_{\mathcal{S}_{u_i}}(P_j)$$

$$m_{i,j} = lift_{\mathcal{S}_{u_i}\mathcal{S}}(P_j)$$

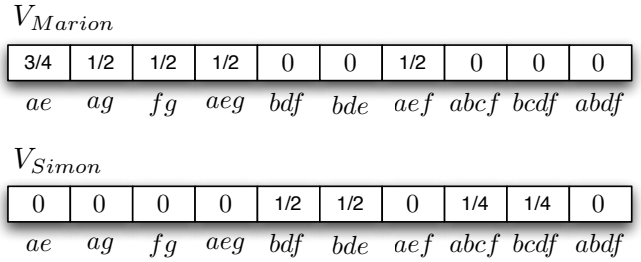


FIGURE 3.9 – Les vecteurs de profils de Marion et Simon avec la mesure du support.

Exemple 36 Supposons que l'heuristique H_1 est utilisée pour extraire les patterns propres de Marion, Simon et Jeanne. $\mathcal{P}_{Marion} = \{ae, aef, aeg, fg\}$, $\mathcal{P}_{Simon} = \{bdf, bde, abcf, bcde\}$ et $\mathcal{P}_{Jeanne} = \{ag, aef, bde, abdf\}$. La taille du vecteur $\mathcal{V}$ est égale à 10 parce que Jeanne

partage 2 patterns propres (aef et bde) avec Marion et Simon. Supposons qu'on utilise la mesure du support. Sur la figure 3.9 les vecteurs de profils de Marion et Simon y sont représentés.

La phase d'authentification consiste à déterminer l'utilisateur correspondant à un ensemble anonyme de sessions. À partir de cet ensemble de sessions, nous devons construire un profil de test et trouver le profil utilisateur (définit lors de la phase d'apprentissage) le plus proche.

3.4.3 Mesures de distance

Ici la mesure de distance permet de comparer un profil utilisateur issu de l'apprentissage avec un profil anonyme construit à partir des sessions de la base de test. A l'aide de cette dernière étape, on obtient un score. La classification du profil anonyme est effectuée par la suite.

Soit $V_{u_i} = \langle m_{i,1}, ..., m_{i,all} \rangle$ et $V_{u_t} = \langle m_{t,1}, ..., m_{t,all} \rangle$, deux vecteurs de profils. Notons $Dis_{Euclidien}(V_{u_i}, V_{u_t})$ et $Sim_{Cosinus}(V_{u_i}, V_{u_t})$ la distance Euclidienne et la similarité par cosinus entre deux vecteurs V_{u_i} et V_{u_t} respectivement :

$$Dis_{Euclidien}(V_{u_i}, V_{u_t}) = \sqrt{\sum_j (m_{t,j} - m_{i,j})^2}$$

$$Sim_{Cosinus}(V_{u_i}, V_{u_t}) = \frac{\sum_j (m_{t,j} \times m_{i,j})}{\sqrt{\sum_j (m_{t,j})^2 \times \sum_j (m_{i,j})^2}}$$

Nous testons aussi les performances de deux autres mesures de similarité. La mesure de Kulczynski introduite en 1927 (cf.[]) et la similarité de Dice introduite par L.R Dice en 1945 et T. Sorensen en 1948 (cf.[**?**]) sont deux mesures statistiques. Elles sont en général utilisées pour comparer deux vecteurs dans $\mathcal{R}^n$.

$$Sim_{Kulczynski}(V_{u_i}, Vu_t) = \frac{\sum_j \min(m_{t,j}, m_{i,j})}{\sum_j |m_{t,j} - m_{i,j}|}$$

$$Sim_{Dice}(V_{u_i}, Vu_t) = \frac{2 \times \sum_j (m_{t,j} \times m_{i,j})}{\sum_j (m_{t,j})^2 + \sum_j (m_{i,j})^2}$$

3.5 Résumé

Dans ce chapitre, nous présentons notre contribution majeure. Un bref état de l'art sur les techniques de classification qui pourraient être adéquates à notre base de données est présenté. Nous nous sommes limités à trois techniques principalement : les arbres de décision, la classification à base de règles d'association et la classification naïve bayésienne. Une description des méthodes a été proposée. La classification naïve bayésienne est notre modèle de référence du fait de ces performances populaires. Nous avons proposé d'estimer les probabilités calculées dans le modèle naïf bayésien à base d'*itemsets* fermés fréquents par deux méthodes présentées dans la version 1 et la version 2. Dans les versions 3 et 4, nous proposons d'améliorer le classifieur naif bayésien à base d'*itemsets* fermés fréquents en utilisant des méthodes de filtrage comme la sélection des meilleurs descripteurs avec le test χ^2.

La classification à base d'*itemsets* fermés fréquents discriminants est avantageuse car basée sur l'essentiel de l'information. Après l'extraction des *itemsets* fermés fréquents, une sélection des *itemsets* fermés fréquents discriminants est effectuée. Des profils sont construits pour chaque utilisateur qui représente une classe du processus de classification. Une classe est composée d'un ensemble d'*itemsets* fermés fréquents discriminants caractérisant l'utilisateur associé à la classe. Pendant l'étape de classification, un profil anonyme est comparé aux différents profils construits à partir de la base d'apprentissage grâce aux mesures de distance telles que la similarité par cosinus, la similarité de Dice et de Kulczynski. Nous avons aussi utilisé la distance euclidienne et l'avons présentée dans ce manuscrit bien que les dimensions de l'espace de l'étude ne soient pas indépendants. Nous pensons que les résultats sont intéressants à analyser.

Des questions peuvent se poser sur la façon de construire le profil de l'utilisateur. En effet, nous utilisons le support et le lift pour remplir le vecteur de profil. Cette technique est essentiellement inspirée de [2]. Il est possible de construire le vecteur de profil en utilisant d'autres types de mesure.

Il existe bien sûr quelques méthodes qui ont la réputation d'avoir de bonnes performances sur des données comme les nôtres. Les forêts d'arbres décisionnels introduits en 2001 dans [132] permettent de meilleurs performances. L'algorithme des forêts d'arbres décisionnels effectue un apprentissage sur de multiples arbres de décision entraînés sur des sous-ensembles de données légèrement différents.

Chapitre 4

Contexte applicatif : navigation sur Internet

Sommaire

2.1 Introduction 70

2.2 Définitions des notions de base 72

2.3 *Itemsets* fermés fréquents 75

2.3.1 Formalisation 75

2.3.2 Extraction d'*itemsets* fermés fréquents 81

2.4 Outils de discrimination 85

2.4.1 Outils statistiques 87

2.4.2 Stabilité des *itemsets* fermés fréquents 88

2.5 Résumé et discussion 89

L'homme et sa sécurité doivent
constituer la première préoccupation
de toute aventure technologique.

Albert Einstein, (1879 - 1955).

4.1 Introduction

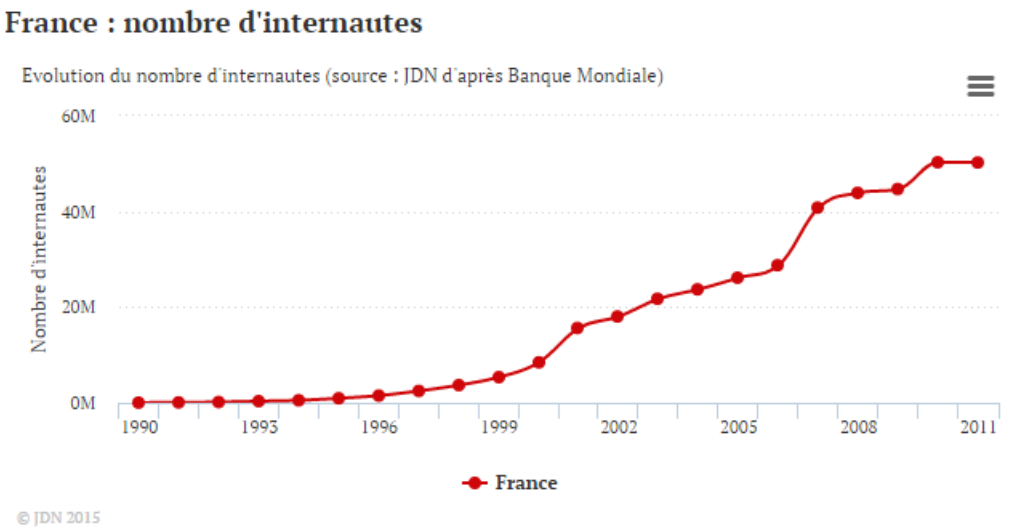
Cette thèse est un contrat CIFRE (Conventions Industrielles de Formation par la Recherche) entre l'entreprise Almerys [1] et le LIMOS (Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes). Cette collaboration devrait fournir à l'entreprise des outils d'analyse du comportement des utilisateurs de leurs services réunis au sein d'un espace numérique de vie, eBeeHome. Nous n'avons malheureusement pas eu les données issues de cet espace du fait des évolutions de la plateforme qui n'ont pas permis à l'entreprise de colleccter des données nécessaires. Pour cette raison, nous avons travaillé sur des données de navigation sur Internet d'étudiants de l'université Blaise Pascal de Clermont-Ferrand, en accord avec Almerys. Toutes les méthodologies présentées dans les chapitres précédents sont appliquées aux données de navigation d'étudiants. Ce chapitre montre le contexte de la navigation sur Internet, ainsi que les données associées.

L'université Blaise Pascal, composée de 16036 étudiants, 1166 enseignants et 908 personnels administratifs en 2014-2015 [2], fournit un accès internet par un moyen filaire ou sans fil. Les données recueillies sont l'ensemble des pages visitées sur Internet par les universitaires qui passent par le réseau de l'université. En effet, pour accéder au réseau de l'université, il est nécessaire d'avoir un compte accessible par une authentification faible (login et mot de passe). L'authentification implicite peut être adaptée à ce contexte également. Il est possible de faire appel à l'authentification implicite lorsque le système détecte un vol ou un piratage de compte.

Internet est devenu le plus grand réseau informatique accessible au public. Ce réseau de réseaux, sans centre névralgique, est composé de millions de réseaux aussi bien publics que privés, universitaires, commerciaux et gouvernementaux. L'information est transmise par Internet grâce à un ensemble standardisé de protocoles de transfert de données, qui permet l'élaboration d'applications et de services variés comme le

1. Almerys est un acteur majeur dans le domaine du traitement industriel des données numériques et est un opérateur de tiers-payant santé. https ://www.almerys.com, accès le 19/10/2015.
2. http ://www.univ-bpclermont.fr/, accès 24/10/2015

courrier électronique, la messagerie instantanée, le pair-à-pair et le World Wide Web [3]. Un internaute est un utilisateur d'Internet. En France, le nombre d'internautes a triplé de 2000 à 2011. En 2011, près de 77% de la population sont des internautes [4].



	France	Rang Europe	Rang monde
Nombre d'internautes (2011)	50 235 586	n°4 sur 44	n°9 sur 195
Part des internautes dans la population	76,77 %	n°15 sur 44	n°24 sur 195

FIGURE 4.1 – Évolution du nombre d'internautes en France

Avec des millions de nouvelles pages web publiées chaque jour, l'utilisation du web explose ces dernières années. De très grandes bases de données apparaissent. L'application des technique de fouille de données au web, appelée fouille de données web ou en anglais *web data mining*, est devenu naturellement un domaine de recherche prisé. Un sous domaine de la fouille de données web est la fouille de comportements utilisateurs ou en anglais *web usage mining (WUM)*. On dira WUM pour *web usage*

3. L'internet ayant été popularisé par l'apparition du World Wide Web (WWW), les deux sont parfois confondus par le public non averti. Le World Wide Web n'est pourtant que l'une des applications de l'Internet.

4. http ://www.journaldunet.com/, accès le 01/11/2015

mining dans la suite. L'extraction de connaissance à partir de données logs HTTP permet la construction des patrons d'accès web afin d'améliorer ou de personnaliser un site web. Le processus WUM est composé du pré-traitement, de la fouille de données et de l'analyse des motifs extraits. Dans ce chapitre, nous nous concentrons sur la première étape à savoir le nettoyage des données.

Afin de mettre en place un système d'authentification implicite, les méthodes développées dans cette thèse permettent de répondre à la question suivante.

Peut-on trouver un comportement caractéristique d'un internaute ?

Les données sont préparées afin de répondre à cette question.

- Les données présentent beaucoup de bruits dûs aux pages publicitaires intrusives et non-sollicitées, en particulier. Il est nécessaire de les nettoyer.
- Il n'est pas possible de repérer le début d'une session dans les données. Pour extraire des *itemsets* fermés fréquents, il est nécessaire de construire l'équivalent d'une base de transactions, une base de données de sessions.

Dans ce chapitre, nous décrivons nos données dans la section 4.2, ensuite présentons un état de l'art sur la fouille de données de navigation dans la section 4.3. Le nettoyage des données est traité dans 4.4. Enfin dans la dernière section, nous résumons le chapitre.

4.2 Description des données

Les données sont enregistrées au fur et à mesure des actions sur un serveur. Un traitement en amont est effectué afin de préserver l'anonymat des internautes. Nous récupérons les données ensuite sous le format CSV [5]. Dans un fichier, nous avons l'ensemble des requêtes HTTP représentant les activités de la journée de tous les utilisateurs qui ont utilisé le réseau universitaire. Une requête pour une ressource web est faite par un client à un serveur web. Une ressource web est une ressource accessible par une version du protocole HTTP ou un protocole similaire. Un serveur web est un serveur qui donne accès à des ressources web. Les requêtes sont ordonnées chronologiquement. Une journée commence en générale à 8h et se finit à 00h. Nos fichiers contiennent en moyenne 130000 actions (nombre de lignes du fichier). Nous

5. Comma-Separated Values est un format représentant des données tabulaires sous forme de valeurs séparées par des virgules.

avons quatre colonnes dans nos fichiers : l'adresse IP, un identifiant utilisateur, la date et l'heure ou l'horodatage, et la page web visitée.

Adresse IP

L'adresse IP (Internet Protocol) est un numéro d'identification qui est attribué de façon permanente ou provisoire à chaque appareil connecté à un réseau informatique utilisant l'Internet Protocol. L'adresse IP est à la base du système d'acheminement (le routage) des messages sur Internet.Dans le cas du réseau universitaire, les adresses IP sont souvent les mêmes. Il n'est donc pas intéressant de les conserver dans cette étude.

Identifiant utilisateur

Un identifiant qui est le hash [6] du login remplace le login. En effet, pour des raisons de confidentialité, il est nécessaire d'utiliser une fonction de hachage irréversible.

Horodatage

La date et l'heure de visite d'une page web sont aussi disponibles sur le fichier. L'heure est précise au seconde près. Des doublons sont de ce fait présents dans le fichier. Il est nécessaire de les enlever parce qu'ils n'apportent pas d'informations supplémentaires.

Page visitée

Une page web visitée est de cette forme : www.google.com. On travaille avec les noms de domaine de l'ordinateur hébergeant la ressource demandée. L'URL (Uniform Resource Locator) de la page visitée ou de la ressource demandée est plus riche car elle a la structure suivante : protocole/nom de domaine/chemin.

> **Exemple 37** 629f4cf8369f4c3f44a2faeeb8e8af4b3783d081 est le hash du login de l'utilisateur qui a visité www.google-analytics.com, le 15 novembre 2013 à 22h 45min et 42s. Cette utilisateur utilise l'adresse IP 172.17.181.98.

Nous avons choisi de travailler avec 6 mois de données. Les données de travail sont extraites du 15 novembre 2013 au 15 mai 2014. Cette période correspond à une forte

6. Un hash cryptographique est une série de chiffres et lettres générée par une fonction de hash cryptographique. Une fonction de hash cryptographique est un algorithme permettant de générer une sortie, un hash, à partir d'un mot ou une phrase par exemple.

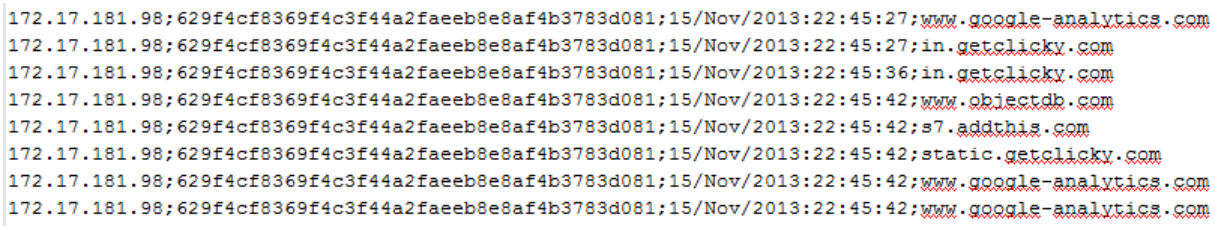


FIGURE 4.2 – Fichier des données d'entrée

activité dans le milieu universitaire. Avant de procéder au nettoyage des données, nous avons fait une étude statistiques de nos données. Sur 6 mois, nous avons 1, 5 Go de données et 176 fichiers. Normalement, nous devrions avoir 181 fichiers, mais il y a 5 jours sans aucune activité enregistrée. Après suppression des doublons, la base de données est composée de 7800143 requêtes. Dans le tableau 4.1, nous présentons le nombre d'utilisateurs, le nombre d'adresses IP et le nombre de sites distincts.

Adresses IP	**Utilisateurs**	**Noms de domaine**
451	3327	173200

TABLE 4.1 – Statistiques descriptives de la base de données

4.3 Fouille des données de navigation sur Internet

L'observation du comportement des internautes a fait l'objet de plusieurs travaux. Le contexte particulier de l'authentification implicite pour la navigation Web a été étudié dans [2] et [65].

Dans l'étude de [65], une base de données de 10 utilisateurs sur 1 mois est utilisée pour tester le modèle proposé. Les auteurs utilisent des données de navigation web multi-sites pour faire de l'authentification. Ils se basent sur la date et heure des rêquetes mais aussi sur l'URL visitée. Une étape de pré-traitement leur permet d'extraire d'autres types de données pour leur étude tel que le genre de la page visitée (site d'information, de e-commerce...). Le modèle proposé dans [65] utilise la machine à vecteurs de support (cf. [133]) de LibSVM sous le logiciel Weka [7] pour construire le profil des utilisateurs. Cette étude conclut que les caractéristiques utilisées ne sont

7. Weka (Waikato Environment for Knowledge Analysis) est une suite de logiciels d'apprentissage automatique. Écrite en Java et développée à l'université de Waikato, Nouvelle-Zélande, Weka est un logiciel libre.

pas suffisantes pour authentifier ou distinguer les utilisateurs. La taille de la base de données est aussi un paramètre déterminant qui pourrait expliquer pourquoi leur modèle n'est pas capable de fournir de bons résultats.

Le travail de [2] permet de faire de l'identification sur Internet en se basant sur des données de navigation web multi-sites également. Elle utilise comme éléments caractéristiques du comportement de l'internaute : le nom de domaine visité, le nombre de pages vues, l'heure de démarrage d'une session et sa durée. Le jeu de données, collecté par un fournisseur de services, est constitué des 300 premières connections de 2798 utilisateurs sur une période de 12 mois. Les profils utilisateurs sont composés d'*itemsets* fréquents de taille 1, extraits avec l'algorithme Apriori (cf. [134]). Le modèle décrit dans [2] est basé sur des calculs de fréquence d'apparition des *itemsets*. L'auteur compare deux approches de sélection d'*itemsets* fréquents en particulier : le support et le lift. L'étude montre que pour des comportements anonymes de petite taille (jusqu'à une vingtaine de sites visités) les modèles les plus efficaces restent des modèles de classification traditionnels comme les arbres de décision. Par contre, dès que le comportement anonyme dépasse une trentaine de sites les modèles de classification à base de support et de lift sont les plus précis.

Dans [135], les auteurs appliquent leur méthode d'extraction de comportements d'internautes à partir de deux mois de données d'utilisation d'un site [8]. La sémantique de navigation de l'internaute sur un site est capturée. Dans cette étude, une session de navigation est un ensemble ordonné de requêtes HTTP tel que deux requêtes consécutives ne soient pas séparées par plus de 30 minutes, ce qui est très proche de $25,5$ minutes (cf. [136]). En effet, un temps de pause de $25,5$ minutes permet de définir une session. A partir d'un ensemble de sessions d'URL, les auteurs en déduisent le nombre de fois qu'une page a été demandée, le nombre de fois qu'une page a été la première page d'une session et enfin le nombre de fois qu'une page a été la dernière page d'une session. Les auteurs proposent une modélisation des parcours sur un site sous forme de grammaire probabiliste hypertextuelle ou en anglais *Hypertext Probabilistic Grammar (HPG)* qui permet de fouiller des bases de parcours et d'identifier des séquences [9] récurrentes suivies par les internautes. Les auteurs utilisent un modèle à base de N-grammes (cf. [137]) et supposent que les N dernières pages visitées affectent la probabilité de visite de la prochaine page. Dans cette étude, le pré-traitement des données n'a pas été effectué.

8. http ://machines.hyperreal.org/, accès 02/11/2015

9. Une séquence $\mathscr{S}$ est une liste ordonnée d'*itemsets* non vides. Une séquence utilise le principe de précédence c'est-à-dire chaque élément de la liste est précédé des éléments qui l'ont précédé dans les transactions d'un client donné.

Dans la littérature, les requêtes HTTP sont des ensembles d'URL. L'information extraite est plus riche. Dans un contexte multi-sites, l'étude présentée dans cette thèse utilise des noms de domaine entre autres pour définir des comportements caractéristiques aux internautes.

4.4 Pré-traitements sur les données

La phase de pré-traitement est très importante car elle permet d'améliorer la qualité de la base de données. Plus les données sont nettoyées et meilleurs sont les résultats issus de l'étape de fouille de données.

Dans le cadre de cette étude, les requêtes enregistrées dans la base de données qui n'ont pas été faites par les internautes, sont supprimées. En effet, les requêtes issues de l'affichage d'un pop-up ainsi que les requêtes provenant d'antivirus, de navigateurs, et de diverses applications ne montre pas le comportement de l'utilisateur. Nous appelons les requêtes provenant d'antivirus, de navigateurs et de diverses applications, des requêtes issues de « robots ». Les pop-ups sont de par leur nature, involontaires et non désirés. Ils sont supprimés par l'utilisation de techniques de filtrage empruntées aux logiciels anti-publicités pouvant se trouver sur le Web. Durant l'étape de pré-traitement, nous changeons le format des données afin d'avoir une base de données similaire à une base de transactions. Ce format est adapté à l'extraction d'*itemsets* fermés fréquents. Nous définissons pour cela les sessions et mettons en place une base de sessions. Enfin nous adaptons nos fichiers au format d'entrée des algorithmes.

4.4.1 Page de publicité

Pour les étapes de pré-traitement, nous nous inspirons des travaux décrits dans [138]. N'ayant pas accès à des informations comme la topologie des pages Web consultées, il est compliqué d'adapter les méthodes de la littérature (cf. [138]).

Le nettoyage a pour but de ne conserver que les requêtes effectuées par un utilisateur, et non celles provenant de serveurs de publicités produisant des fenêtres intempestives. Pour cela, nous utilisons les filtres pour bloquer les publicités disponibles sur le Web. Nous utilisons des listes de filtrage de régies publicitaires [10]. Nous parcourons les requêtes, et pour chaque requête, nous recherchons s'il s'agit d'une publicité. Cette recherche s'effectue via la consultation en ligne des listes, ou via un fichier de cache pour accélérer les requêtes. Ce fichier de cache est régulièrement mis à jour. La

10. http ://winhelp2002.mvps.org/hosts.txt, http ://pgl.yoyo.org/, https ://easylist-downloads.adblockplus.org/easylist.txt, accès 29/10/2015.

consultation en ligne des listes est effectuée avec des requêtes HTTP non-bloquantes pour gérer le cas où les ressources sur le Web sont inaccessibles. Ce traitement nous a permis de réduire de 12.5% le nombre de requêtes.

La gestion des robots a pour but de supprimer toutes les requêtes effectuées par l'ordinateur, par les navigateurs et par les moteurs de recherche. Ces requêtes sont reconnaissables par leurs grand nombre d'occurrence en un très court laps de temps. Les requêtes provenant de robots sont écartées à l'aide d'un algorithme qui les repère grâce à leur fréquence d'apparition. En effet, les robots de navigateur/moteurs de recherche ont une vitesse d'exécution supérieure aux humains qui leur permet d'atteindre plusieurs pages en un temps très faible. Pour ce faire nous calculons le nombre de requêtes par seconde et nous effectuons un seuillage. Nous enlevons arbitrairement toutes les requêtes apparaissant plus de trois fois durant la même seconde.

Enfin, du fait que l'horodatage des données est précis à la seconde près, nous avons des doublons dans les données. Nous supprimons les doublons parce que ça ne nous apporte pas une information exploitable.

4.4.2 Définition des sessions

Dans nos données, il n'existe pas de notion de session. Nous avons l'ensemble des requêtes des utilisateurs sous forme de flux. L'objectif des sessions est de regrouper les requêtes, afin de créer un sous-ensemble rationnel de sites visitées représentant un comportement utilisateur sur une période. Les sessions sont construites à partir de l'horodatage des requêtes. Lors de la navigation sur Internet, une session est définie dans la littérature par une pause de $25,5$ minutes de l'internaute (cf. [136]). Lorsque nous avons implémenté cette méthode, notre base de session était pauvre. Nous n'avions pas assez de sessions pour faire de la fouille de données. Nous décidons alors de construire des sessions de taille fixe. Par exemple, un enchainement de 10 requêtes constitue une session de taille fixe 10. Dans le tableau 4.2, 10 est la taille des sessions. Ici, nous nettoyons les données et sélectionnons trente utilisateurs. Nous avons alors 7698 sessions dans notre base. Les internautes ont en moyenne 257 sessions.

	Minimum	**Maximum**	**Moyenne**	**Écart-type**
#sessions/utilisateur	101	733	257	289

TABLE 4.2 – Statistiques descriptives d'un jeu de données de 30 utilisateurs

4.5 Résumé

Dans ce chapitre, nous avons présenté le contexte applicatif relatif à l'étude de données de navigation sur Internet.

Le pré-traitement effectué sur la base de données permet ensuite d'appliquer la méthodologie présentée dans le chapitre 3. Cette étape permet particulièrement de supprimer les pages publicitaires de la base de données mais aussi de construire les sessions des utilisateurs. Dans le chapitre suivant, les résultats issus de la classification sont présentés. L'intérêt et l'originalité de cette application est d'être la première à notre connaissance à mettre en œuvre l'étude comportementale des internautes à base de motifs fermés fréquents pour de l'authentification implicite.

Chapitre 5

Expérimentations

Sommaire

3.1 Introduction 92

3.2 Motivations : limites des approches existantes 95

3.3 Classification naïve bayésienne 99

3.3.1 Adaptation simple de la classification naïve bayésienne . . 99

3.3.2 Classification naïve bayésienne à base d'*itemsets* fermés fréquents discriminants 101

3.4 Classification à base d'*itemsets* fermés fréquents discriminants 105

3.4.1 Sélection des patterns propres 108

3.4.2 Calcul des profils utilisateurs 112

3.4.3 Mesures de distance 114

3.5 Résumé 115

L'esprit n'use de sa faculté créatrice
que quand l'expérience lui en
impose la nécessité.

Henri Poincaré, (1854 - 1912).

5.1 Introduction

Nous avons présenté le contexte applicatif utilisé dans ces travaux dans le chapitre précédent. Nous avons aussi abordé le processus de transformation du format des données. Des jeux de données obtenus à l'issu de cette transformation sont utilisés en entrée des méthodes exposées dans les chapitres précédents. Ici, nous présentons les résultats obtenus. Nous présentons également un logiciel de gestion des données. Cette interface permet de traiter les données et de tester les méthodes proposées plus facilement.

Dans la section 5.2, les résultats obtenus avec les méthodes se basant sur la classification de Bayes sont présentés ainsi que les résultats obtenus de la classification à base d'*itemsets* fermés fréquents discriminants. Le logiciel de visualisation des résultats est ensuite présenté dans la section 5.3. Enfin, nous discutons de la connaissance extraite à l'aide des méthodes proposées et donnons un résumé du chapitre dans la section 5.4.

5.2 Application des différentes méthodes

Nous avons un jeu de données de référence de 30 utilisateurs. Ces utilisateurs sont les plus actives de la base de données c'est-à-dire qu'ils possèdent le plus grand nombre de sessions. Il représente 2% de la base de données, et ce choix est justifié par les travaux décrits dans [2]. En effet, l'auteur recommande de prendre les utilisateurs les plus actifs lorsqu'il s'agit de faire une recherche d'objets fréquents. Dans [2], 2% des utilisateurs sont sélectionnés pour l'étude. Nous appliquons également nos modèles à des jeux de données de la littérature [1] afin de comparer nos travaux avec quelques travaux existants.

1. http ://www.kdnuggets.com/, accès 02/11/2015

Evaluation

Pour évaluer la fiabilité des méthodes, nous construisons une matrice de confusion. Chaque colonne de la matrice représente le nombre d'occurrences d'une classe estimée, tandis que chaque ligne représente le nombre d'occurrences d'une classe réelle. Un des intérêts de la matrice de confusion est qu'elle montre rapidement si le système parvient à classifier correctement.

Classes	**Alice par le classifieur**	**Bob par le classifieur**
Alice	24	19
Bob	5	56

TABLE 5.1 – Exemple de matrice de confusion

Exemple 38 La matrice de confusion du tableau 5.1 montre que 24 sessions ont été prédites comme appartenant à Alice et ces sessions appartiennent bien à Alice. Le classifieur a prédit également que 5 de Bob appartiennent à Alice. Le clasifieur idéal maximise les éléments de la diagonale de la matrice.

La base de donnée est divisée en deux ensembles : la base d'apprentissage et la base de test. Ce partitionnement est réalisé de manière aléatoire. En général, $\frac{2}{3}$ des données constitue la base d'apprentissage et le reste permet de construire la base de test soit $\frac{1}{3}$ des données. Rappelons que la base d'apprentissage permet de construire le modèle. Ensuite la fiabilité du modèle est estimée avec la base de test. Ce processus, répété plusieurs fois, est appelé *random subsampling* (cf. [139]).

L'évaluation de l'*accuracy* nécessite d'abord la définition de quatre termes. Les vrais positifs (VP) représentent les observations positives qui ont été correctement classées par le classifieur. VP est le nombre de vrais positifs. Les vrais négatifs (VN) sont les observations négatives qui ont été correctement classées par le classifieur. VN est le nombre de vrais négatifs. Les faux positifs (FP) sont les observations négatives qui ont été incorrectement classées comme positives. FP est le nombre de faux positifs. Enfin, les faux négatifs (FN) sont les observations positives qui ont été classées comme négatifs. FN est le nombre de faux négatifs. P est le nombre d'observations positives soit $P = VP + FN$ et N est le nombre d'observations négatives soit $N = FP + VN$.

Exemple 39 Sur la tableau 5.1, $VP = 24$, $VN = 56$, $FP = 5$ et $FN = 19$. Dans cet exemple, $P = 43$ et $N = 61$.

La mesure de l'*accuracy* d'un classifieur sur une base de test donné donne le pourcentage d'observations de la base de test qui ont été correctement classifiées.

$$accuracy = \frac{VP + VN}{P + N}$$

Exemple 40 Sur la tableau 5.1, $accuracy = 76\%$ soit 76% des sessions de test ont été bien classées.

Dans la suite, nous utilisons notre jeu de données de référence constitué des 30 utilisateurs les plus actifs de la base de données de six mois. Nous exécutons les modèles naïfs bayésienne 100 fois afin de lisser les résultats. Sur tous les graphiques présentés par la suite, l'abscisse d'un point représente le nombre de sessions dans la base de test et l'ordonnée, l'*accuracy* en pourcentage calculé à partir de la matrice de confusion de chaque méthode. Notons que nous exécutions aussi nos méthodes sur des bases de données de taille variable. La base de données des 30 utilisateurs est partitionnée afin de tester des flux de sessions de taille différente.

5.2.1 Classification naïve bayésienne

Dans cette section, nous présentons les résultats issus de la classification naïve bayésienne et de ses variantes.

Influence du correctif

Rappelons que le correctif de Laplace permet d'éviter le problème suivant : lorsqu'un site de la base de test n'a pas été rencontré dans la base d'apprentissage alors la valeur de la probabilité est nulle. Cette valeur nulle annule l'effet des autres valeurs de probabilités qui permettent de calculer la probabilité qu'une session de test appartienne à une classe. Une classe est un utilisateur. Dans nos expérimentations, nous n'avons pas mis en place une fonction qui calcule le correctif de Laplace. Nous avons fait varier un facteur multiplicatif et avons estimé le correctif de Laplace. Dans notre étude, nous avons en moyenne 257 sessions par utilisateur soit 2570 sites visités. La base d'apprentissage représente $\frac{2}{3}$ des données. La base d'apprentissage est constituée à peu près de 1700 sites visités par utilisateur soit 51400 sites au total. Pour estimer le correctif de Laplace, il est nécessaire de connaitre le support de chaque site de la base d'apprentissage. Le correctif de Laplace est estimé à 10^{-6} .

Sur la figure 5.1, nous présentons les résultats de l'application de la classification naïve bayésienne sur notre jeux de donnée (courbe bleue et verte). Le site visité est le descripteur. Les résultats s'effondrent très rapidement. Le facteur correctif influe

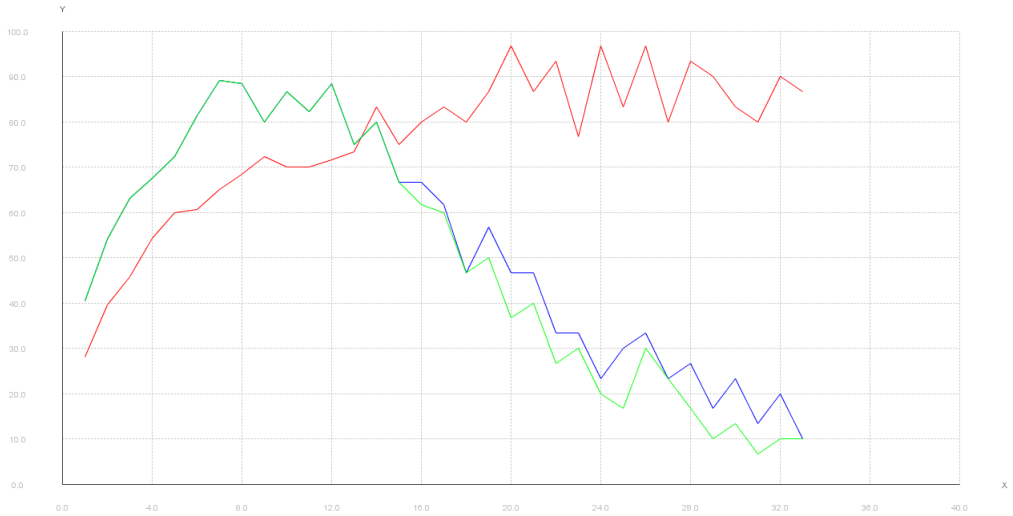


FIGURE 5.1 – Influence d'un correctif sur la classification de Bayes.

sur les résultats. Cependant, on observe que quelque soit le correctif non nul utilisé, les résultats restent inchangés pour les flux de test de petites tailles (entre 1 et 8 sessions). Le correctif a une plus grande influence lorsque l'utilisateur visite beaucoup de sites. En effet, plus il visite des sites, plus la probabilité d'accéder à des sites qu'il n'a jamais visité, dans la base de test, augmente. Dans tous les cas, la classification naïve bayésienne donne des résultats médiocres lorsque le nombre de sites visités augmente. L'explication de la chute de la courbe est donnée par la multiplication d'un grand nombre de probabilités qui tendent vers la valeur nulle.

Classification naïve bayésienne à base d'*itemsets* fermés fréquents discriminants

Dans le protocole expérimental, si la session test X ne contient aucun *itemset* fermé fréquent discriminant connu dans la base d'apprentissage de l'utilisateur alors $P(X|u_i)$ est égale au correctif de Laplace soit de l'ordre de 10^{-3}.

Version 1 et Version 2

La courbe verte du graphique 5.2 montre les résultats de la classification naïve de bayes à base de fermés fréquents discriminants. La médiocrité des résultats peut s'expliquer par le calcul de support. En effet, il est quasi impossible de retrouver tous les fermés fréquents discriminants d'un utilisateur dans une session test. Du

coup, la valeur de ce support dépasse très rarement 1. Une manière de résoudre le problème est de supposer que si on trouve qu'un fermé fréquent discriminant de l'utilisateur est contenu dans la session test alors l'utilisateur est reconnu. La courbe verte du graphique 5.3 montre les résultats d'une expérimentation qui prend en compte l'hypothèse précédente. Cependant cette hypothèse n'améliore pas les résultats.

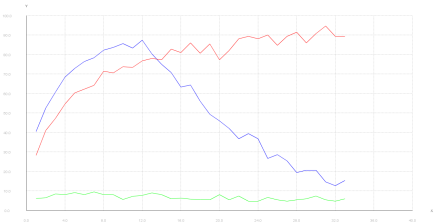


FIGURE 5.2 – Bayes à base de fermés.

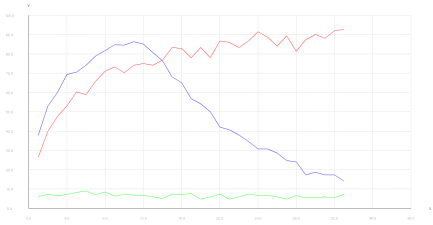


FIGURE 5.3 – Bayes à base de fermés avec l'hypothèse d'amélioration.

Version 3

En appliquant les filtres proposés, les résultats de la classification à base d'*itemsets* fermés fréquents discriminants sont améliorés de 2 à 5 %.

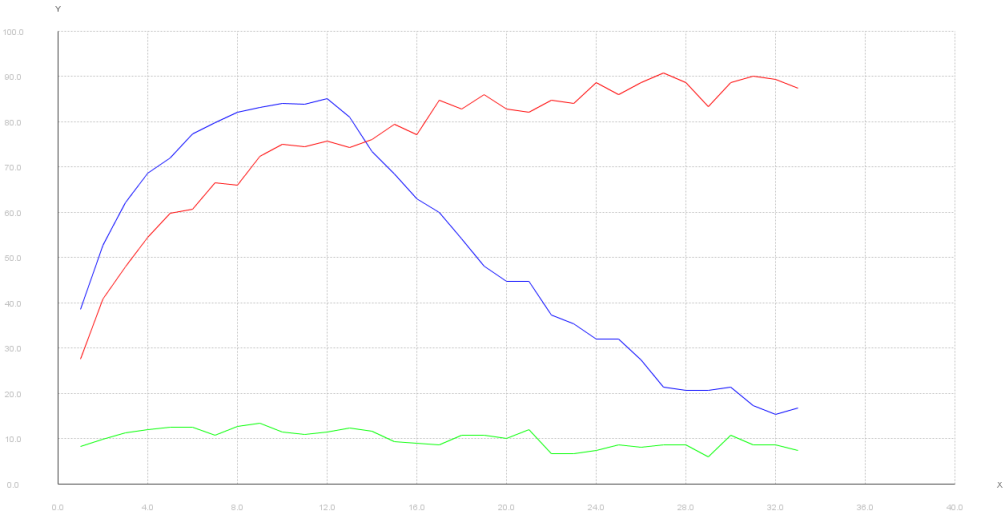


FIGURE 5.4 – Bayes à base de fermés fréquents discriminants + filtre.

Version 4

A partir d'un code écrit en JAVA et trouvé sur Github [129], nous avons adapté la méthode de classification de texte par Bayes à notre étude. La méthode consiste à utiliser la classification naïve bayésienne sur une base d'apprentissage épurée à l'aide d'algorithme de sélection. Le processus de sélection est mis en place avant le processus d'apprentissage. La sélection permet de réduire la quantité de données d'apprentissage, et améliore la vitesse de la phase d'apprentissage. La méthode de sélection utilisée ici est le test de χ^2. La base d'apprentissage, constituée de 2/3 des données est réduit à une base constituée que de descripteurs qui satisfont le test d'indépendance de χ^2. Cet ensemble réduit est utilisé dans l'algorithme de classification, ici la classification naïve bayésienne. Ces algorithmes sélectionnent les descripteurs les plus représentatives de la base d'apprentissage. Ce test permet de tester l'indépendance de deux événements. Ici, il est utilisé pour tester si l'occurrence d'un site visité particulier et l'occurrence d'une classe particulière sont indépendants. Dans cette implémentation, un score de χ^2 est calculé pour chaque descripteur potentiel (un site visité qui permettrait de décrire un utilisateur) et en fonction de chaque classe. Si le score est supérieur à une valeur critique de 10,83 (un paramètre fixé dans le code) alors le descripteur et la classe (utilisateur) sont dépendants. Si les deux événements sont dépendants alors on sélectionne le descripteur (site web visité) pour la classification.

Les résultats issus de la classification de Bayes après sélection avec le test de χ^2 sont meilleurs que ceux issus des autres modèles proposés lorsqu'on a une session test. En effet, avec une session test, le modèle basé sur le test de χ^2 est capable de retrouver un utilisateur parmi 20 avec une précision d'environ 50 %.

La figure 5.5 compare les performances des différents modèles naïfs bayésiens. L'exécution est lancé 50 fois. Cette figure montre particulièrement que l'utilisation du filtre de χ^2 en amont est très efficace pour classer les sessions anonymes de petites tailles. Nous observons que la classification naïve bayésienne à base d'*itemsets* fermés fréquents discriminants permet d'avoir des résultats médiocres.

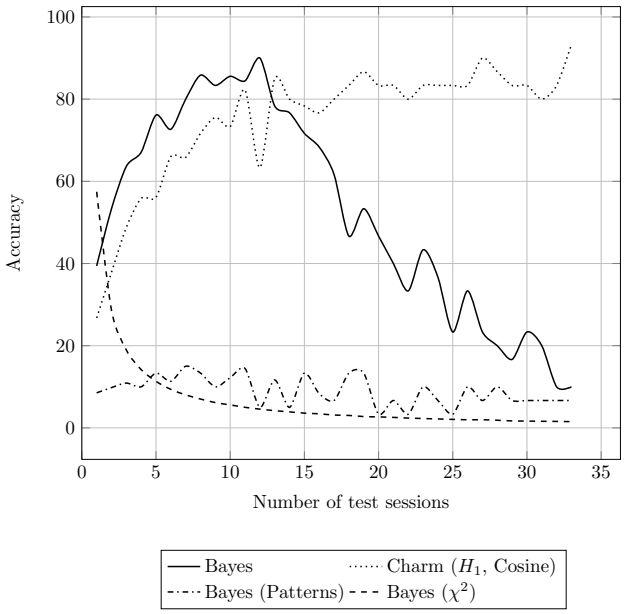


FIGURE 5.5 – Performances comparatives des modèles naïfs de *Bayes*, *Bayes*(χ^2) et *bayésien à base d'*itemsets *fermés fréquents discriminants*.

5.2.2 Classification à base de d'*itemsets* fermés fréquents discriminants

Description du protocole expérimental

L'algorithme 4 décrit le protocole expérimental proposé. La première boucle permet de définir le nombre d'utilisateurs totaux parmi lesquels une session sera classée. La deuxième boucle indique le nombre d'informations porté par chaque flux de test (une seule session, ou bien une agrégation de plusieurs sessions). Enfin, la troisième boucle permet de lisser les résultats en créant plusieurs jeux d'exécution (cf. *random subsampling*). Dans la table 5.3, nous fixons X à 100. La boucle ligne 10 calcule les *itemsets* fermés fréquents spécifiques à chaque utilisateur et établit les vecteurs de profil. La boucle ligne 13 calcule la valeur de chaque vecteur de profil utilisateur. Enfin, les boucles aux lignes 16 et 18 classent les données tests et calculent le taux de précision.

Algorithme 4 : Procédure d'expérimentation

Entrées : $\bigcup_i \mathcal{S}_{u_i}$: l'ensemble des sessions des n utilisateurs ;
X : nombre d'exécutions ;
Résultat : La précision moyenne des modèles sélectionnés ;

```
1  début
2      pour (N = {2, 5, 10, 20, 30}) faire
3          pour (S = {1, 10, 20, 30}) faire
4              pour (z = 1, ..., X) faire
5                  Sélectionner N utilisateurs aléatoirement ;
6                  Pour chaque utilisateur, sélectionner
                   SN = min(|S_{u_i}|, i = 1, ..., N) ;
7                  Prendre les 2/3 des SN sessions pour chaque utilisateur pour
                   former la base d'apprentissage ;
8                  Prendre le reste des SN sessions pour former la base de test ;
9                  P^k_all ← ∅ (le vecteur de profil global pour chaque modèle k) ;
10                 pour chaque (u_i, i = 1, ..., N) faire
11                     Calculer les itemsets fermés fréquents propres P^k_{u_i}
                       (1 ≤ |P^k_{u_i}| ≤ 10) ;
12                     P^k_all ← P^k_all ∪ P^k_{u_i} ;
13                 pour chaque (u_i, i = 1, ..., N) faire
14                     Calculer le vecteur V^k_{u_i} en utilisant soit le support, soit le lift ;
15                 Initialiser à 0 la matrice de confusion M^k de la méthode k ;
16                 pour chaque (u_i, i = 1, ..., N faire
17                     Calculer le flux de test T_{u_i} (|T| est fixée, T ∈ T_{u_i}) ;
18                     tant que (T_{u_i} ≠ ∅) faire
19                         Prendre SW sessions de T_{u_i} pour construire V^k_T ;
20                         u_a ← max(simil(V^k_{u_i}, V^k_T)) ou min(dist(V^k_{u_i}, V^k_T)) ;
21                         M^k[u_i][u_a] ← M^k[u_i][u_a] + 1 ;
22             Calculer la précision moyenne de k à partir de M^k ;
```

Performances comparatives de H_1, H_2 et H_3

A partir des *itemsets* fermés fréquents propres à chaque utilisateur, nous calculons l'ensemble $\mathcal{P}_{all}$ comme étant l'ensemble des*itemsets* fermés fréquents propres qui définissent le vecteur de profil de chaque utilisateur. Nous utilisons le support d'un

itemset fermé fréquent comme une valeur numérique pour chaque composante du vecteur. La table 5.2 fournit la taille de chaque vecteur de profil, ainsi que la distribution de la taille des *itemsets* fermés fréquents propres au sein de celui-ci par rapport aux heuristiques choisies. Avec 30 utilisateurs et 10 *itemsets* fermés fréquents propres par utilisateurs, la taille maximale du vecteur profil est de 300. Le table montre qu'en effet, les vecteurs n'atteignent pas la taille maximale, car certains utilisateurs peuvent partager les mêmes *itemsets* fermés fréquents propres. On observe que plus de la moitié des *itemsets* fermés fréquents propres sélectionnés par les heuristiques sont de taille inférieure ou égale à 3.

	Taille	|1|	|2|	|3|	|4|	|5|	|6|	|7|
H_1	199	18%	31%	26%	16%	7%	2%	0%
H_2	167	57%	29%	9%	3%	1%	1%	0%
H_3	199	24%	20%	18%	14%	10%	9%	5%

TABLE 5.2 – Taille du vecteur profil et distribution des *itemsets* fermés fréquents propres par rapport à leur taille.

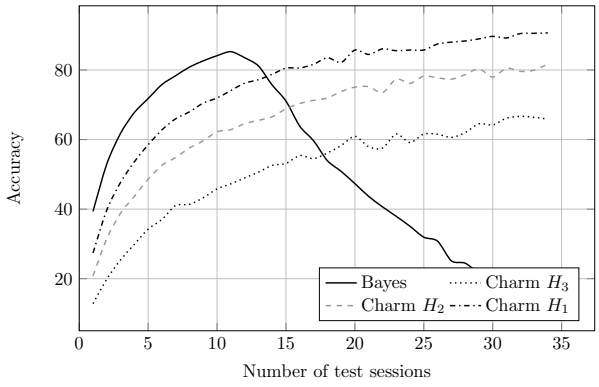


FIGURE 5.6 – Performance comparative de H_1, H_2, H_3 et Bayes.

La figure 5.6 permet d'observer que le classifieur naïf bayésien est la méthode la plus efficace lorsque le flux de test est constitué de sites internet issus de 1 à 7 sessions (jusqu'à 70 sites internet). Ces résultats sont en concordance avec ceux donnés dans [2]. Finalement, cette figure montre clairement l'efficacité de l'heuristique

H_1 par rapport à H_2 et H_3. Rappelons que l'heuristique H_1 choisit *itemsets* fermés fréquents propres parmi les *itemsets* fermés fréquents qui ont la plus grande valeur de $tf \times idf$. Les *itemsets* fermés fréquents de petite taille (un, deux ou trois sites) sont très prédictifs sur notre jeu de données.

Performances comparatives avec [2]

Dans [2], l'auteur utilise comme motif, les *itemsets* fréquents extraits avec l'algorithme Apriori (cf. [134]). Dans son étude, il compare, en particulier, deux méthodes de calcul du vecteur de profil. Dans les deux cas, les *itemsets* fréquents propres sont de taille 1 et sont choisis parmi les sites internet les plus fréquents. La première méthode, appelée *support-based profiling*, utilise le support de chaque *itemsets* fréquents comme valeur numérique pour chaque composante du vecteur de profil. La seconde méthode, appelée *lift-based profiling*, utilise la mesure du lift. Afin de comparer les performances de H_3 avec les deux modèles *support-based profiling* et *lift-based profiling*, nous avons minutieusement répliqué le protocole expérimental décrit dans [2] sur notre jeu de données. Les résultats sont donnés dans la table 5.3.

Sur la gauche de la table 5.3, nous trouvons le nombre d'utilisateurs et les modèles sélectionnés. Chaque colonne est définie par le nombre de sessions de test exécutées simultanément. Ces sessions sont de taille 10. Les données de la table 5.3 montrent que la méthode H_1 offre des précisions bien plus significatives que les deux modèles proposés dans [2] dans tous les scénarios. Cependant, le classifieur naïf bayésien reste le plus performant lorsque le groupe de sessions est de taille 1 (ce qui concorde avec les observations décrites dans [2]). Chaque ligne de la Table 5.3 correspond à une partition de la base de données. Pour la première ligne, nous classons 1 utilisateur parmi 2, pour la deuxième ligne, nous classons 1 utilisateur parmi 5, etc. Cela nous permet de suivre l'évolution des précisions en fonction du nombre d'utilisateurs parmi lesquels les sessions sont classées.

# of users		1	10	20	30
2	Support	65	89	95	97
	Lift	67	90	97	98
	Charm H_1	72	98	**99**	**100**
	Bayes	**85**	**99**	73	61
5	Support	40	74	83	88
	Lift	41	78	86	88
	Charm H_1	49	90	**95**	**98**
	Bayes	**67**	**96**	56	34
10	Support	27	66	79	80
	Lift	29	64	77	80
	Charm H_1	37	83	**92**	**94**
	Bayes	**54**	**91**	51	24
20	Support	19	55	68	75
	Lift	21	58	68	74
	Charm H_1	30	76	**86**	**90**
	Bayes	**43**	**87**	48	19
30	Support	16	53	64	70
	Lift	17	54	64	69
	Charm H_1	26	72	**83**	**89**
	Bayes	**39**	**83**	46	19

TABLE 5.3 – Performances comparatives avec l'étude décrit dans [2]

La figure 5.7 offre une bonne représentation des résultats, particulièrement lors de la croisée des précisions entre les courbes de Bayes et de H_1. Sur cette figure, les performances comparatives entre le modèle naïf bayésien, le *support-based profiling*, le *lift-based profiling* et H_1 sont présentées.

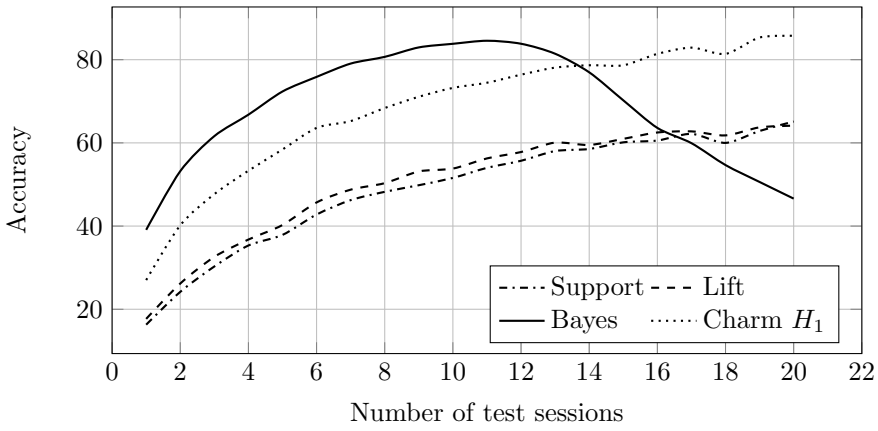


FIGURE 5.7 – Performance comparative de Bayes, *support-based profiling*, *lift-based profiling* et H_1.

Performances comparatives des fonctions de distance

Les figures 5.8 et 5.9 montre l'impact du choix de la fonction de distance sur les performances des modèles. Sur la figure 5.8, nous comparons les performances de la similarité par cosinus et la distance euclidienne. Avec la similarité par cosinus la méthode avec l'heuristique H_1 et la méthode avec le lift sont améliorées de 5%. Sur la figure 5.9, nous comparons la similarité par cosinus avec deux autres fonctions de distance. Nous pouvons voir que les résultats issus des méthodes avec les trois fonctions sont très proches pour les ensembles de sessions test petits. A partir d'un ensemble de 12 sessions, les performances de la méthode à base de H_1 utilisant la similarité de Kulczynski sont meilleures (à peu près 7%).

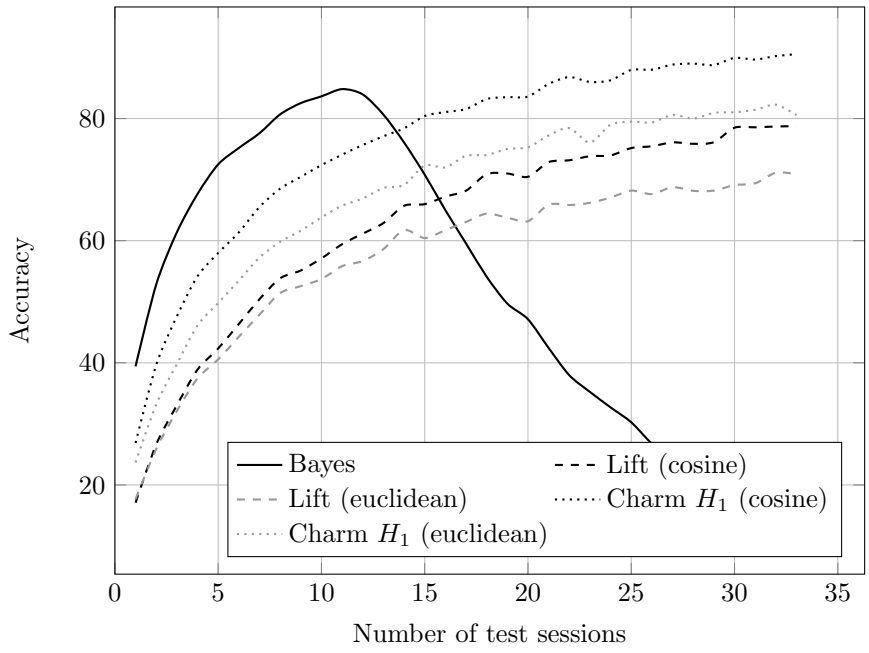


FIGURE 5.8 – Performance comparative de H_1 avec la similarité par cosinus et la distance euclidienne, Bayes et *lift-based profiling*.

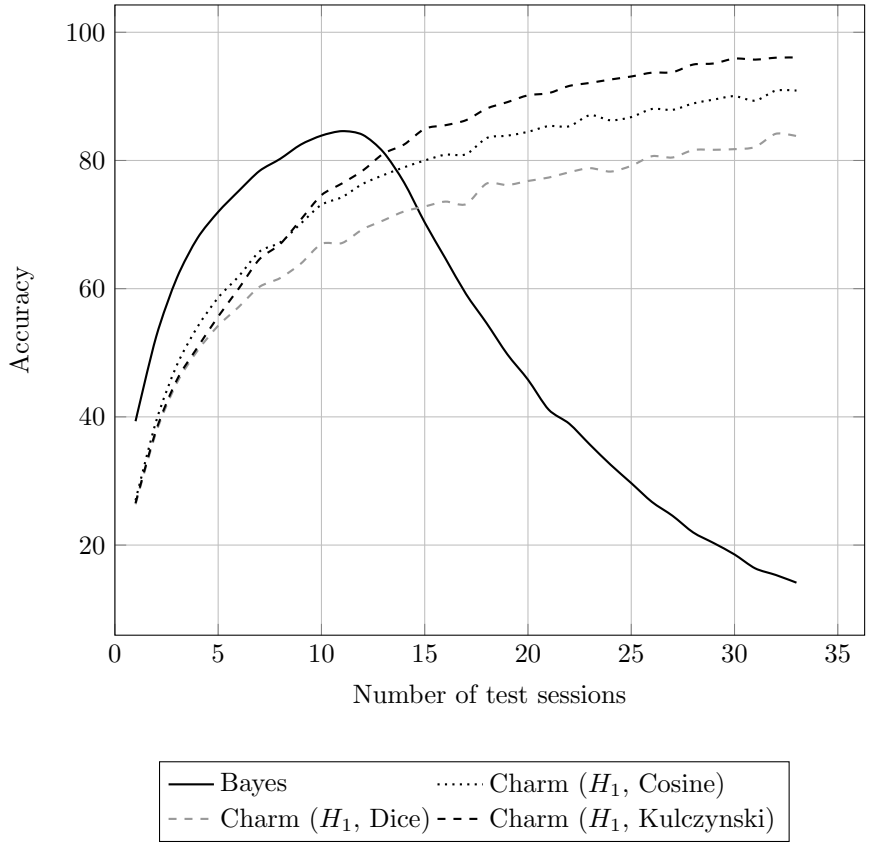


FIGURE 5.9 – Performance comparative de H_1 avec la similarité par cosinus, de Dice et de Kulczynski.

5.3 Logiciel de reconnaissance comportementale

5.4 Résumé et Discussion

Les résultats sont prometteurs dans l'ensemble bien que le passage à l'échelle pose des problèmes à certaines méthodes présentées.

La distance eucliedienne utilisée dans nos traavaux n'est pas adaptée à notre contexte d'étude.

Conclusion

Cette thèse est axée autour du thème de la confiance numérique au sein d'un espace numérique de travail. Nous avons travaillé sur deux éléments de confiance complémentaires : le modèle général de confiance et l'authentification implicite.

Produire un sentiment de confiance au sein des services numériques est un questionnement récent soulevé par les fournisseurs de services sur Internet. La création et le maintien de la confiance dans n'importe quel contexte sont des problématiques posées par les organismes, les médias mais aussi les États. Des modèles généraux de confiance ont été présentés dans la littérature. Ces modèles sont destinés dans la majorité des cas aux sites de vente en ligne. Nous avons proposé un modèle de confiance théorique adapté à un contexte particulier, eBeeHome. S'appuyant sur le modèle de détection d'intrusion dans les réseaux décrit dans [7], notre modèle est généralisable et applicable à d'autres contextes tels que la banque en ligne. Dans le modèle décrit dans cette thèse, nous formalisons le contexte d'étude et proposons des outils statistiques afin d'ouvrir des niveaux de service en fonction des niveaux de confiance.

Il est difficile d'exploiter les ensembles de données volumineux et complexes avec des outils classiques. Dans le cadre de cette thèse, nous nous sommes intéressés à un type de données particulier, l'*itemset* et à un domaine d'application particulier, la navigation sur internet. Les méthodes proposées peuvent être généralisées et appliquées à d'autres domaines où les données sont des *itemsets* comme les données textuelles. L'étude de données de navigation pour l'identification et l'authentification des internautes est une problématique compliquée mais pertinente. Beaucoup de travaux traitent de ce sujet, cependant des limites demeurent encore :

- Les données de navigation sur Internet sont très bruitées et nécessitent un pré-traitement particulier ;
- Les *itemsets* fermés fréquents extraits sont nombreux et il est difficile aujourd'hui de choisir les plus discriminants en fonction du contexte de l'étude ;

- Pour identifier ou authentifier un internaute, il est nécessaire que le profil construit corresponde réellement à son profil et que le taux d'erreur de reconnaissance soit minimisé.

Les travaux réalisés dans la deuxième partie de cette thèse apportent des solutions à ces trois limites identifiées.

Peu de travaux sur le pré-traitement des données de navigation sur Internet existent dans la littérature. Nous consacrons une partie de ces travaux à cette étape très importante car nous avons des données assez particulières. Ensuite, les techniques de classification à base d'*itemsets* fermés fréquents proposées dans l'état de l'art présentent certains inconvénients pour notre contexte d'étude.

L'étude des *itemsets* fermés fréquents discriminants est au cœur de ces travaux. Nous avons développé lors de cette thèse :

- Un logiciel de pré-traitement des données adapté à notre contexte d'étude ;
- Des algorithmes de sélection des meilleurs *itemsets* fermés fréquents discriminants pour notre étude ;
- Une technique de classification permettant d'authentifier un internaute.

5.5 Contributions

Modèle général de confiance

Le modèle général de confiance proposé permet d'attribuer des niveaux de confiance autorisant l'accès à des niveaux de service. Dans ce modèle, nous utilisons les représentations de la confiance décrites dans [11]. Les notions de réputation et de recommandation sont prises en compte dans le modèle. Des outils statistiques qui permettent de calculer des indices en considérant des politiques de confiance et de décision données par le fournisseur de services sont présentés. L'architecture de notre modèle s'appuie sur celui décrit dans [7]. L'anonymat des internautes est préservé en utilisant des analyseurs globales de confiance. Une architecture décentralisée permet de maintenir le système en cas d'attaques. Une étude profonde de l'état de l'art nous a permis de construire ce modèle.

Logiciel de pré-traitement des données

Un logiciel de pré-traitement des données a été fourni également lors de cette thèse. Le logiciel permet de transformer les fichiers de requêtes utilisateurs en entrée en des

fichiers de sessions nettoyés en sortie. Il permet aussi d'avoir un descriptif statistique des données en sortie. Rappelons que le logiciel permet d'enlever les doublons, de supprimer les pages publicitaires et enfin de construire les sessions de taille fixe. La taille des sessions est un paramètre que l'utilisateur du logiciel peut régler. Le pré-traitement des données s'inspire des travaux décrits dans [138] qui s'appliquent aux données issues de la navigation web à travers plusieurs sites Internet.

Classification de Bayes à base d'*itemsets* fermés fréquents discriminants

Nous avons utilisé la classification de Bayes sur nos données pré-traitées. Les performances de cette méthode décroissent lorsque les jeux de données de test deviennent grands. Nous proposons des variantes de cette classification classique afin de pallier cette faiblesse. Les variantes proposées se basent sur *itemsets* fermés fréquents discriminants. Les résultats ne s'améliorent pas. En effet, retrouver un site descripteur en utilisant la classification originelle de Bayes étant difficile, retrouver un ensemble de sites descripteur est encore plus compliqué. Par contre la classification de Bayes en utilisant le test de χ^2 pour sélectionner les meilleurs descripteurs donne des résultats encourageants lorsque la base de test est petite. Un passage à l'échelle montre les défauts de cette méthode.

Heuristiques de sélection d'*itemsets* fermés fréquents

Notre principale contribution est la proposition des heuristiques de sélection des meilleurs descripteurs. Les heuristiques présentées sont simples et faciles à implémenter. Nous avons comparé plusieurs heuristiques de sélection sur nos données. En utilisant le $tf \times idf$ emprunté au domaine de la recherche d'information, nous avons pu améliorer les résultats de la classification. Le $tf \times idf$ permet d'extraire les éléments les plus importants d'une base de données.

Classification à base d'*itemsets* fermés fréquents discriminants

Dans cette étude, nous avons proposé un modèle de fouille de données pour l'authentification implicite d'internautes. Nous avons proposé des algorithmes simples et originaux permettant d'extraire un ensemble d'*itemsets* fermés fréquents discriminants caractérisant chaque internaute. Les *itemsets* fermés fréquents discriminants extraits sont de taille variable et correspondent à des fermés issus du système de fermeture

induit par l'ensemble des sessions. En reproduisant le protocole expérimental décrit dans [2], nous avons montré que les performances de notre modèle sont sensiblement meilleures que certains modèles proposés dans la littérature. Nous avons aussi montré le rôle déterminant de la fonction de distance. Les performances de notre système de classification sont comparées avec celles de la classification bayésienne, qui est notre modèle de référence. Les jeux de données utilisés sont très importants dans l'exécution des algorithmes, les différents passages à l'échelle effectués au cours de nos recherches ont montré qu'une heuristique, au départ assez classique, pouvait devenir très efficace par la suite avec l'augmentation de la masse de données.

5.6 Perspectives

Implémenter le modèle général de confiance

Nous avons proposé ce modèle en s'appuyant sur le modèle décrit dans [7] qui a été implémenté. Les résultats de la thèse d'Anirban Basu sont satisfaisants, ce qui nous laisse penser que cette approche pourrait apporter une solution au problème de production de confiance. Cependant le modèle d'Anirban Basu a pour objectif la détection d'intrusion dans les réseaux. Ce modèle est complémentaire aux procédures de sécurité. D'après nos hypothèses présentées dans le chapitre, la confiance se présente comme un substitut de la sécurité. L'implémentation de ce modèle nous permettrait de valider ces hypothèses dans le contexte de la navigation sur un site internet. Le modèle de confiance proposé reste théorique. Tester les performances de ce modèle est nécessaire afin de le valider.

Optimiser le pré-traitement des données

Le logiciel de pré-traitement des données pourra être amélioré. En effet, dans [138], les auteurs travaillent sur des requêtes HTTP d'un ensemble de sites web. Ils possèdent la cartographie des sites. Les données de navigation sur Internet sont traitées efficacement car les auteurs détiennent des informations complètes sur requêtes. Dans notre étude, nous travaillons avec les noms de domaine des requêtes effectuées. Les informations sur lesquelles nous nous sommes basés pour l'étape de fouille de données sont incomplètes. Ensuite, nous avons utilisé des filtres pour supprimer les sites publicitaires. Ces filtres sont un ensemble de sites identifiés comme des sites publicitaires. Il est possible d'avoir des sites publicitaires non identifiés dans les listes de filtre. Nous pouvons aussi rencontrer des cas pour lesquels l'internaute demande

à accéder à un site publicitaire. Nous ne gérons pas ces cas et il sera intéressant de se pencher dessus.

Enfin on pourra tester le logiciel sur des bases de données plus importante. Nous pourrions exécuter nos procédures sur des données issues d'un an d'exploitation.

Améliorer les heuristiques de sélection

Bien que prometteurs, nous pensons que les heuristiques proposées peuvent être largement améliorées. Nous avons travaillé avec le $tf \times idf$ et le test de χ^2 mais il existe plusieurs outils statistiques qui pourraient être testés sur nos données.

Modéliser les objets à la base de la classification autrement

Nous avions émis l'idée d'utiliser la multi-valuation des données et de ne plus considérer un site internet comme présent ou absent de notre ensemble (gestion binaire des sites internet), mais de considérer le nombre de pages visitées. Les valeurs de chaque fonction de distance et le positionnement de chaque profil dans l'espace commun des sites importants deviendrait alors plus précis et complexe et permettrait une meilleure distinction de chaque profil utilisateur.

Intuitivement, la classification à base de séquences pourrait être adaptée à notre contexte. En effet, les requêtes des utilisateurs ont un ordre chronologique et donc se présentent comme un ensemble de données ordonné.

Authentification implicite

L'authentification des utilisateurs est, à l'heure actuelle, peu précise. Il est nécessaire, encore une fois, de bien comprendre comment fonctionnent les distances utilisées, dans leurs profondeurs, afin d'en déterminer des seuils efficaces permettant de bien distinguer et de bien cerner chaque profil utilisateur. Nous avons implémenté une mesure de la courbe de ROC pour chaque utilisateur afin de déterminer des seuils efficaces mais les résultats étaient très différents en fonction de l'utilisateur.

Publication dans le cadre de la thèse

Conférences internationales avec comité de lecture

- **Olivier Coupelon, Diyé Dia, Yannick Loiseau, Olivier Raynaud**, *Am I Really Who I Claim To Be ?*, The 2014 World Conference on Information Systems and Technologies (WorldCIST'14), Madère, Portugal ;
- **Diyé Dia, Olivier Coupelon, Yannick Loiseau, Olivier Raynaud**, *Implicit Authentication*, poster, The 27th International Conference of the Florida Artificial Intelligence Research Society (FLAIRS-27), Pensacola Beach, USA ;
- **Olivier Coupelon, Diyé Dia, Fabien Labernia, Yannick Loiseau, Olivier Raynaud**, *Using Closed Itemsets for Implicit User Authentication in Web Browsing*, The Eleventh International Conference on Concept Lattices and Their Applications (CLA 2014), Košice, Slovaquie.

Conférences et ateliers nationaux avec comité de lecture

- **Diyé Dia, Olivier Coupelon, Yannick Loiseau, Olivier Raynaud**, *Suis-je celui que je prétends être ?*, 25es Journées francophones d'Ingénierie des Connaissances (IC2014), Clermont-Ferrand, France ;
- **Olivier Coupelon, Diyé Dia, Fabien Labernia, Yannick Loiseau, Olivier Raynaud**, *Using Closed Itemsets for Implicit User Authentication in Web Browsing*, The first Symposium on Digital Trust in Auvergne (SDTA 2014), Clermont-Ferrand, France.

Bibliographie

[1] Jian Pei, Jiawei Han, Runying Mao, et al. Closet : An efficient algorithm for mining frequent closed itemsets. In *ACM SIGMOD workshop on research issues in data mining and knowledge discovery*, volume 4, pages 21–30, 2000.

[2] Yinghui Catherine Yang. Web user behavioral profiling for user identification. *Decision Support Systems*, (49) :pp. 261–271, 2010.

[3] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI magazine*, 17(3) :37, 1996.

[4] Daniel Kaplan and Renaud Francou. *La confiance numérique*. éditions FYP.

[5] Baromètre CDC-ACSEL. Résultats de la 3ème édition du baromètre caisse des dépôts/acsel sur la confiance des français dans le numérique. 2013.

[6] Alain Peyrefitte. *La société de confiance*. Odile Jacob.

[7] Anirban Basu. *A Reputation Framework for Behavioural History*. PhD thesis, University of Sussex, UK, 2010.

[8] Adam B. Seligman. Complexité du rôle, risque et emergence de la confiance. *Réseaux*, 4(108) :pp. 37–61, 2001.

[9] Artz Donovan and Gil Yolanda. A survey of trust in computer science and the semantic web. *Web Semantics : Science, Services and Agents on the WWW*, 5 :pp. 58–71, 2007.

[10] C. Adams, G. Bochmann, and Shi J. A trust model with statistical foundation. *18th International Federation for Information Processing (IFIP) World Computer Congress TC1 WG1.7 Workshop on Formal Aspects in Security and Trust (FAST), IFIP*, 173 :pp. 145–158, 2005.

[11] Stephen Paul Marsh. *Formalising Trust as a Computational Concept*. PhD thesis, University of Stirling, UK, 1994.

[12] Alfarez Abdul-Rahman and Stephen Hailes. Supporting trust in virtual communities. In *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*, pages 9–pp. IEEE, 2000.

[13] Felix Gomez Marmol and Gregorio Martínez Pérez. Security threats scenarios in trust and reputation models for distributed systems. *computers & security*, 28(7) :545–556, 2009.

[14] Lik Mui, Mojdeh Mohtashemi, and Ari Halberstadt. A computational model of trust and reputation. In *System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on*, pages 2431–2439. IEEE, 2002.

[15] Manuel Graòa and Adrian Agreda. Trust models and applications in communication and multi-agent systems. *Int. J. Know.-Based Intell. Eng. Syst.*, 17(2) :187–193, April 2013.

[16] Audun Jøsang, Tanja Ažderska, and Stephen Marsh. Trust transitivity and conditional belief reasoning. In *Trust Management VI*, pages 68–83. Springer, 2012.

[17] Sini Ruohomaa and Lea Kutvonen. Trust management survey. In *Trust Management*, pages 77–92. Springer, 2005.

[18] Van-Hoan Vu. *Infrastructure de gestion de la confiance sur internet*. PhD thesis, Ecole Nationale Supérieure des Mines de Saint-Etienne, France, 2010.

[19] Howard Chivers. Trust without identification. In *Security in Pervasive Computing*, pages 1–2. Springer, 2006.

[20] Łukasz Krzywiecki, Przemysław Kubiak, and Mirosław Kutyłowski. Stamp and extend - instant but undeniable timestamping based on lazy trees. pages pp. 5–24, 2012.

[21] J. Kohl and C. Neuman. The Kerberos Network Authentication Service (V5). RFC 1510 (Proposed Standard), September 1993. Obsoleted by RFC 4120.

[22] Tatyana Ryutov, Li Zhou, Clifford Neuman, Travis Leithead, and Kent E Seamons. Adaptive trust negotiation and access control. In *Proceedings of the tenth ACM symposium on Access control models and technologies*, pages 139–146. ACM, 2005.

[23] Jiangtao Li, Ninghui Li, and William H. Winsborough. Automated trust negotiation using cryptographic credentials. In *Proceedings of the 12th ACM Conference on Computer and Communications Security*, CCS '05, pages 46–57, New York, NY, USA, 2005. ACM.

[24] M. Winslett, T. Yu, K. E. Seamons, A. Hess, J. Jacobson, R. Jarvis, B. Smith, and L. Yu. Negotiating trust on the web. *IEEE Internet Comput.*, 6(6) :pp. 30–37, 2002.

[25] Messaoud Benantar. *Access control systems : security, identity management and trust models*. Springer Science & Business Media, 2006.

[26] Nathan Dimmock, András Belokosztolszki, David Eyers, Jean Bacon, and Ken Moody. Using trust and risk in role-based access control policies. In *Proceedings of the ninth ACM symposium on Access control models and technologies*, pages 156–162. ACM, 2004.

[27] Audun Josang, Kent A Varmedal, Christophe Rosenberger, and Rajendra Kumar. Service provider authentication assurance. In *Privacy, Security and Trust (PST), 2012 Tenth Annual International Conference on*, pages 203–210. IEEE, 2012.

[28] Dorothy Elizabeth Robling Denning. *Cryptography and Data Security*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1982.

[29] Ehud Gudes, Nurit Gal-oz, and Alon Grubshtein. Methods for computing trust and reputation while preserving privacy. *Proceedings of the 23rd Annual IFIP WG 11.3 Working Conference on Data and Applications Security XXIII*, pages pp. 291–298, 2009.

[30] Zeinab Noorian and Mihaela Ulieru. The state of the art in trust and reputation systems : a framework for comparison. *Journal of theoretical and applied electronic commerce research*, 5(2) :97–117, 2010.

[31] Audun Jøsang, Roslan Ismail, and Colin Boyd. A survey of trust and reputation systems for online service provision. *Decision support systems*, 43(2) :618–644, 2007.

[32] Yao Wang and Julita Vassileva. Toward trust and reputation based web service selection : A survey. *International Transactions on Systems Science and Applications*, 3(2) :118–132, 2007.

[33] Songjie Wei and Jelena Mirkovic. Building reputations for internet clients. *Electronic Notes in Theoretical Computer Science*, 179(0) :17 – 30, 2007. Proceedings of the Second International Workshop on Security and Trust Management (STM 2006).

[34] Agnieszka Danek, Joana Urbano, Ana Paula Rocha, and Eugénio Oliveira. Engaging the dynamics of trust in computational trust and reputation systems. In *Agent and Multi-Agent Systems : Technologies and Applications*, pages 22–31. Springer, 2010.

[35] Ayman Tajeddine, Ayman Kayssi, Ali Chehab, and Hassan Artail. Fuzzy reputation-based trust model. *Applied soft computing*, 11(1) :345–355, 2011.

[36] Kamal K Bharadwaj and Mohammad Yahya H Al-Shamri. Fuzzy computational models for trust and reputation systems. *Electronic Commerce Research and Applications*, 8(1) :37–47, 2009.

[37] Audun Jsang and Roslan Ismail. The beta reputation system. In *Proceedings of the 15th bled electronic commerce conference*, volume 5, pages 2502–2511, 2002.

[38] Andrew Whitby, Audun Jøsang, and Jadwiga Indulska. Filtering out unfair ratings in bayesian reputation systems. In *Proc. 7th Int. Workshop on Trust in Agent Societies*, volume 6, pages 106–117, 2004.

[39] José de SP Guedes, Viviane Torres Da Silva, and Carlos JP De Lucena. A reputation model based on testimonies. In *Agent-Oriented Information Systems IV*, pages 37–52. Springer, 2008.

[40] Félix Gómez Mármol, Joao Girao, and Gregorio Martínez Pérez. Trims, a privacy-aware trust and reputation model for identity management systems. *Computer Networks*, 54(16) :2899–2912, 2010.

[41] Elena Zheleva, Aleksander Kolcz, and Lise Getoor. Trusting spam reporters : A reporter-based reputation system for email filtering. *ACM Transactions on Information Systems (TOIS)*, 27(1) :3, 2008.

[42] Nicola Mezzetti. A socially inspired reputation model. In *Public Key Infrastructure*, pages 191–204. Springer, 2004.

[43] SJ Knapskog. A metric for trusted systems. In *Proceedings of the 21st National Security Conference*, pages 16–29. Citeseer, 1998.

[44] Anna Gutowska, Andrew Sloane, and Kevan A Buckley. On desideratum for b2c e-commerce reputation systems. *Journal of computer science and technology*, 24(5) :820–832, 2009.

[45] Véronique LeGrand, Stéphane Ubéda, Joël Morêt-Bailly, Agnes Rabagny, Laurent Guihéry, and Jean-Philippe Neuville. Vers un modèle de confiance pour les objets communicants : une approche sociale-draft. 2004.

[46] D Harrison McKnight and Norman L Chervany. The meanings of trust. 1996.

[47] Juan Pablo Soto, Aurora Vizcaíno, Javier Portillo-Rodríguez, and Mario Piattini. Why should i trust in a virtual community member ? In *Groupware : Design, Implementation, and Use*, pages 126–133. Springer, 2009.

[48] Audun Jøsang, Stephen Marsh, and Simon Pope. Exploring different types of trust propagation. In *Trust management*, pages 179–192. Springer, 2006.

[49] Mojdeh Mohtashemi and Lik Mui. Evolution of indirect reciprocity by social information : the role of trust and reputation in evolution of altruism. *Journal of Theoretical Biology*, 223(4) :523–531, 2003.

[50] Viet-An Nguyen, Ee-Peng Lim, Hwee-Hoon Tan, Jing Jiang, and Aixin Sun. Do you trust to get trust ? a study of trust reciprocity behaviors and reciprocal trust prediction. In *SDM*, pages 72–83. SIAM, 2010.

[51] James C Cox. How to identify trust and reciprocity. *Games and economic behavior*, 46(2) :260–281, 2004.

[52] Debra Lauterbach, Hung Truong, Tanuj Shah, and Lada Adamic. Surfing a web of trust : Reputation and reciprocity on couchsurfing. com. In *Computational Science and Engineering, 2009. CSE'09. International Conference on*, volume 4, pages 346–353. IEEE, 2009.

[53] Caroline Nevejan and Frances Brazier. Granularity in reciprocity. *AI & society*, 27(1) :129–147, 2012.

[54] Marsh Stephen and Briggs Pamela. Examining trust, forgiveness and regret as computational concepts. *Computing with Social Trust (Part 1)*, pages pp. 9–43, 2009.

[55] Stephen Marsh, Anirban Basu, and Natasha Dwyer. Rendering unto cæsar the things that are cæsar's : Complex trust models and human understanding. In *Trust Management VI*, pages 191–200. Springer, 2012.

[56] David A. Marca and Clement L. McGowan. *SADT : Structured Analysis and Design Technique*. McGraw-Hill, Inc., New York, NY, USA, 1987.

[57] F. Lagier. Cybercriminalité : 120.000 victimes d'usurpation d'identité chaque année en france. *Le populaire du centre*, 2013.

[58] Cagri I. Guvence-Rodoper, Izak Benbasat, and Ronald T. Cenfetelli. Adoption of b2b exchanges : Effects of it-mediated website services, website functionality, benefits, and costs. *ICIS 2008 Proceedings*, 2008.

[59] Darren Filson. The impact of e-commerce strategies on firm value : Lessons from amazon.com and its early competitors. *The Journal of Business*, 77(S2) :pp. S135–S154, 2004.

[60] Yinghui Catherine Yang and Balaji Padmanabhan. Toward user patterns for online security : Observation time and online user identification. *Decision Support Systems*, 48(4) :548–558, 2010.

[61] Gustavus J Simmons. A survey of information authentication. *Proceedings of the IEEE*, 76(5) :603–620, 1988.

[62] Javier Lopez, Rolf Oppliger, and Günther Pernul. Authentication and authorization infrastructures (aais) : a comparative survey. *Computers & Security*, 23(7) :578–590, 2004.

[63] Kasper B Rasmussen, Marc Roeschlin, Ivan Martinovic, and Gene Tsudik. Authentication using pulse-response biometrics. In *Proceedings of the Network and Distributed Systems Security Symposium (NDSS)*, 2014.

[64] Josef Bigun, Julian Fiérrez-Aguilar, Javier Ortega-Garcia, and Joaquin Gonzalez-Rodriguez. Combining biometric evidence for person authentication. In *Advanced Studies in Biometrics*, pages 1–18. Springer, 2005.

[65] Myriam Abramson and David W. Aha. User authentication from web browsing behavior. *Proceedings of the Twenty-Sixth International Florida Artificial Intelligence Research Society Conference*, pages pp. 268–273, 2013.

[66] Markus Jakobsson, Elaine Shi, Philippe Golle, and Richard Chow. Implicit authentication for mobile devices. *Proceeding HotSec'09 Proceedings of the 4th USENIX conference on Hot topics in security*, pages pp. 9–9, 2009.

[67] Rui He, Man Yuan, Jianping Hu, Hong Zhang, Zhigang Kan, and Jian Ma. A novel service-oriented aaa architecture. 3 :pp. 2833–2837, 2003.

[68] Tobias Stockinger. Implicit authentication on mobile devices. *the Media Informatics Advanced Seminar on Ubiquitous Computing*, 2011.

[69] Elaine Shi, Yuan Niu, Markus Jakobsson, and Richard Chow. Implicit authentication through learning user behavior. *M. Burmester et al. (Eds.) : ISC 2010, LNCS 6531*, page pp. 99–113, 2011.

[70] PN Ali Fahmi, Elyor Kodirov, Deok-Jai Choi, Guee-Sang Lee, A Mohd Fikri Azli, and Shohel Sayeed. Implicit authentication based on ear shape biometrics using smartphone camera during a call. In *Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on*, pages 2272–2276. IEEE, 2012.

[71] Nashad Ahmed Safa, Reihaneh Safavi-Naini, and Siamak F Shahandashti. Privacy-preserving implicit authentication. In *ICT Systems Security and Privacy Protection*, pages 471–484. Springer, 2014.

[72] Alexander De Luca, Alina Hang, Frederik Brudy, Christian Lindner, and Heinrich Hussmann. Touch me once and i know it's you ! : implicit authentication based on touch screen patterns. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 987–996. ACM, 2012.

[73] Hassan Khan and Urs Hengartner. Towards application-centric implicit authentication on smartphones. In *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications*, page 10. ACM, 2014.

[74] Sausan Yazji, Xi Chen, Robert P Dick, and Peter Scheuermann. Implicit user re-authentication for mobile devices. In *Ubiquitous Intelligence and Computing*, pages 325–339. Springer, 2009.

[75] Anil K. Jain, Arun Ross, and Salil Prabhakar. An introduction to biometric recognition. *IEEE Trans. on Circuits and Systems for Video Technology*, 14 :4–20, 2004.

[76] Rodrigo de Luis-Garcia, Carlos Alberola-Lopez, Otman Aghzout, and Juan Ruiz-Alzola. Biometric identification systems. *Signal Processing*, 83(12) :2539–2557, 2003.

[77] Mickaël Fabrègue. *Extraction d'informations synthétiques à partir de données séquentielles : Application à l'évaluation de la qualité des rivières*. PhD thesis, Université de Strasbourg, France, 2014.

[78] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, pages 487–499, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc.

[79] Bernhard Ganter and Rudolf Wille. *Formal concept analysis - mathematical foundations*. Springer, 1999.

[80] Mohammed J. Zaki and Ching jui Hsiao. Charm : An efficient algorithm for closed itemset mining. pages 457–473, 2002.

[81] Helen Pinto, Jiawei Han, Jian Pei, Ke Wang, Qiming Chen, and Umeshwar Dayal. Multi-dimensional sequential pattern mining. In *Proceedings of the Tenth International Conference on Information and Knowledge Management*, CIKM '01, pages 81–88, New York, NY, USA, 2001. ACM.

[82] Gang Fang, Zukuan Wei, and Qian Yin. An algorithm of constrained spatial association rules based on binary. In *Proceedings of the 5th International Symposium on Neural Networks : Advances in Neural Networks, Part II*, ISNN '08, pages 21–29, Berlin, Heidelberg, 2008. Springer-Verlag.

[83] Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. *SIGMOD Rec.*, 29(2) :1–12, May 2000.

[84] R. Agrawal and R. Srikant. Mining sequential patterns. In *Data Engineering, 1995. Proceedings of the Eleventh International Conference on Data Engineering*, pages 3–14, Mar 1995.

[85] Jian Pei, Jiawei Han, Behzad Mortazavi-asl, Helen Pinto, Qiming Chen, Umeshwar Dayal, and Mei chun Hsu. Prefixspan : Mining sequential patterns efficiently by prefix-projected pattern growth. pages 215–224, 2001.

[86] Xifeng Yan, Jiawei Han, and Ramin Afshar. Clospan : Mining closed sequential patterns in large datasets. In *In SDM*, pages 166–177, 2003.

[87] Aí da Jiménez, Fernando Berzal, and Juan-Carlos Cubero. Frequent tree pattern mining : A survey. *Intell. Data Anal.*, 14(6) :603–622, November 2010.

[88] Akshay Deepak, David Fernández-Baca, Srikanta Tirthapura, Michael J. Sanderson, and Michelle M. Mcmahon. Evominer : Frequent subtree mining in phylogenetic databases. *Knowl. Inf. Syst.*, 41(3) :559–590, December 2014.

[89] X. Yan and Jiawei Han. gspan : graph-based substructure pattern mining. In *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*, pages 721–724, 2002.

[90] Prakash Shelokar, Arnaud Quirin, and íScar CordóN. A multiobjective evolutionary programming framework for graph-based data mining. *Inf. Sci.*, 237 :118–136, July 2013.

[91] Chuntao Jiang, Frans Coenen, and Michele Zito. A survey of frequent subgraph mining algorithms. *The Knowledge Engineering Review*, 28 :75–105, 3 2013.

[92] Mohammed J. Zaki and Ching-Jui Hsiao. Efficient algorithms for mining closed itemsets and their lattice structure. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, 17 :2005, 2005.

[93] S. Ben Yahia, T. Hamrouni, and E. Mephu Nguifo. Frequent closed itemset based algorithms : A thorough structural and analytical survey. *SIGKDD Explor. Newsl.*, 8(1) :93–104, June 2006.

[94] Karam Gouda and MohammedJ. Zaki. Genmax : An efficient algorithm for mining maximal frequent itemsets. *Data Mining and Knowledge Discovery*, 11(3) :223–242, 2005.

[95] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. *SIGMOD Rec.*, 22(2) :207–216, June 1993.

[96] Julien Rabatel. *Extraction de motifs contextuels : Enjeux et applications dans les données séquentielles.* PhD thesis, Université de Montpellier II, France, 2011.

[97] Ansaf Salleb. *Recherche de motifs fréquents pour l'extraction de règles d'association et de caractérisation.* PhD thesis, Université d'Orléans, France, 2003.

[98] Nicolas Pasquier, Yves Bastide, Rafik Taouil, and Lotfi Lakhal. Discovering frequent closed itemsets for association rules. In *Database Theory—ICDT'99*, pages 398–416. Springer, 1999.

[99] Nicolas Pasquier, Yves Bastide, Rafik Taouil, and Lotfi Lakhal. Efficient mining of association rules using closed itemset lattices. *Information systems*, 24(1) :25–46, 1999.

[100] Nicolas Pasquier, Yves Bastide, Rafik Taouil, and Lotfi Lakhal. Pruning closed itemset lattices for association rules. In *BDA'1998 international conference on Advanced Databases*, pages 177–196, 1998.

[101] Yves Bastide, Nicolas Pasquier, Rafik Taouil, Gerd Stumme, and Lotfi Lakhal. Mining minimal non-redundant association rules using frequent closed itemsets. In *Computational Logic—CL 2000*, pages 972–986. Springer, 2000.

[102] Yves Bastide, Rafik Taouil, Nicolas Pasquier, Gerd Stumme, and Lotfi Lakhal. Mining frequent patterns with counting inference. *ACM SIGKDD Explorations Newsletter*, 2(2) :66–75, 2000.

[103] Gerd Stumme, Rafik Taouil, Yves Bastide, Nicolas Pasquier, and Lotfi Lakhal. Intelligent structuring and reducing of association rules with formal concept analysis. In *KI 2001 : Advances in Artificial Intelligence*, pages 335–350. Springer, 2001.

[104] Gerd Stumme, Rafik Taouil, Yves Bastide, Nicolas Pasquier, and Lotfi Lakhal. Computing iceberg concept lattices with titanic. *Data & knowledge engineering*, 42(2) :189–222, 2002.

[105] Takeaki Uno, Tatsuya Asai, Yuzo Uchida, and Hiroki Arimura. Lcm : An efficient algorithm for enumerating frequent closed item sets. In *FIMI*, volume 90. Citeseer, 2003.

[106] Claudio Lucchese, Salvatore Orlando, and Raffaele Perego. Dci closed : A fast and memory efficient algorithm to mine frequent closed itemsets. In *FIMI*, 2004.

[107] Sadok BEN YAHIA and Engelbert MEPHU NGUIFO. Approches d'extraction de règles d'association basées sur la correspondance de galois. *Ingénierie des systèmes d'information*, 9(3-4) :23–55, 2004.

[108] Rudolf Wille. Concept lattices and conceptual knowledge systems. *Computers & mathematics with applications*, 23(6) :493–515, 1992.

[109] Nicolas Pasquier. *Data mining : algorithmes d'extraction et de reduction des regles d'association dans les bases de donnees*. PhD thesis, 2000. Thèse de

doctorat dirigée par Lakhal, Lotfi Sciences et techniques Clermont Ferrand 2 2000.

[110] Jianyong Wang, Jiawei Han, and Jian Pei. Closet+ : Searching for the best strategies for mining frequent closed itemsets. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 236–245. ACM, 2003.

[111] Sergey Brin, Rajeev Motwani, and Craig Silverstein. Beyond market baskets : Generalizing association rules to correlations. *SIGMOD Rec.*, 26(2) :265–276, June 1997.

[112] Hong Cheng, Xifeng Yan, and Philip S. Yu. Direct discriminative pattern mining for effective classification. In *In ICDE*, 2008.

[113] Gerard Salton. *Automatic text processing : The transformation, analysis and retrieval of information by computer*. Addison Wesley, 1989.

[114] SO Kuznetsov. Stability as an estimate of the degree of substantiation of hypotheses derived on the basis of operational similarity. *Automatic Documentation and Mathematical Linguistics*, 1990.

[115] Aleksey Buzmakov, Sergei O. Kuznetsov, and Amedeo Napoli. Is concept stability a measure for pattern selection ? *Procedia Computer Science*, 31 :918 – 927, 2014. 2nd International Conference on Information Technology and Quantitative Management, {ITQM} 2014.

[116] Mikhail Klimushkin, Sergei Obiedkov, and Camille Roth. Approaches to the selection of relevant concepts in the case of noisy data. In *Formal concept analysis*, pages 255–266. Springer, 2010.

[117] Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining : Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, 2011.

[118] M. J. Zaki and M. J. Wagner. Data mining and analysis : Fundamental concepts and algorithms. page 624.

[119] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1) :81–106, 1986.

[120] Bing Liu, Wynne Hsu, and Yiming Ma. Integrating classification and association rule mining. pages 80–86, 1998.

[121] Wenmin Li, Jiawei Han, and Jian Pei. Cmar : Accurate and efficient classification based on multiple class-association rules. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, ICDM '01, pages 369–376, Washington, DC, USA, 2001. IEEE Computer Society.

[122] Olivier Raynaud, Yannick Loiseau, Diyé Dia, Olivier Coupelon, and Fabien Labernia. Using closed itemsets for implicit user authentication in web browsing. In *CLA 2014, Proceedings of the Eleventh International Conference on Concept Lattices and Their Applications*, 2014.

[123] Ricco Rakotomalala and Stéphane Lallich. Construction d'arbres de décision par optimisation. *Revue d'Intelligence Artificielle*, 16(6) :685–703, 2002.

[124] James Dougherty, Ron Kohavi, Mehran Sahami, et al. Supervised and unsupervised discretization of continuous features. In *Machine learning : proceedings of the twelfth international conference*, volume 12, pages 194–202, 1995.

[125] William W Cohen. Learning trees an rules with set-valued features. 1996.

[126] J. R. Quinlan and R. M. Cameron-jones. Foil : A midterm report. In *In Proceedings of the European Conference on Machine Learning*, pages 3–20. Springer-Verlag, 1993.

[127] Y. Okada, T. Tada, K. Fukuta, and T. Nagashima. Audio classification based on a closed itemset mining algorithm. In *Computer Information Systems and Industrial Management Applications (CISIM), 2010 International Conference on*, pages 60–65, Oct 2010.

[128] Eyke Hüllermeier and Johannes Fürnkranz. Comparison of ranking procedures in pairwise preference learning. In *Proceedings of the 10th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU-04), Perugia, Italy*, 2004.

[129] Vryniotis Vasilis. JAVA implementation of Multinomial Naive Bayes Text Classifier.

[130] Roberto J Bayardo Jr, Rakesh Agrawal, and Dimitrios Gunopulos. Constraint-based rule mining in large, dense databases. In *Data Engineering, 1999. Proceedings., 15th International Conference on*, pages 188–197. IEEE, 1999.

[131] L. Szathmary. *Symbolic Data Mining Methods with the Coron Platform*. PhD Thesis in Computer Science, University Henri Poincaré – Nancy 1, France, Nov 2006.

[132] Leo Breiman. Random forests. *Machine learning*, 45(1) :5–32, 2001.

[133] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3) :pp. 273–297, 1995.

[134] R Agrawal, H Mannila, R Srikant, H Toivonen, and Verkamo A.I. Fast discovery of association rules. *Advances in Knowledge Discovery and Data Mining, Ch.12, AAAI Press*, 1996.

[135] Jose Borges and Mark Levene. Data mining of user navigation patterns. In *Web usage analysis and user profiling*, pages 92–112. Springer, 2000.

[136] Lara D Catledge and James E Pitkow. Characterizing browsing strategies in the world-wide web. *Computer Networks and ISDN systems*, 27(6) :1065–1073, 1995.

[137] Eugene Charniak. *Statistical language learning*. MIT press, 1996.

[138] Doru Tanasa and Brigitte Trousse. Le prétraitement des fichiers logs web dans le "web usage mining" multi-sites. *Journées Francophones de la Toile (JFT'2003)*, pages 113–122, 2003.

[139] Bradley Efron and B Efron. *The jackknife, the bootstrap and other resampling plans*, volume 38. SIAM, 1982.