École nationale supérieure d'informatique et d'analyse des systèmes

PROJET DE FIN D'ANNÉE

Etude et implémentation de la confiance dans un réseau social

---

Soutenu par :

GUEZRI AMINE **OURAHIM AISSA**

Encadré par : Professeur D. BOUZIDI

Examiné par : Professeur R. AJHOUN

Remerciement

On tient à remercier particulièrement Dr. BOUZIDI pour la confiance qu'il nous accordés, pour son aide et ses précieux conseils apportés durant la réalisation de notre projet.

Nos remerciements également Dr. AJHOUN pour avoir accepté de juger notre travaille.

Résumé

Les réseaux sociaux offrent la possibilité de partager des informations entre les différents utilisateurs. Parmi les données qui transitent dans le réseau, on compte aussi ceux issues des vies privées. Avec la croissance de la quantité de ces données personnelles, et l'existence potentiel d'utilisateurs malveillant, la protection de la vie privée d'un utilisateur devient une préoccupation majeure.

Dans les réseaux sociaux actuel, il est très difficile de détecter un utilisateur malveillant. Une des solutions qui ont été proposées, c'est de concevoir des mécanismes qui vont permettre à chaque utilisateur de trouver des pairs de confiance avec lesquels il peut partager ses données sans se soucier des attaques tels que la divulgation et l'altération des données ainsi que l'usurpation des identités, en assurant par la même occasion les services de sécurité : Confidentialité, Intégrité et Authenticité.

Avec ce mécanisme, on peut échanger nos données en toute sécurité et de manière ouverte. Ces mécanismes sont en fait des algorithmes de calcul de confiance.

Dans ce projet, on va étudier des algorithmes de calcul de confiance au sein des réseaux sociaux, et implémenter les trois algorithmes à savoir TidalTrust, MoleTrust et MaxTrust tout en les comparants.

Mots-clés : Confiance, vie privée, sécurité, réseaux sociaux, TidalTrust, MaxTrust, MoleTrust.

Abstract

Social networks provide the ability to share information between users. Among the data that pass through the network, there are also those from private lives. With the growth in the amount of this personal data, and the possibility of a malicious user, protecting the privacy of a user becomes a major concern.

In today's social networks, it is very difficult to detect a malicious user. One of the solutions that have been implemented, that is to say, that can allow users to find others with trusted data, such as identity theft and secure at the same time , Security Services: Confidentiality, Integrity and Authenticity.

With this mechanism, we can exchange our data safely and openly. These mechanisms are in fact the algorithm for calculating confidence.

In this project, we will study algorithms for calculating trust in the network of social networks, and implement the three algorithms namely TidalTrust, MoleTrust and MaxTrust while comparing.

Keywords: Trust, Privacy, Security, Social Networks, TidalTrust, MaxTrust, MoleTrust.

Listes des figures

Table des matières

Introduction générale

« L'homme est un être sociable ; la nature l'a fait pour vivre avec ses semblables. -Aristote-». De cette citation, on peut affirmer que les interactions entre les êtres humains sont d'une priorité maximale. Nos interactions sont limitées par les relations nouées et les expérience vécus. Ce mélange entre interactions et relations sociales engendre la formation des groupes ou bien de réseau social.

De nos jours, les nouvelles technologies jouent un rôle primordial dans notre vie quotidienne. Elle nous offre la possibilité de communiquer et d'échanger les informations de manière facile et rapide, ce qui engendre, par analogie, la création des communautés en ligne, appelé cette-fois ci par Réseau Social en ligne.

Les réseaux sociaux en ligne jouent le rôle d'un outil de partage d'informations entre les différents utilisateurs. Les grandes quantités d'information, incluant les données privées, et qui transitent dans les réseaux sociaux relèvent beaucoup de problèmes au niveau de la sécurité de ces données.

Dans un réseau social, il a des milliers d'utilisateurs. Parmi eux, il a bien évidemment les malveillants. La question que chaque utilisateur peut se poser est : Avec qui pourrai-je partager mes informations et mes données privées ? En d'autres mots : A qui pourrai-je faire **confiance** ? Pour répondre à ces questions, il nous faut un paramètre ou un indicateur pour identifier les gents en lesquelles on peut faire confiance. Avec un

indicateur pareil, l'utilisateur peut partager ces données privées sans se soucier de leurs divulgations ou bien utilisations frauduleuses…

Cet indicateur qu'est la confiance peut être calculé avec plusieurs algorithmes. Il existe deux types d'algorithmes de calcul de confiance : Ceux qui se basent directement sur l'architecture du réseau (MoleTrust et Tidal Trust) et ceux qui ont recours à une transformation du réseau (réseaux de flots, énergies ou résistif) ; on cite comme exemple (FloodTrust, Advogato, RN-Trust …). Chaque algorithme possède des avantages et des inconvénients. Dans ce projet, on va plutôt se focaliser sur les algorithmes appartenant à la première catégorie puisqu'ils ont une complexité minimale.

Les algorithmes qui se basent sur l'architecture du réseau social ont un inconvénient principal : C'est la manière d'explorer les chemins (ils prennent les plus court chemins) On va alors étudier une version meilleure de ces algorithmes qui à la fois possède une complexité minimal O(n), et qui propose une nouvelle manière d'exploration des chemins. Pour évaluer les différents algorithmes, nous allons faire une étude comparative. Le meilleur algorithme c'est celui qui fait le taux minimum d'erreur (ie qui a une précision maximal) . Pour pouvoir comparer, il nous faut une référence, on va utiliser alors un jeu de donnée (DataSet) qui s'appelle « Residence Hall ». On va ensuite implémenter une interface graphique englobant les trois algorithmes pour faciliter l'étude comparatif. Cette interface va nous offrir par la même occasion la possibilité de visionner les graphes qui traduisent les liaisons entre les différents utilisateurs (Nœuds).

Ce travail est structuré en quatre chapitre, dont voici une brève présentation :
Dans le premier chapitre, nous présentons le contexte général du projet. On va poser notre problématique, définir les objectifs de notre projet, et décrire la démarche qu'on a suivi.

Dans le deuxième chapitre, nous définirons les réseaux sociaux et leurs architectures. Nous étudions aussi des mécanismes de contrôle d'accès, qui assurent la politique de sécurité définis sur un réseau social.

On va par la suite définir le concept de « vie privée », tout en présentant les menaces potentielles qui peuvent y touchés. On va décrire par la suite des solutions existantes qui assurent la sécurité de nos données personnelles contre tout types d'attaque (divulgation ou vente des données …)

Les modèles d'évaluation de la confiance feront l'objet du quatrième chapitre. On va étudier de manière générale les différents types de ses modèles : à savoir les modèles qui sont basés sur l'architecture des réseaux sociaux, ou sur l'interaction, et aussi ceux qui regroupe les deux (Hybride).

Une étude sur les deux types d'algorithmes d'évaluation de confiance (local et globale) fera l'objet du troisième chapitre. Nous allons citer les avantages et les inconvénients de chacun d'eux.

Le dernier chapitre est dédié pour la partie réalisation. On va implémenter les trois algorithmes : TidalTrust, Mole Trust et MaxTrust tout en les comparants en se basant sur différents critères (Erreur absolue moyenne, Erreur quadratique moyenne, Précision et couverture)

Enfin, nous clôturons notre travail par une conclusion générale, et quelques perspectives.

Chapitre 1 : Contexte générale du projet

Introduction

Ce premier chapitre décrit l'environnement dans lequel le projet a été initié : les réseaux sociaux. Ensuite, il présente les problèmes de confiance produite de la nature ouverte de ses réseaux. Enfin, il expose les objectifs pour remédier à ce problème.

Périmètre de projet

En raison de la nature ouverte des réseaux sociaux, les utilisateurs souffrent d'une insuffisance au niveau de la sécurité de leurs données privées. Ainsi l'une des premières questions à résoudre pour l'amélioration des réseaux sociaux, serait d'assurer un minimum de sécurité. Cela revient à assurer une confiance entre chaque paire d'utilisateurs en interaction.

Dans ces conditions, notre projet portera sur l'implémentation d'un modèle de confiance et de réputation dans un réseau sociale et comparer ses résultats avec d'autres modèles déjà existants.

Problématique

Dans les réseaux sociaux on échange des données, on partage des fichiers … Le problème est ici : C'est la *confidentialité* des données : voir circuler sur le Net des données que nous aurions souhaité limiter à la seule *sphère privée*.

Une fois rendues publiques, ces informations peuvent malheureusement nuire a notre carrière professionnelle, a la sérénité de notre famille, voire mettre en péril notre sécurité et notre identité numérique.

Objectif de projet

L'objectif de notre projet est la mise en place d'un outil pour intégrer les algorithmes de calcul de confiance dans les Réseaux sociaux.

La démarche suivie

Nous avons divisé notre projet en trois grandes parties : Tout d'abord le contexte général du projet, ensuite les études des algorithmes et enfin la partie implémentation. La première partie est consacré principalement à l'analyse et la compréhension du projet. La deuxième partie consiste en l'étude et la compréhension des algorithmes d'évaluation de confiance. La dernière parie porte sur l'implémentation de quelques algorithmes de notre choix, tout en les comparants en se basant sur des paramètres bien précis.

Notre temps a été répartis comme suit pour bien répartir les charges et pouvoir gérer notre temps de manière optimale.

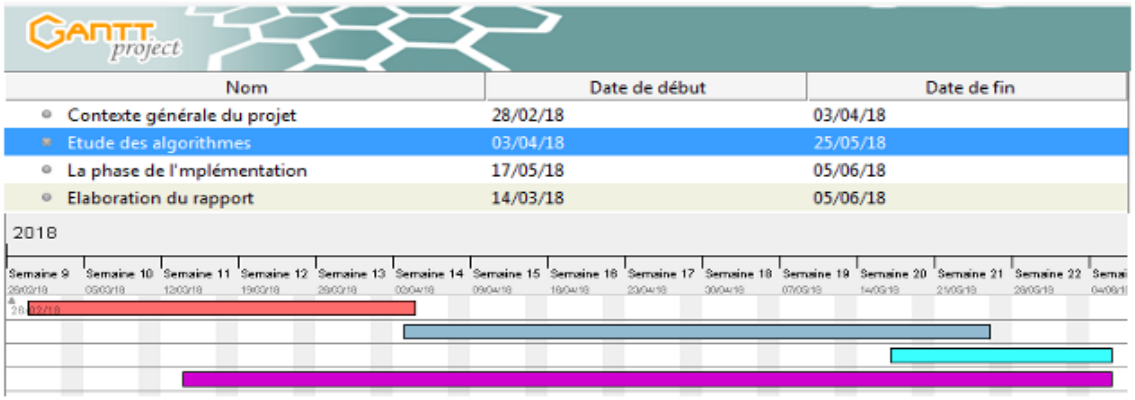

Figure 1 : Diagramme de Gantt.

Conclusion

Dans ce chapitre, nous avons introduit le problème de gestion de confiance, l'objectif générale de notre projet ainsi que la démarche qu'on a suivi. Dans le chapitre suivant on va voir de manière détaillé la relation entre la confiance et les réseaux sociaux.

Chapitre 2 : la confiance dans les Réseaux Sociaux

Introduction

Les interactions de réseaux sociaux sont devenues indispensables dans notre vie quotidienne et cela résulte principalement au développement rapide que connaît le monde de la digitale. Dans le cadre de la collecte de l'information et du Big data, ces réseaux sociaux ont tendances de créer des graphes sociales composé des utilisateurs et leurs amitiés. Mais, il a des informations plus importantes à extraire et qui sont plus riche que des simples relations d'amitiés : *C'est la confiance*.

Les connaissances sur la confiance dans les réseaux sociaux peuvent également être utilisées dans *les systèmes de recommandation*, ce qui les rendent encore plus intéressantes pour les chercheurs : En se basant sur ce système de recommandation, on peut économiser le temps et les efforts pour sélectionner des profils pour un entretien d'embauche par exemple.

Dans cette partie, on va voir les différents types de réseaux sociaux ainsi que leurs différentes architectures, ensuite nous allons voir de manière détaillé l'importance de la vie privée. Puis on va mettre l'accent sur la confiance : connaître les types et les propriétés de confiance. On va finaliser notre travaille par une étude détaillée des modèles d'évaluation de confiance

1. Les réseaux sociaux

1.1. Qu'est-ce qu'un réseau social ?

Les réseaux sociaux n'ont pas un *modèle cohérent et reconnu* alors nombreuses définitions ont été donnés.

En gros, les réseaux sociaux sur Internet ont été définis des services basés web qui permettent aux individus de construire un profil public ou semi-public dans un système fermé, de créer une liste utilisateurs avec lesquels ils partagent une connexion (lien entre deux utilisateurs), de visualiser et d'explorer leur liste de contacts (ensemble de

connexions) et celles des autres au sein du système. La nature et la nomenclature de ces connexions peuvent varier d'un site à un autre.

Aujourd'hui un réseau social c'est une site Web offrant plusieurs fonctionnalités aux adhérents. On cite quelques fonctionnalités qui constituent les piliers de la notion de *réseau social* :

- Être en contact avec d'autres utilisateurs.
- Pouvoir Interagir avec le contenu publié par les autres utilisateurs.
- Contrôler l'accès à notre propre contenu.

On donne l'exemple de Facebook :

- On est toujours en contact avec les autres utilisateurs (Poke, Demande d'ajout …)
- On interagit avec le contenu publié (les pouces bleus, les commentaires…)
- Contrôler l'accès avec les options 'Edit Post Privacy'

1.2. Classification des réseaux sociaux

On peut classifier les réseaux sociaux selon six critères :

- Les Publications
- Réseautage ou Networking (créer des relations professionnelles).
- La collaboration (communication collaborative)
- La discussion (Forum)
- Le chat
- Le partage

Figure 2 : Classification des réseaux sociaux

1.3. Les caractéristiques des réseaux sociaux

- Points positifs

Les principales caractéristiques des réseaux sociaux peuvent être résumé en trois éléments : Le caractère communautaire, le modèle participatif, et la personnalisation de l'information.

➢ *Le caractère communautaire :*

Le regroupement d'individus par centres d'intérêts afin d'interagir entre eux.

❖ Newsgroups/usenet (forums de discussion par mails), listes de diffusion, chats / IRC.

❖ Regroupement autour d'un site web (ou « réseaux de sites » : rings) de personnes partageant un même centre d'intérêt, avec un forum éventuellement.

❖ Commentaires sur sites / blogs (cas de Rue89 où les lecteurs peuvent devenir auteurs).

- *Le modèle participatif :*

Faciliter le partage d'information sur les réseaux sociaux (« Web 2,0 » a facilité aux utilisateurs de mettre en ligne du contenu textes, image ou vidéos).

- Points négatifs

Les critiques sur les réseaux sociaux tournent autour de plusieurs arguments mettant en cause les fournisseurs de ces services sur :

- L'exploitation des données personnelles à des fins commerciales ou politique (Exemple de Facebook : Vente des informations privées à des fins politique, voir la compagne électorale du Président des états unis d 'Amérique).

- La modification des systèmes sans avertissement et notamment l'installation de fonctionnalités 'par défaut' qui suscitent une surveillance de la part des utilisateurs.

1.4. Les architectures des réseaux sociaux (RS)

La question de l'architecture des réseaux sociaux est très importante : On peut critiquer le fait qu'un réseau social soit « Un système centralisé » comme le cas de Facebook, ou bien glorifier celui qui soit « Pair à Pair » comme le cas de BitTorrent.

Ce qu'on veut dire par cela, l'architecture est très importante car elle peut donner un préjugé sur les réseaux sociaux.

Dans cette partie, on va voir toutes les architectures existantes.

1.4.1. Les architectures centralisées

Cette architecture repose sur le principe d'autorité centrale. Elle est représenté par le système Client-Serveur : Existence d'un serveur central auquel se connectent tous les clients.

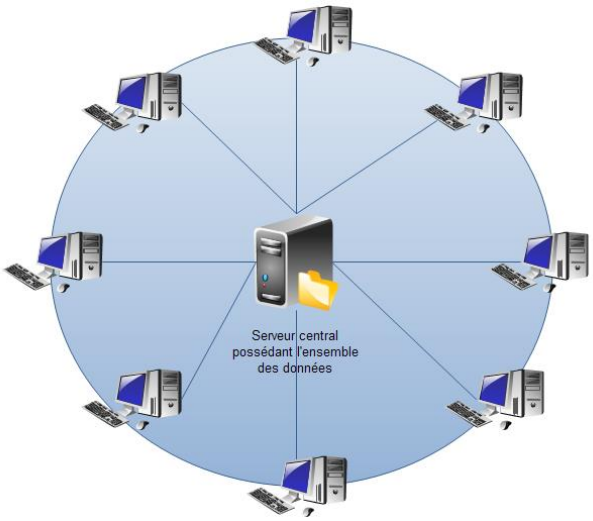

Figure 3 : Architecture centralisée

L'avantage de cette architecture est qu'elle est structurée, partagé donc rapide au niveau de la recherche des ressources partagées. Or, elle relève quelques problèmes de sécurité (Pour détruire le réseau, il suffit d'attaquer la source qui est le serveur centrale) et de robustesse (Le serveur est le seul maillon faible, étant donné que tout le réseau est architecturé atour de lui).

Dans le contexte des réseaux sociaux, l'idée dominante c'est qu'un système centralisé est mauvais, puisque on est à la merci d'une entité unique, individu, entreprise ou gouvernement, qui peut décider du jour au lendemain de nous couper l'accès par exemple.

Vu cette idée dominante, beaucoup de gens vont chercher à faire un équivalent décentralisé de ces systèmes. Passons alors, pour voir de manière plus détailler c'est quoi un système décentralisé.

1.4.2. Les architectures décentralisées

Les systèmes décentralisés sont des systèmes répartis de nœuds interconnectés. Internet est un exemple de réseau distribué puisqu'il ne possède aucun nœud central. Les architectures distribuées reposent sur la possibilité d'utiliser des objets qui s'exécutent sur des machines réparties sur le réseau et communiquent par messages au travers du réseau. Un exemple de réussite de cette architecture est le modèle Peer-to-Peer. Ce modèle est appliqué pour le partage de fichier.

On trouve comme avantage : Résistance aux pannes (Réplication des ressources), Extensibilité (On peut passer de 100 à 10 000 nœuds sans problème). Et comme inconvénients : Régulation (Application difficile des lois : contenu immoral, Droit d'auteurs...), *QoS* ou qualité de service (Existence d'une ligne peu fiable ou un débit peu élevé).

1.4.3. Les architectures hybrides

La différence entre l'architecture hybride et celle purement décentralisée est que la première maintient toujours l'utilisation d'un élément central. Elle se distingue également de l'architecture *client/serveur* par le fait que le partage des ressources est maintenu par les pairs. D'où la définition : un réseau pair à pair est classé dans les réseaux pairs à pair hybride si l'existence d'une entité centrale est nécessaire pour fournir une partie de service offert par le système.

1.5. Architecture des réseaux sociaux en ligne

L'architecture de référence d'un RSL est donnée par la figure 3. Elle est composée de plusieurs couches :

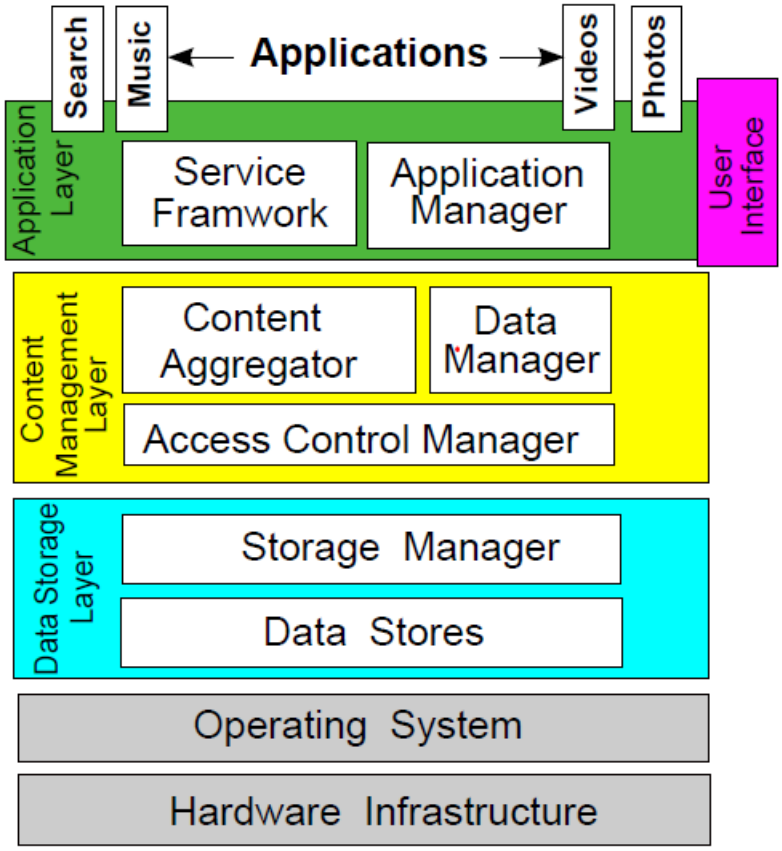

Figure 4 : Architecture des réseaux sociaux en ligne.

Le composant qui nous intéresse ici en terme de confiance c'est Access Control Manager, ce dernier met en place des modèles de contrôles d'accès afin de limiter l'accès au ressources et les protégées contre n'import quelle types d'attaques prévenant des utilisateurs malveillants.

Il existe plusieurs modèles qui assure ce service, à savoir RBAC, TCAC, DAC, MAC etc.

Comme exemple on va étudier TCAC car ce dernier met en œuvre une seuil de

confiance a vérifié on cas d'accès aux ressources .

schéma illustratif :

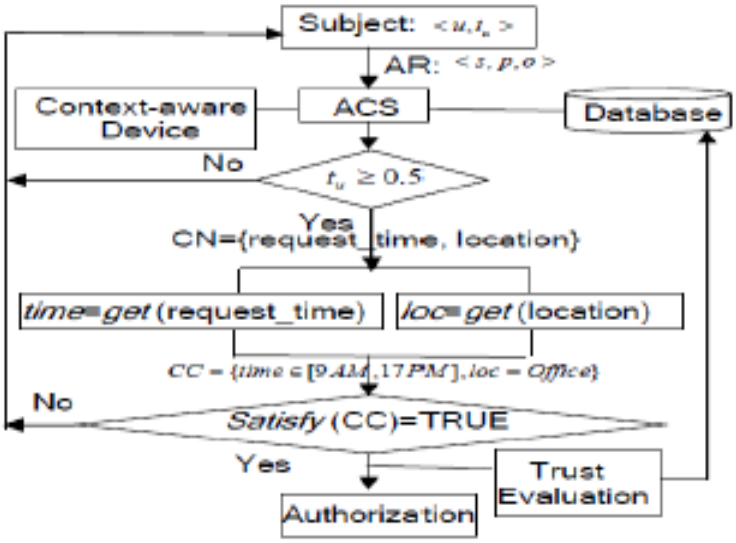

Figure 5 : Schéma illustratif du modèle TCAC.

Notre sujet <u,t> c'est le demandeur d'accès qui a t comme valeur de confiance . Dans une session s, cet utilisateur veut effectuer une opération p sur un objet o.

ACS interroge premièrement le Context-aware Device, si les informations de contexte y compris le temps de la demande d'accès et l'emplacement satisfont la contrainte (heure nette 9 AM et 10 AM , loc=office) par exemple et 'tu' est supérieur au seuil définie par le système, alors l'utilisateur 'u' sera affecté au rôle R .

Enfin, on fait une évaluation de confiance en fonction du comportement de l'utilisateur pendant la session et en fait un 'UPDATE' dans la base de données.

2. la vie privée

Introduction

La vie privée est la capacité, pour une personne ou pour un groupe de personnes, de s'isoler afin de protéger ses intérêts. Les limites de la vie privée ainsi que ce qui est considéré comme privé diffèrent selon les groupes, les cultures et les individus, selon les coutumes et les traditions bien qu'il existe toujours un certain tronc commun.

La vie privée peut parfois s'apparenter à l'anonymat et à la volonté de rester hors de la vie publique. Quand quelque chose est dit « privé » pour une personne, cela signifie que généralement qu'à cette chose sont rattachés des sentiments spéciaux et personnels. Le degré de privatisation de l'information dépend donc de la façon dont le public pourrait la recevoir, ce qui diffère selon les endroits et à travers le temps. La vie privée peut être vue sous un aspect sécuritaire.

Les données privées sont définies comme toutes données impliquant l'identification ou la quasi-identification (Géo-localisation) des utilisateurs et cela en utilisant soit leur adresse IP, identité, profils d'usage.

Cette identification **peut être exploité** par des personnes **malveillantes** (par exemple : Accès illicite, vente de données, ou exploitation commerciale sans consentement…). Cette exploitation illégale des données privée souligne une nécessité d'existence des lois sanctionnant les invasions dans la vie privée par le gouvernement, les corporations ou les individus.

2.1. Les lois de protection de la vie privée

- *Droit international*

La vie privée est protégée au niveau international par l'article 12 de la déclaration

universelle des droits de l'homme de 1948.

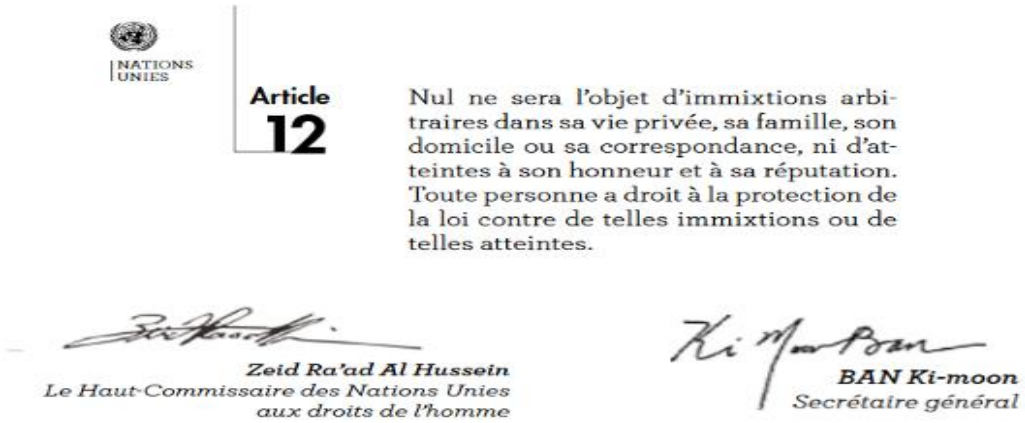

Figure 6 : Loi international de droit privée.

- *Conseil de l'Europe*

Le droit à la vie privée est garanti par l'article 8 de la convention européenne des droits de l'homme, mais doit souvent être équilibré avec le droit à la liberté d'expression, garanti par l'article 10.

- *OCDE : Organisation pour la Coopération et le Développement Économiques*

L'OCDE qui a posé les piliers des lois de la vie privée. Ils sont nommés « Pratiques d'information équitables » et sont du nombre de huit. Dans cette partie, on va citer quelques-uns.

➢ La limitation de la collecte de données :

La collecte de données doit être restreinte et obtenue avec le consentement du propriétaire. En collectant un très grand nombre d'information sur une personne, on va finir par l'identifier.

➢ La limitation de l'utilisation :

Les données personnelles ne doivent pas être divulgués, exploité à des fins autres que celle spécifiés.

2.2. La vie privée dans les réseaux sociaux

La génération d'aujourd'hui ont tendance de communiquer et de nouer des relations d'amitié en ligne ce qui engendre des grandes quantités d'informations transitant entre les différents utilisateurs des réseaux sociaux à savoir des informations à caractère personnel.

La protection de notre vie privée est loin d'être satisfaite par les mécanismes de contrôle d'accès qui existent aujourd'hui : Ils sont nécessaires mais non suffisants. Avec la nature ouverte des réseaux sociaux, il a toujours une probabilité que les informations échangées soient interceptées et exploitées sans l'autorisation de leurs propriétaires.

L'exploitation non autorisée de nos informations peut venir même de la part des propriétaires de ces réseaux sociaux : Vente d'information et collecte d'information pour des raisons politiques.

La vie privée dans les réseaux sociaux est alors en menaces. C'est ce qu'on va voir de manière détaillée dans la partie suivante.

2.2.1 Les menaces à la vie privée

Il a plusieurs menaces à la vie privée, on cite quelques-uns :

- *Accès non autorisé à l'information (Unauthorized Access to Information) :*

Lorsque des milliers de documents internes de la multinationale Sony ont été rendus publics en 2014, la faute a initialement été attribuée à des pirates nord-coréens. Peu à peu toutefois, des indications ont pointé vers la participation d'un ancien employé de Sony frustré, qui aurait aidé les pirates dans leur cyberattaque.

Ce sont alors des attaques lancées par des attaquants externes ou des utilisateurs interne pour avoir l'accès aux informations internes.

- *La vente de données (Le marché du Data-Purchasing) :*

Le principe est simple : via une plateforme assez intuitive pour être accessible au plus grand nombre et après avoir créé son compte utilisateur, l'internaute est amené à compléter questionnaires ou missions qu'il pourra ensuite échanger contre un bien, des services ou même de l'argent. Dans ce cas au moins, nous savons qu'on a remplis un formulaire et il a une possibilité que nos informations soient vendues : Nous pouvons alors donner des informations invalides.

Le cas critique c'est la collecte d'informations directement du réseau social dans le savoir. Avec cette méta-data, on peut faire des conclusions importantes et classé les gens selon l'ordre : Pouvoir d'achat, influence politique … et vendre ces informations sans le consentement explicite des utilisateurs.

- *La révélation des informations personnelles :*

La révélation des informations à caractère personnelle sans l'approbation des propriétaires. Des gens appartenant au cercle d'amitié peuvent révélés nos informations. C'est un grand problème qui n'a pas de solution : La sécurité se fait des deux parts ; Au niveau des fournisseur d'accès mais aussi dans le niveau des utilisateurs.

2.2.2 Les solutions existantes

Il existe beaucoup de solutions pour garantir la protection de la vie privée et qui dépendent par la même occasion des architectures de ces réseaux sociaux, on peut citer :

- **Les solutions centralisées**

La plupart des RS connus, tels que Facebook, LinkedIn, Twitter ou Google+ , sont des RS centralises dans lesquels les données des utilisateurs sont sous le contrôle d'une entité centrale qui les stocke sur ces serveurs . La majorité des RS sont basées

sur une architecture centralisée, dans laquelle les données des utilisateurs sont stockées sur un serveur contrôle par les fournisseurs des RS.

- **Les solutions hybride**

Ceux sont des solutions basées sur l'observation du fonctionnement de certains RS sur de fausses données. La plupart des solutions sont implémentées comme plugins Firefox. Ce type de solutions était initialement introduit par NOYB (qui signifie Noof Of Your Bisiness), les informations personnelles de l'utilisateurs sont d'abord cryptées, ensuite le texte chiffre est remplace par un pseudo sélectionné d'une manière aléatoire a partie d'un dictionnaire public afin de le faire rassembler a des données légitimes. Le service en ligne peut fonctionner sur les données cryptées, mais seuls les utilisateurs autorises peuvent décoder et décrypter le résultat. Malheureusement, cette solution peut seulement crypter des petits textes telles que les informations personnelles du profil et ne permet pas de crypter des textes longs.

3. Confiance

3.1. Définition de la confiance

Tout d'abord, l'une des composantes majeures de l'interaction humaine est la confiance. Pour cela, il est essentiel de chercher comment définir ce concept de tel importance.

Malheureusement, on ne peut pas donner une définition d'une part unique et d'autre part qui englobe tous les aspects de la confiance. L'une des plusieurs définition est : C'est une mesure ou une évaluation 'sur une échelle de 10 par exemple' qu'on associe à une entité ; cela nous donne une idée sur le comportement ou l'attitude de cette entité,

malgré le manque de capacité de surveiller ou de contrôler l'environnement dans lequel elle opère.

En psychologie sociale et en sociologie, la confiance est une hypothèse faite sur le futur comportement d'autrui. Il s'agit d'une conviction selon laquelle une personne serait capable d'agir d'une certaine manière face à une situation donnée.

La confiance désigne, d'autre part, la familiarité au sein des relations (du traitement entre les personnes lorsqu'elles sont à l'aise les unes avec les autres). Elle suppose une suspension, du moins temporaire, de l'incertitude par rapport aux actions des autres. Lorsqu'un individu fait confiance à un autre, il est certain de pouvoir prédire ses actions et ses comportements. La confiance rend donc les relations sociales plus simples.

3.2. L'importance de la confiance dans les réseaux sociaux

Le concept de FOAF (l'amie de mon amie est mon amie) est répandu dans plusieurs de réseaux sociaux, l'hypothèse derrière ce concept est que la relation amitié , qui est la base de chaque relation , est transitive. Ainsi, les relations de FOAF impliquent indirectement que la confiance est &également transitive dans les réseaux sociaux. La confiance est propagatrice, mais pas transitive. Par conséquent, il y a un risque aux informations confidentielles des membres dans tels réseaux sociaux, la confiance entre les utilisateurs est la vraie source de puissance et de développement pour n'importe quelle communauté.

3.3. Les Types de Confiance

La confiance a de nombreux aspects différents. Quand nous parlons d'aspects De confiance, nous considérons une perspective à partir de laquelle nous regardons cette

confiance. Cette perspective donne souvent une sémantique différente à la confiance

- *Aspect calculatoire de la confiance :*

Définit la confiance comme le résultat d'un calcul au nom du membre pour maximiser ses enjeux dans l'interaction.

- *Aspect relationnel de la confiance :*

Cet aspect de la confiance est le résultat d'interactions répétées entre les membres. En informatique, cet aspect de confiance s'appelle confiance directe, confiance basée sur des interactions directes entre deux parties.

- *Aspect émotionnel de la confiance :*

Définit la confiance comme la sécurité et le confort en s'appuyant sur l'autre. En psychologie, c'est le résultat de relations interpersonnelles directes entre membre et l'autre.

- *Aspect institutionnel de la confiance :*

Définit la confiance à la suite d'une institution fournissant un environnement qui encourage la coopération entre les membres et pénalise les mauvais comportements.

- *Aspect attitudinale de la confiance :*

L'aspect attitudinale de la confiance reconnaît que, au cours de leur vie, les gens développent des attentes généralisées quant à la fiabilité des autres.

3.4. Les propriétés de Confiance

La confiance peut avoir différentes propriétés car elle peut varier dans le temps. Les propriétés de la confiance ont une grande influence sur la détermination du type de

confiance recherché et modélisé. On va citer les plus importantes de ces propriétés :

- ***Contexte spécifique***

Si on fait confiance à une personne A, on le fait cela dans le cadre d'un contexte. Supposons que A est astronaute, alors on va lui faire confiance en ses qualités d'astronaute et on ne va pas lui faire confiance en tant que médecin par exemple.

- ***Dynamique***

La confiance n'est pas stable : Elle varie dans le temps et les nouvelles expériences : Avec les expériences acquises dans la vie, notre vision de confiance peut changer carrément et cela implique peut impliquer par exemple une restriction du domaine de confiance.

- ***La transitivité***

On peut décrire cette propriété par un exemple : Si A fait confiance à B, et B fait confiance à C, alors A fait confiance à C. Cela est très important puisque A ne connaît pas C, mais de même il fait confiance.

Avec le principe de transitivité, on construit une chaîne de confiance.

- ***Exclusivité***

Le principe de transitivité nous donne la possibilité de faire confiance à une personne qu'on connaît pas, mais il ne dit pas comment se comporter lorsque plusieurs chaînes de confiance recommandent un nombre différent de gens de confiance. L'idée c'est composé toutes les informations provenant des différentes chaînes de confiance pour faire des conclusions bien précise.

- ***Subjectivité***

La confiance est un concept subjectif. Moi personne A, je fais confiance à plusieurs

personnes B, C, D. Mais ma confiance est subjective : Ce n'est pas le même degré de confiance (je fais confiance à B plus que C), la question qui se pose ici alors est comment avoir une précision au niveau de la chaîne de confiance si on connaît pas l'ordre de préférence de chacun. Heureusement, que nous calculons le degré de confiance à partir des interactions entre les utilisateurs et pas de leurs visions subjectives consécutives.

Malgré cela, on ajoute que des études sont faites ; il a une forte corrélation et correspondance entre la confiance calculée à partir des interactions entre les utilisateurs et les préférences subjectives de chaque utilisateur envers les autres.

- ***Asymétrique :***

Supposons A et B deux individus. A peut faire confiance à B plus que B ne le fait à A. Cela résulte de la diversification des opinions personnels (par exemple une expérience vécue qui a changé nos perceptions et nos idées sur la confiance).

- ***Progressive :***

Si la confiance est faible entre les utilisateurs, cela est traduit bien évidemment par une faible interaction entre eux, et ce qui induit à une confiance encore plus faible qu'auparavant.

- ***Sensible :***

Par nature, la confiance prend beaucoup de temps à s'établir. Mais un seul événement peut avoir un très impact même la destruction de cette confiance.

3.5. La représentation de la Confiance.

La confiance peut être représenté selon deux approches : probabiliste et progressive. Les approches probabilistes sont les plus ancienne, ils se basent tout

simplement sur le calcule de probabilité qu'un utilisateur peut faire confiance à un autre. D'autre part, les approches progressives estiment les valeurs de confiance dans une certaine mesure, plutôt que d'être justes ou fausses. Dans les approches progressives, les valeurs de confiance ne sont pas interprétées comme des probabilités, mais plutôt comme des valeurs où une valeur plus élevée correspond à une confiance plus élevée.

Les différences dans la représentation de confiance ont un très grand effet sur le calcul de la confiance dans les algorithmes peer-to-peer, et même l'ensemble des algorithmes dépendent de la représentation de confiance.

3.6. Les sources d'informations

Il existe trois sources principales d'informations de confiance sur les réseaux sociaux :

- *Attitudes :* Les attitudes représentent le degré de ressemblance ou d'aversion d'un individu. Ils constituent une image qui peut être positive ou négative d'une entité. Les attitudes sont dérivées des interactions de l'utilisateur.

- *Expériences :* Les expériences décrivent la perception du membre dans ses interactions les unes avec les autres. Les expériences sont souvent utilisées comme source d'informations dans les réseaux peer-to-peer lors du calcul de la confiance entre les nœuds du réseau.

- *Comportements :* Les comportements sont identifiés par des modèles d'interactions et sa principale source d'information est l'interaction. Cela peut dépendre du type, de la fréquence ou du changement d'interaction. Tous ces comportements peuvent être utilisés pour déterminer et calculer la confiance entre les membres du réseau.

4. Modèles d'évaluation de la confiance

On peut calculer la confiance entre les différents utilisateurs en utilisant plusieurs méthodes.

- Les modèles de confiance basés sur l'architecture réseau.
- Les modèles de confiance basés sur l'interaction.
- Les modèles de confiance hybrides.

4.1 Modèles de confiance basés sur l'architecture réseau

Dans ce modèle d'évaluation de confiance, un réseau de confiance est créé pour chaque membre. En fait, c'est un graphe et ses nœuds sont les utilisateurs. La confiance entre deux membres A et B est représenté par un arc liant les deux nœuds associés dans le graphe.

Comme exemple de modèle de confiance basé sur le réseau est Tidal Trust.

4.2 Modèles de confiance basés sur l'interaction

Dans ce type de modèle on utilise seulement l'interaction au sein du réseau pour calculer la confiance sociale. Ici, il a deux types de confiance :

- *Confiance de popularité :* C'est à dire faire confiance à un utilisateur en l'acceptant membre dans la communauté.

- *Confiance de l'engagement :* Avec la confiance que la communauté investis sur chaque membre, Ils sont à la hauteur de cette confiance et cela est traduit par leurs engagements au sein de la communauté.

La combinaison de ces deux types de confiance constitue la base pour déterminer la confiance sociale dans la communauté.

Les modèles de confiance sociale basés sur l'interaction ne prennent en compte que les interactions dans la communauté pour calculer la confiance, mais ignorent la structure du réseau social.

4.3 Modèles de confiance hybrides

L'idée c'est qu'il ne faut pas négliger la structure de réseau social car il peut nous fournir des informations importantes sur la façon dont les membres se relient entre eux. C'est alors une source importante d'information pour calculer la confiance.

Les modèles de confiance hybrides sont alors en quelque sorte une combinaison des deux premiers modèles. On a vu de manière générale les modèles de calcul de confiance : modèle basé sur le réseau, l'autre modèle basé sur les interaction et enfin le modèle hybride qui est la combinaison des deux premiers. Passons maintenant a quelque algorithme connue de calcul de confiance.

Conclusion

Dans ce chapitre, nous avons présentés les différentes architectures des réseaux sociaux. Ensuite, nous avons définis les menaces qui peuvent touchés à la vie privée dans les réseaux sociaux et nous avons vus les différentes solutions qui existe. Finalement, nous avons définis la confiance, ses types et ses propriétés, et nous avons mis le point sur l'importance ultime que joue cette notion au sein des réseaux sociaux. Ainsi nous avons donnés un exemple de mécanisme de contrôle d'accès qui se base sur la confiance.

Chapitre 3 : État de l’art sur les Algorithmes de confiance

Introduction

Les algorithmes de calcul de confiance sont conçus pour relever les problèmes touchant la sécurité de notre vie privée au sein des réseaux sociaux, en formant des cercles de confiance. A l'aide de ce paramètre qu'est *la confiance*, on peut partager nos informations sans se souciés des utilisateurs malveillants qui sont. Grâce à ces algorithmes, exclues de notre cercle de confiance.

1. Les algorithmes de calcul de confiance

Nous avons plusieurs algorithmes de calcul de confiance. La question qui se pose alors c'est : Quel algorithme choisir ?

Avant de voir les méthodes de calcul de confiance appliqué pour chaque algorithme, le paramètre de complexité joue un rôle très important (On ne va pas choisir par exemple un algorithme qui calcul la confiance avec une très grande précision mais possède une grande complexité).

Dans la partie suivante, on va étudier de manière détaillé le paramètre de complexité.

1.2. La complexité d'un algorithme

L'analyse de la complexité d'un algorithme consiste en l'étude formelle de la quantité de ressources (par exemple de temps ou d'espace) nécessaire à l'exécution de cet algorithme.

L'approche la plus classique est donc de calculer le temps de calcul dans le pire des cas. Sinon, Il existe au moins trois alternatives à l'analyse de la complexité dans le pire des cas.

La *complexité en moyenne* des algorithmes, à partir d'une répartition probabiliste des tailles de données. Elle tente d'évaluer le temps moyen que l'on peut attendre de l'évaluation d'un algorithme sur une donnée d'une certaine taille.

La *complexité amortie des structures de données*, consiste à déterminer le coût de suites d'opérations.

L'*analyse lisse d'algorithme*, plus récente, se veut plus proche des situations réelles en calculant la complexité dans le pire des cas sur des instances légèrement bruitées.

Il existe plusieurs classes de complexité qui sont prédéfinie, On va citer quelques-uns dans le tableau suivant :

Temps	Type de complexité	Temps pour n = 5	Temps pour n = 10	Temps pour n = 20
$O(1)$	complexité constante	10 ns	10 ns	10 ns
$O(\log(n))$	complexité logarithmique	10 ns	10 ns	10 ns
$O(\sqrt{n})$	complexité racinaire	22 ns	32 ns	45 ns
$O(n)$	complexité linéaire	50 ns	100 ns	200 ns
$O(n\ log^*(n))$	complexité quasi-linéaire	50 ns	100 ns	200 ns
$O(n\log(n))$	complexité linéarithmique	40 ns	100 ns	260 ns
$O(n^2)$	complexité quadratique (polynomiale)	250 ns	1 µs	4 µs
$O(n^3)$	complexité cubique (polynomiale)	1.25 µs	10 µs	80 µs
$2^{\mathrm{poly}(\log(n))}$	complexité sous-exponentielle	30 ns	100 ns	492 ns
$2^{\mathrm{poly}(n)}$	complexité exponentielle	320 ns	10 µs	10 ms
$O(n!)$	complexité factorielle	1.2 µs	36 ms	770 ans
$2^{2^{\mathrm{poly}(n)}}$	complexité doublement exponentielle	4.3 s	10^{278} ans	...

Figure 7 : Les types de complexité.

Nous avons vu un paramètre de classification des algorithmes en générale qui est *la complexité*. Dans la partie suivante, on va citer les paramètres avec lesquels on peut comparer les différents algorithmes de calcul de confiance en spécifique.

1.2. Paramètre de performance des algorithmes de calcul de confiance

Il existe beaucoup d'algorithmes de calcul de confiance. Il nous faut alors définir des paramètres pour pouvoir classifier ces algorithmes selon leur performances (un algorithme peut être performant dans le cas d'un chemin court…)

- **T*ype (Local ou Global)***

Les algorithmes locaux fournissent des valeurs de confiance personnalisées qui dépendent des points de vu des utilisateurs évaluateurs.

Soit A, B et C trois utilisateurs, A évalué le niveau de confiance de C par x, tandis que B l'évalue par y (Il n'est pas nécessaire que x soit égale à y).

En revanche les algorithmes globaux fournissent la même valeur de confiance '*réputation*' pour un utilisateur donné pour tous les utilisateurs.

- ***Exploration des chemins***

L'ensemble des chemins, entre la source et la cible, pris en considération par l'algorithme pour le calcul de la confiance.

- ***Seuil***

Valeur de confiance minimale pour qu'un utilisateur soit pris en considération dans le calcul de la confiance.

- ***Complexité***

Temps (nombre d'étapes) nécessaires pour l'exécution de l'algorithme.

❖ **Scalabilité**

Aptitude d'exécution de l'algorithme sur des grands réseaux sans perdre ses performances.

❖ **Précision**

Pourcentage d'utilisateurs fiables (de confiance) recommandés par l'algorithme.

❖ **Couverture**

Pourcentage d'utilisateurs pour lesquels l'algorithme est capable de générer des recommandations.

❖ **Taux d'erreur**

Différence entre les valeurs de confiance calculées par l'algorithme et les valeurs réellement fournies par les utilisateurs.

2. Etude sur les algorithmes de calcul de confiance

2.1 Les algorithmes globale

2.1.1 Advogato

➢ **Description de l'algorithme**

Cet algorithme fonctionne en suivant 3 étapes : L'assignation ou l'affectation de capacités, ensuite la conversion du graphe et enfin le calcul du flot maximum. Le calcul de confiance se fait par rapport à un petit groupe de membres fiables qui sont appelés *SEEDS*.

➢ **Analyse de l'algorithme**

❖ *Avantage :*

Faible complexité. Elle est d'ordre O(f*m) ou m et le nombre des arcs du graphe et

f le flot maximum. Dans ce cas f = n (le nombre de sommets). Elle est alors d'ordre O(n*m).

❖ Inconvénient *:*

- Dans notre calcul, on se base juste sur le plus court chemin. On néglige la confiance entre les utilisateurs.
- La relation de propagation n'a pas de sens. Prenons l'exemple : La source a une capacité de 8. Si la source à 2 voisins, ils auront une capacité de 4, mais si par exemple elle à 4 voisins, ils auront une capacité de 2.

2.1.2 Eigen Trust

- **Description**

L'algorithme EigenTrust est un algorithme de calcul de confiance pour les réseaux peer-to-peer, développé par Sep Kamvar, Mario Schlosser et Hector Garcia-Molina. L'algorithme fournit à chaque nœud dans le réseau une valeur de confiance globale unique basée sur son historique des téléchargements et vise ainsi à réduire le nombre de fichiers inauthentiques dans un réseau P2P.

Les systèmes peer-to-peer disponibles aujourd'hui (comme Gnutella) sont ouverts, souvent anonymes et manquent de responsabilité. Ainsi, un utilisateur malveillant peut introduire dans le réseau peer-to-peer des ressources qui peuvent être inauthentiques, corrompues ou malveillantes (Virus). Cela reflète mal la crédibilité des systèmes peer-to-peer actuels.

Avec l'algorithme EigenTrust, chaque pair du système a une valeur de confiance globale unique. Tout pair demandant des ressources sera en mesure d'accéder à la valeur de confiance d'un pair et d'éviter de télécharger des fichiers provenant des utilisateurs non approuvés.

- **Algorithme**

L'algorithme d'EigenTrust est basé sur la notion de transitivité de la confiance: si un nœud i approuve un nœud j, il ferait également confiance aux pairs approuvés par j. Chaque nœud calcule la valeur de confiance locale sij pour tous les pairs avec lesquelles il a interagi.

$$s_{ij} = \mathrm{sat}(i,j) - \mathrm{unsat}(i,j)$$

sat(i,j) : Le nombre de réponse satisfaisante que le nœud i à reçus du nœud j
unsat (i,j) : Le nombre de réponse insatisfaisante que le nœud i à reçus du nœud j .

Le risque qu'on peut avoir est : Les nœuds malveillants peuvent attribuer arbitrairement des valeurs de confiance locales élevées à des nœuds mal intentionnés et des valeurs de confiance locales arbitrairement basses à de bons pairs.

La valeur de confiance locale normalisée cij est alors :

$$c_{ij} = \frac{\max(s_{ij}, 0)}{\sum_j \max(s_{ij}, 0)}$$

Basé sur l'idée de la confiance transitive, on peut créer un vecteur de confiance.

$$t_{ik} = \sum_j c_{ij} c_{jk}$$

Connaissant les valeurs cij pour l'ensemble du réseau, on va avoir alors une matrice C. De ce fait, le vecteur de confiance ti qui définit la valeur de confiance tik est donné par :

$$\bar{t}_i = C^T \bar{c}_i.$$

l'équation ci-dessus, si l'on suppose que C'est apériodique et fortement connexe, les puissances de la matrice C convergeront vers une valeur stable à partir d'un certain rang. D'ou la formule suivante :

$$\bar{t} = (C^T)^x \bar{c}_i.$$

A noté que t est bien la valeur de confiance globale pour notre réseaux.

2.2. Les algorithmes local

2.2.1 TidalTrust

Dans cet algorithme, on illustre les relations entre les personnes par un graphe orienté et on affecte a chaque arc une valeur de confiance sur l'échelle de 10. Donc comme prérequis : il faut avoir un graphe orienté et les confiances entre les utilisateurs directement connectés est déjà définis.

- **Description**

L'algorithme TidalTrust est proposé par Golbeck en 2005. Dans cet algorithme, nous évaluons la confiance entre les utilisateurs par des valeur discrètes et sur l'échelle de 10. Pour Golbeck, cette représentation de la confiance est plus instinctive que de la représentation par des valeurs continues.

Les relations entre les personnes sont représentées par un graphe orienté, et dans chaque arc on associe une valeur de confiance appartenant inclus dans l'ensemble {0,1}.

L'idée c'est que, chaque utilisateur attribue des valeurs de confiance à un ensemble de personnes, voisins directs, dont il fait confiance.

Comme prérequis de l'algorithme, il nous faut un graphe orienté comportant des arcs étiqueter (par les valeurs de confiance). Pour calculer la confiance entre une source 's' et une cible 't', l'algorithme suit deux étapes :

Etape 1 : Trouver un chemin reliant s et t, tout en évaluant les nœuds qu'on parcourt.

Etape 2 : Après avoir trouvé t, le seuil (Threshold) est défini. On retourne vers la cible s tout en faisant des mises à jour sur les confiances des différents nœuds et en

parcourant que les chemins qui satisfaits la contrainte du seuil. Et cela est donné par la formule suivante

$$t_{i,k} = \frac{\sum_{j \in adj(i) | t_{i,j} \geq max} t_{ij} * t_{jk}}{\sum_{j \in adj(i) | t_{i,j} \geq max} t_{ij}}$$

* max : Threshold

* t i,k : la confiance que fait i à k

* j appartenant à adj(i) : ce sont tout les noeuds voisins de i

- **Analyse de l'algorithme**

  - *Avantage :*

    La complexité de cet algorithme est faible. Elle est de l'ordre de O(n+m) avec n et m désignent successivement le nombre des nœuds et le nombre d'arête dans le graphe.

  - *Inconvénient :*

La restriction faite sur les plus courts chemins peut engendrer la non prise en considération des autres chemins (perte d'informations). Deuxièmement, le grand problème se pose lorsqu'il y a un chemin unique (chaîne) entre la source et la cible. Supposons qu'il existe une longue chaîne unique entre la source et la cible et chaque sommet de la chaîne à une valeur de confiance égale à 9 vers son voisin. Supposons qu'il existe une autre chaîne avec la même distance où chaque sommet à une valeur de confiance égale à 1 vers son voisin à l'exception de l'avant dernier (le sommet juste avant la cible) qui a une valeur de confiance égale à 9 vers la cible. Avec l'algorithme *TidalTrust*, dans les deux cas la valeur de confiance calculée entre *s* et *t* est égale à 9. Cependant, il est clair que la valeur de confiance calculée dans le premier cas doit être supérieure que celle calculée dans le deuxième cas.

2.2.2 MoleTrust

L'algorithme *MoleTrust* est proposé par Massa. Il est utilisé dans le site Epinions.com qui est un réseau social dédié à la vente de produits commerciaux. Cet algorithme permet de calculer la confiance qu'un utilisateur qui a envers un autre utilisateur en parcourant le graphe de confiance *G* qui a la même forme que l'architecture de notre réseau social. En parcourant le chemin entre la source et la cible, on va, au même temps, propager la confiance le long des arcs.

La confiance d'une cible donnée dépend des confiances que les autres utilisateurs ont envers elle et des confiances de ces derniers.

Principalement, le calcul de la confiance entre une source *s* et une cible *t* se fait en deux étapes.

La première étape consiste à enlever les circuits de notre graphe confiance

La deuxième étape consiste en la propagation de la confiance depuis la source *s* jusqu'au sommet cible.

Le calcul de la confiance d'un sommet visité est donnée par l'équation suivante :

$$b(x_j) = \frac{\sum_{k \in p(j)} b(x_k) T(x_k, x_j)}{\sum_{k \in p(j)} b(x_k)},$$

$b(x_j)$: confiance du sommet x_j ;

$T(x_k; x_j)$: valeur de confiance qu'un sommet x_k a envers le sommet x_j ;

$p(j)$: l'ensemble des prédécesseurs du sommet x_j .

Restriction MoleTrust

- En prend en considération que les nœuds qui ont une valeur de confiance supérieur ou égal à un seuil définit statiquement au début.

- La profondeur maximale est fixée à 5. Donc les chemins entre deux nœuds au-delà de 5 niveau ne sont pas pris en considération.

2.2.3 MaxTrust

MaxTrust est un algorithme de calcule de confiance, ce dernier prend en charge les insuffisances des deux algorithmes MoleTrust et TidalTrust et préserve leurs avantages.

- **Exploitation des chemins de confiance**

  Prenant par exemple, c'est x est amie de y et y est amie de z , si on veut calculer la confiance entre x et z les algorithmes TidalTrust et MoleTrust ne prennent pas en considération la confiance entre x et y , par contre le point fort du l'algorithme MaxTrust c'est que ce dernier prend en considération la confiance entre x et y , pour éviter de tomber dans le problèmes de perte d'informations.

- **Propagation de la confiance**

  Lorsqu'on veut calculer la confiance d'un sommet s envers un sommet x qui ne sont pas en relation directe MaxTrust utilise le principe de la propagation de la confiance du sommet source vers le sommet cible , en utilisant la fonction de propagation suivant :

$$t_{sx} = \frac{t_{sr} \times t_{rx}}{\frac{1}{2}(t_{sr} + t_{rx})}.$$

- **Agrégation des valeurs de confiances**

L'agrégation des valeurs de confiance intervient lorsque plusieurs chemins de confiance issus de source(s) mènent vers un même sommet donné x. Dans ce cas la confiance du sommet s envers le sommet x se calcule en agrégeant les confiances des sommets prédécesseurs 1, 2, …, n envers x. Pour réaliser cette agrégation on utilise la fonction d'agrégation suivant :

$$t_{sx} = \frac{\sum_{i=1}^{n}(t_{si} \times t_{ix})}{\frac{1}{2}\sum_{i=1}^{n}(t_{si} + t_{ix})},$$

- **Description de l'algorithme**

Soit deux utilisateurs i et j qui sont directement connecté ont une confiance mutuelle qui résulte d'une interaction directe. Notons ti,j la confiance que fait i à j.

Construisons un graphe à titre d'exemple pour simuler le fonctionnement de cet algorithme. Le graphe sera évidemment orienté puisque l'une des propriétés de la confiance qu'elle est asymétrique : Donc la confiance que A fait à B n'est pas forcement égale à la confiance que B fait à A.

Un exemple de graphe est ci-dessous. Les utilisateurs sont modélisés par les nœuds tandis que les arêtes représentent le degré de confiance entre les différents nœuds. Le but de l'algorithme c'est de mesurer la confiance entre deux utilisateurs qui ne sont pas directement connecté (*Source* et *Sink*).

2.3.5.2 Analyse de l'algorithme

Avantage :

La complexité de cet algorithme est faible. Elle est de l'ordre de O(n+m) avec n et

m désignent successivement le nombre des nœuds et le nombre d'arête dans le graphe.

Inconvénient :

La restriction faite sur les plus courts chemins peut engendrer la non prise en considération des autres chemins (perte d'informations). Deuxièmement, le grand problème se pose lorsqu'il y a un chemin unique (chaîne) entre la source et la cible. Supposons qu'il existe une longue chaîne unique entre la source et la cible et chaque sommet de la chaîne à une valeur de confiance égale à 9 vers son voisin. Supposons qu'il existe une autre chaîne avec la même distance où chaque sommet à une valeur de confiance égale à 1 vers son voisin à l'exception de l'avant dernier (le sommet juste avant la cible) qui a une valeur de confiance égale à 9 vers la cible. Avec l'algorithme *TidalTrust*, dans les deux cas la valeur de confiance calculée entre *s* et *t* est égale à 9. Cependant, il est clair que la valeur de confiance calculée dans le premier cas doit être supérieure que celle calculée dans le deuxième cas.

Chapitre 4 : Réalisation

Introduction

Ce chapitre décrit la phase de l'implémentation des modèles de confiance TidalTrust, MoleTrust et MaxTrust. Tout d'abord, on va définir les paramètres qu'on va prendre en compte lors de la comparaison entre les trois algorithmes. Ensuite, on va présenter le « Data-set » nommé 'Resident hall'. Enfin, nous allons implémenter les trois algorithmes et pour en finir faire une évaluation.

1. Paramètre de comparaison d'algorithmes

Le but des techniques prédictives et prévisionnelles est d'envisager d'une façon ou d'une autre ce qui devrait être, selon l'idée qu'on se fait d'une réalité. Cette idée est ensuite modélisée et validée sous forme d'équation(s).

Seulement voilà, il y aura toujours des *ÉCARTS* entre nos résultats et cette facétieuse réalité. Les raisons sont multiples et n'allons pas les énumérer. Le cas le plus pervers est celui d'un modèle qui prévoit pile poil une valeur mais où une erreur de mesure de la réalité laisse croire qu'il est mauvais.

On peut alors comparer des algorithmes en se basant sur leur capacité de prédire des valeurs de confiance existante dans le DataSet. Le meilleur algorithme c'est ce lui qui fait cette prévision avec un minimum d'erreur.

1.1. Erreur Absolue Moyenne (EAM)

Elle mesure la déviation absolue moyenne entre les valeurs de confiance prédites et les valeurs réelles. La formule de EAM est donnée par l'équation suivante :

$$EAM = \frac{\sum_{i=1}^{N} |p_i - r_i|}{N},$$

Ou N désigne le nombre des valeurs prédite par l'algorithme et « pi » la valeur de confiance prédite et « ri » la valeur de confiance réelle

1.2. Erreur Quadratique Moyenne « EQM »

Elle mesure la déviation moyenne des carrés entre les valeurs de confiance prédites et les valeurs de confiance réelles. L'EQM est donnée par l'équation suivante :

$$EQM = \sqrt{\frac{\sum_{i=1}^{N}(p_i - r_i)^2}{N}}.$$

Réel	Prévu	e	e^2	|e|
9	10	-1	1	1
15	15	0	0	0
20	20	0	0	0
24	25	-1	1	1
29	30	-1	1	1
36	35	1	1	1
42	40	2	4	2
43	45	-2	4	2
52	50	2	4	2
54	55	-1	1	1
	moy écarts :	-0,1	17	11

Nombre d'observations	10
Carré moy. erreurs (MSE)	1,7
Erreur quadratique moy.	1,304
MAE	1,1

Figure 8 : Exemple de calcul des erreurs de prédiction.

1.3. Précision

Elle mesure le pourcent*age* d'utilisateurs fiables recommandés par l'algorithme.

$$Prc = \frac{N_f}{N_f + N_{nf}}.$$

Notons que Nf et le nombre d'utilisateur fiable et Nnf celui des utilisateurs non fiables recommandés par l'algorithme.

1.4. Couverture

La couverture est proportionnellement liée à la précision. Ce paramètre nous donne une idée sur la capacité de l'algorithme de générer des recommandation (définition des nœuds fiable).

La relation est donnée par : $$Cvr = \frac{N_f}{N_{tf}}.$$

2. Le jeu des données Data Set

Un jeu de données (en anglais *dataset*) est un ensemble de valeurs (ou données) où chaque valeur est associée à une variable (ou attribut) et à une observation. Une variable décrite l'ensemble des valeurs décrivant le même attribut et une observation contient l'ensemble des valeurs décrivant les attributs d'une unité (ou individu statistique).

- **Residence hall**

Ce réseau dirigé contient des évaluations de confiance entre 217 résidents vivant dans une résidence située sur le campus de l'Université nationale australienne.

Dans le fichier contenant le DataSet, chaque ligne est associée à une liaison entre deux nœuds. Le premier champ c'est l'identifiant de l'utilisateur x, le deuxième champ c'est l'identifiant d'un autre utilisateur y et qui est en relation avec x (même ligne). Le troisième champ désigne la confiance que fait x à y. Cette confiance est évaluée sur une échelle de 5.

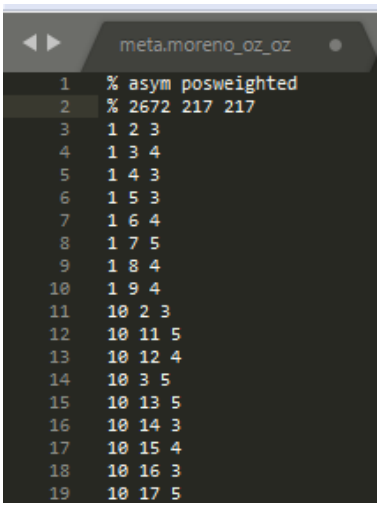

Figure 9 : Format DataSet.

- **Caractéristique de Residence hall**

Dans le DataSet « Residence hall », la distribution des valeurs de confiance sont donnés comme suit :

Val. de conf.	1	2	3	4	5	Total
Effectif	38	110	1624	602	298	2672
Pourcentage	1.4 %	4.12 %	60.78 %	22.53 %	11.15 %	100 %
Cumulé	1.42 %	5.54 %	66.32 %	88.85 %	100 %	-

Figure 10 : Distribution des valeurs de confiance du DataSet.

3. Implémentation des algorithmes

Cette partie est organisée comme suit : nous décrivons le langage de programmation et l'environnement de développement utilisés pour l'implémentation des algorithmes proposés, à savoir le langage Java et l'environnement Eclipse. Ensuite nous présentons le pseudo-code de chaque algorithme implémenté à savoir TidalTrust, MoleTrust et MaxTrust.

3.1. Langage de programmation

Pour l'implémentation de nos algorithmes , nous utilisons le langage de programmation Java est un langage de programmation orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy(cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au SunWorld.

3.2. Environnement de développement

Pour le développement de nos algorithmes , nous utilisons l'environnement de développement intégré (IDE) Eclipse est un environnement de développement intégré (IDE) utilisé dans la programmation informatique , et est l'IDE Java le plus utilisé. Il contient un espace de travail de base et un système de plug-in extensible pour personnaliser l'environnement. Eclipse est principalement écrit en Java et son utilisation principale est le développement d'applications Java, mais il peut également être utilisé pour développer des applications dans d'autres langages de programmation viadesplugins,notamment Ada , ABAP , C , C++ , C# etc.

3.3. Pseudo-code

Après avoir présenté le fonctionnement des algorithmes à savoir TidalTrust, MoleTrust et MaxTrust dans le chapitre précèdent, nous allons donner dans ce qui suit leurs pseudo-code.

3.3.1 Pseudo-code de l'algorithme TidalTrust

```
fonction TidalTrust(source , cible , seuil, nombreNoeud)

        lniveau=niveau(source,nombreNoeud)

        treshold=treshold(source,cible)

        source_atteint=false

        for all noeud in lniveau(source +1) do
                if noeud is cible do
                        return confiance (noeud )
                end if
        end do

        niveau=niveau(cible)

        while source_atteint not atteint  do

            for all noeudx n lniveau(niveau-1) do
                if noeudx is source do
                    for all noeudy in lniveau (niveau ) do
                        if noeudy successeur noeudx and confiance(noeudx,noeudy)  >= treshold do

                            confianceNoeud(noeudx)=RelationAgregation()

                        end if
                    end for
                    source_atteint = true
                end if
            end for

            if source_atteint not atteint do
                    for all noeudx in lniveau(niveau-1) do
                            for all noeudy in lniveau (niveau )do
                                    if(noeudy preecesseur noeudx and niveau not niveauc(cible ) )
                                            confianceNoeud(noeudx)=confiance (noeudx ,noeudy)

                                    else
                                        if  noeudy preecesseur noeudx and confiance (noeudx ,noeudy) >= treshold do
                                                confianceNoeud(noeudx)=RelationAgregation()
                                        end if

                                    end if

                            end do
                    end do
            end if

        niveau=niveu-1

        end while

    return confianceNoeud(source)
```

Figure 11 : Pseudo-code de l'algorithme TidalTrust

3.3.2 Pseudo-code de l'algorithme MoleTrust

```
fonction MoleTrust( confiance(matrice),source , cible , seuil, nombreNoeud)

        lniveau=niveau(source,nombreNoeud)


        if profondeur(source,cible) > 5 do
                return 0
        end do

        cible_atteint=false

        set niveau=1

        while cible_atteint!=true do

            for all noeud in lniveau(niveau) do
                    if cible atteint do
                        cible_atteint = true
                    end if
            end for


            if cible_atteint not atteint do$
                    for all noeudx in lniveau(niveau+1) do
                            for all noeudy in lniveau (niveau )do
                                    if(confiance(noeudy ,noeudx)!=0 and confianceNoeud(noeudy) >= seuil )
                                            confianceNoeud(noeudx)=numerateur/denomirateur
                                    end if
                            end do
                    end do
            end if


        niveau=niveu+1

        end while


    return confianceNoeud(cible)
```

Figure 12 : Pseudo-code de l'algorithme MoleTrust

3.3.3 Pseudo-code de l'algorithme MaxTrust

```
fonction MaxTrust( confiance(matrice),source , cible , seuil, nombreNoeud)

        lniveau=niveau(source,nombreNoeud)


        cible_atteint=false

        set niveau=0

        for all noeud in lniveau(niveau +1) do
                calcule_confiance1(noeud )
                if noeud is cible do
                        return confiance (noeud )
                end if
        end do

        niveau = 1

        while cible_atteint!=true do

            for all noeudx n lniveau(niveau) do
                    for all noeudy in lniveau (niveau ) do
                        if noeudy successeur noeudx do
                            calcule_confiance2(noeudy)
                        end do
                    end do
                    if noeudx isatteint do
                        cible_atteint = true
                    end if
            end for


            if cible_atteint not atteint do
                    for all noeudx in lniveau(niveau+1) do
                            for all noeudy in lniveau (niveau )do
                                    if(noeudy preecesseur noeudx and confianceNoeud(noeudy) >= seuil )
                                            confianceNoeud(noeudx)=(2 * numerateur )/denominateur
                                    end if
                            end do
                    end do
            end if


        niveau=niveu+1

        end while


    return confianceNoeud(cible)
```

Figure 13 : Pseudo-code de l'algorithme MaxTrust

4. Évaluation des algorithmes

Après avoir implémenter les algorithmes à savoir TidalTrust, MoleTrust et MaxTrust. Nous comparons leurs résultats en utilisant un jeu de données de test réel, à savoir le jeu données Residence hall, la comparaison se fera selon les quatre mesures suivantes :

- L'erreur Absolue Moyenne(EAM).
- L'Erreur Quadratique Moyenne (EQM).
- La Précision (Prc).
- La Couverture (Cvr).

Résultats : calcule de confiance entre deux résidents

L'application qu'on a développée nous proposent une interface pour calculer la confiance entre deux résidents qui ne sont pas nécessairement interconnecté entre eux, Il faut juste choisir l'algorithme de calcule de confiance désirée et ensuite fournir la source et la destination et le seuil (MaxTrust, MoleTrust).

La figure 13 nous montre un exemple de calcule de confiance entre des résidents en utilisant les différents algorithmes, à savoir TidalTrust, MaxTrust et MoleTrust.

Le paramètre seuil est obligatoire pour les algorithmes MaxTrust et MoleTrust par contre Tidal comme on vu il le calcule d'une manière dynamique (threshold).

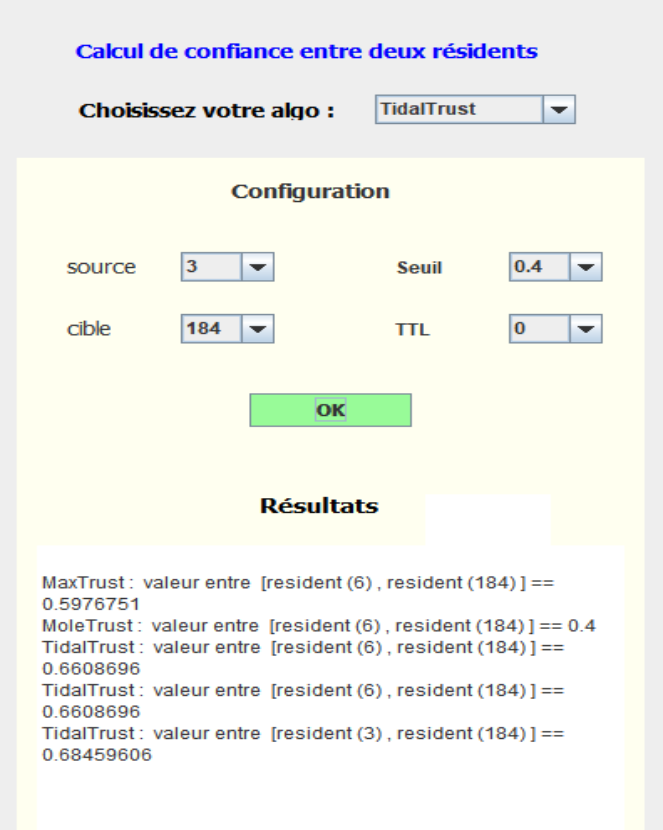

Figure 14 : interface de calcule de confiance entre deux résidents.

Résultats : comparaison des algorithmes

Après avoir calculer la confiance entre deux résidents, une deuxième interface est à notre disposition pour comparer les algorithmes de notre choix par les mesures d'erreur (EAM, EQM, Prc).

Mais d'abord if faut spécifier le pourcentage de l'échantillon avec lequel on veut travailler, Puis on valide et le résultat sera afficher dans un tableau.

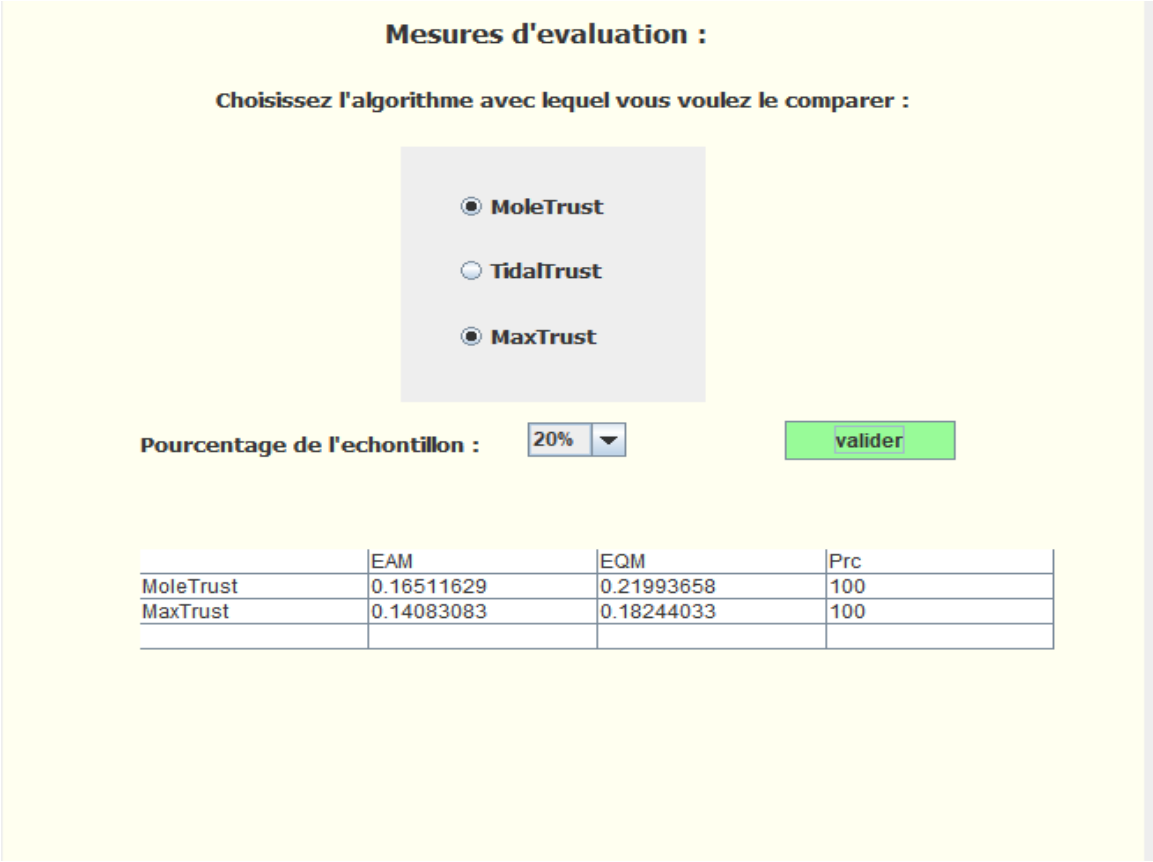

Figure 15 : Interface de comparaison entre les algorithmes.

Conclusion

Dans ce chapitre, nous avons exposés notre implémentation des modèles. Ensuite nous avons comparé les résultats des différents modèles. MaxTrust donne des résultats très satisfaisants par rapport à TidalTrust et MoleTrust.

Conclusion et perspectives

La nature ouverte des réseaux sociaux, pousse les utilisateurs à prendre la protection de leurs vies privées plus au sérieux. De ce fait, il est indispensable de concevoir des mécanismes permettant la reconnaissance et l'identification des utilisateurs malveillant et des utilisateur fiables.

Un exemple de ce mécanisme et le calcul de confiance au sein des réseaux sociaux et qui s'est avérer par la même occasion très important pour la protection de la vie privée.

Dans ce projet, nous avons défini les réseaux sociaux ainsi que leur architecture. Ensuite, nous avons mis le point sur les menaces qui peuvent affectés nos vies privées et proposé quelques solutions. Et nous avons présentés les algorithmes d'évaluation de confiance : TidalTrust, MoleTrust et MaxTrust en mettant l'accent sur leur avantages et inconvénients.

Grace à un jeu de données nommé « Resident hall », nous avons pu faire la comparaison entre les performances des trois algorithmes. Nous avons conclu que l'algorithme MaxTrust donne des résultats très satisfaisants, d'où l'efficacité de ce mécanisme pour la création de cercle de confiance, et alors la protection de la vie privée.

Comme perspectives, nous citons :

- Étude des algorithmes dynamique de calcul de confiance dans le domaine de l'internet des objets (IOT).
- Ajuster de manière dynamique les paramètres (seuil, threshold.) pour pouvoir intégrer un très grand nombre d'algorithmes de calcul de confiance ce qui implique un élargissement du périmètre des tests et de comparaison.

Bibliographie

[1] :Mémoire de fin d'études : Protection de la vie privée à base d'agents dans un système d'e-learning.

[2] : MASTER THESIS : ANALYSIS OF ALGORITHMS FOR DETERMINING TRUST AMONG FRIENDS ON SOCIAL NETWORKS

[3] : Mémoire de fin d'études : Cercles de confiance dans les entrepôts de données : application aux réseaux sociaux

[4] :The EigenTrust Algorithm for Reputation Management in P2P Networks

[5] : Mémoire de fin d'études : ANALYSE DE RISQUE DANS LES SYSTÈMES DE CONTRÔLE D'ACCÈS

[6] : Mémoire pour l'obtention de magister en informatique : Implémentation d'un agent mobile dans un environnement pair à pair

[7] : IRIS: A Novel Method of Direct Trust Computation for Generating Trusted Social Networks

[8] : trust in large-scale peer-to-peer systems. YU, B. SINGH, M, AND SYCARA, K. Developing (2004).

[9] : Doctorat en science, Université Mentouri S. Hacini : Mise en oeuvre de la confiance et de l'adaptabilité pour la protection de l'agent mobile,2008.

[10] :Doctor of philosophy, Department of Computer Science, University of Maryland Computing and applying Trust in Web Based Social Networks. J. A. Golbeck.

[11] : Trust and Trustworthiness. Russell Sage Foundation, New York. R. Hardin.

[12] : Lecture Notes In Computer Science Trust and reputation systems.

Foundations of Security Analysis and Design .A. Josang , A. Aldini, R. Gorrieri