République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de

La Recherche Scientifique

Université Mouloud Mammeri de Tizi-Ouzou

Faculté de Génie Electrique et d'Informatique

Département d'Informatique

Projet de fin d'étude pour l'obtention du diplôme de
Master en Informatique option Systèmes Informatiques

Sécurisation des données dans un environnement de Data Mining

Proposé Par :

Mr. Chaieb Yazid

Réalisé par :

Mlle. Moussaoui Sonia

Promotion : 2012/2013

Remerciements

En premier lieu, je remercie Dieu le tout puissant pour m'avoir aidé afin d'arriver à terme de ce travail qui représente le fruit de plusieurs mois de travail.

Je tiens à remercier mon promoteur Mr Chaieb. Qui m'a orienté et aidé tout au long de mon travail de préparation du mémoire.

Mes remerciements distingués à tous les enseignants du département d'Informatique de l'UMMTO, qui ont su me transmettre tout le savoir et les connaissances nécessaires afin d'avancer dans ma vie future.

Mes sincères remerciements vont également aux membres de jury, qui me font l'honneur de juger ce travail.

Mes chaleureux remerciements à toutes les personnes, qui m'ont aidé de prés ou de loin.

A tous merci.

Dédicaces

Je dédie ce modeste travail :

En première lieu à la plus merveilleuse Maman au monde qui a été un modèle et une source de bonheur quotidien depuis toujours,

A ma grande sœur Sabrina, son mari et mon petit neveu adoré Adam,

A ma sœur Lydia qui me manque beaucoup et son mari,

A mon frère Sammy à qui je souhaite un très bel avenir,

À ma grand-mère à qui je souhaite une longue vie,

A ma deuxième maman, Sissi, mes frères et leurs petites familles,

Ainsi qu'a mon adorable famille, du plus grand au plus petit,

A mon très cher Merzak qui a toujours été là pour moi et à toute sa famille,

À tous mes amis, pour leur soutien,

A la mémoire de Tonton Makhlouf et mon camarade et ami Malik, que dieu les accueils dans son vaste paradis.

Sonia

Sommaire

Sommaire

Introduction Générale

CHAPITRE I : *Grilles de Calcul*

I. INTRODUCTION 1
1. Le concept de grille de calcul 1
2. Les principales caractéristiques d'une grille 2
II. ARCHITECTURE D'UNE GRILLE 3
III. Architecture OGSA 7
1. origine de l'architecture OGSA 7
2. Modèle d'architecture OGSA 7
IV. EVOLUTION DES GRILLES 9

PREMIERE GENERATION 9

DEUXIEME GENERATION 10

TROISIEME GENERATION 11

V. LES DIFFERENTS TYPES DE GRILLES 12
1. Grille d'information 12
2. Les grilles de stockage 12
3. Les grilles de calcules 12
VI. Modèle de déploiement 13
1. Modèle Centralisé 13
1.1. Architecture centralisée pour les grilles de stockage : peer-to-peer assisté 13
1.2. Modèle Client/serveur pour les grilles de calcul 14
1.2.1. L'internet Computing : Le projet SETI 14
1.2.2. Métacomputing : Le projet NetSolve 15
2. Modèle décentralisé ou distribué 17
2.1. Architecture pour Grilles de Stockage : le pure Peer-to-Peer 17
2.2. Vers une architecture distribuée pour les Grilles de Calcul 18
3. Modèle hybride 18
3.1. Un modèle hybride pour les Grilles de Stockage : Les réseaux super nœuds 18
3.2. Une architecture hybride pour les Grilles de Calcul : XtremWeb 19
VII. Principaux enjeux des grilles 21
1. Partage de ressources 22

2. Utilisation efficace des ressources 22
3. Abolition de la distance 22
4. Mise en place de normes 22
VIII. Domaines d'application des grilles 23
1. Calcul distribué 23
2. Calcul à haut débit 23
3. Calcul à la demande 24
4. Traitement massif de données 24
5. Travail collaboratif 24
IX. Avantages et inconvénients des grilles de calcul 24
1. Les avantages 24
2. Inconvénients 25
X. Conclusion 26

CHAPITRE II : *Data Mining et Confidentialité*

I. INTRODUCTION 27
II. Processus ECD 27
3. Définition 28
4. Etapes du processus ECD 28
a. Positionnement du problème 28
b. Collecte et sélection des données 28
c. Nettoyage des données 29
d. Actions sur les attributs 29
e. Construction du modèle Data Mining 30
f. Evaluation du résultat 30
g. Intégration de la connaissance 31
III. Le Data Mining 32
1. Définition 32
2. Principes 32
3. Taches de Data Mining 32
3.1. La classification automatique supervisée 32
3.2. La classification automatique non supervisée 32
3.3. Règles d'association 33
4. Techniques de Data Mining 33
4.1. Fouille distribuée (Distributed Data Mining) 34
4.2. Approches proposées 35
4.2.1. Modèle de Partitionnement des données 35
4.2.2. Perturbation des données 37
5. Domaines d'application 38
6. Difficultés du Data Mining 39
IV. Etat de l'art des algorithmes de Data Mining 40

1. Introduction 40
2. La classification automatique non supervisée 40
2.1. Introduction 40
2.2. Définition 41
2.3. Terminologie 41
2.3.1. Représentation d'une classe 41
2.3.2. Notion de proximité 42
2.3.3. L'inertie 43
2.4. Les algorithmes de Clustering 43
2.4.1. Les algorithmes de partitionnement 43
2.4.1.1. L'algorithme K-means 43
2.4.1.2. L'algorithme K-médoids 45
2.4.1.3. L'algorithme PAM 46
2.4.1.4. L'algorithme CLARA 46
2.4.1.5. L'algorithme CLARANS 46
2.4.2. Les algorithmes hiérarchiques 47
2.4.2.1. L'algorithme BIRCH 48
2.4.2.2. L'algorithme DIANA 49
2.4.3. Les algorithmes basés sur la densité 49
2.4.3.1. L'algorithme DBSCAN 50
2.4.3.2. L'algorithme OPTICS 51
2.4.4. Autres algorithmes 51
2.4.4.1. Algorithme basé sur la grille 51
2.4.4.2. Algorithme basé sur les statistiques 51
2.5. Conclusion 52
3. La classification automatique supervisée 52
3.1. Introduction 52
3.2. Problématique 53
3.3. Les algorithmes de classification 53
3.3.1. Les arbres de décision 53
3.3.2. L'algorithme K-NN 55
3.3.3. Réseau de Neurones 55
V. Confidentialité 57
1. Définition 58
2. Types de confidentialités 60
2.1. Information personnelle 61
2.2. Catégorie d'informations personnelles 62
2.3. Confidentialité des informations personnelles 63
1. L'acquisition et la conversation 63
2. L'intégrité 63
3. L'agrégation et la dérivation des données 63
4. Le partage d'informations 63
5. Contrôle approprié de l'accès 64

2.4. Techniques de gestion de la confidentialité 64
2.5. Recherche de savoir KD et confidentialité 65
2.6. Confidentialité et coopération 66
2.7.
VI. Conclusion 68

CHAPITRE III : *Préservation de la vie privée dans le Data Mining*

I. INTRODUCTION 69
II. Techniques de préservation de la vie privée dans le Data Mining PPDM 70
1. La méthode de randomisation 71
1.1. Définition 71
1.2. Description détaillée 71
2. Anonymisation basée groupe 72
2.1. Le K-anonymat 72
2.1.1. Définition 72
2.1.2. Description détaillée 72
2.1.3. Méthode 73
2.1.4. Les principaux problèmes 75
2.2. Le l-Diversité 77
2.2.1. Définition 77
2.2.2. Description détaillée 77
2.2.3. Méthode 78
2.3. Le modèle t-proximité 78
2.3.1. Définition 78
2.3.2. Description détaillée 78
3. Préservation distribuée de la vie privée 78
4. L'efficacité de l'application de déclassement 81
5. Inférence à partir de résultats 82
III. Conclusion 84

CHAPITRE IV : *Etude expérimentale*

I. INTRODUCTION 86
II. IMPLEMENTATION 86
2.1. Grid Sim 87
2.2. Environnement d'expérimentation 89
2.3. Notions de la cryptographie 89
2.3.1. Types de chiffrements 89
2.3.2. Sécurité 89
2.4. Interprétation 89
2.5. DHD (Data Hiding in Database) 89

2.5.1. Définition 89
III. PRESENTATION DES 3 ALGORITHMES (VAIDYA, DNSP, PSPP) 90
3.1. Protocole DNSP 91
3.2. Protocole de Vaidya 92
3.3. Protocole PSPP 93
3.3.1. L'algorithme 93
3.3.2. Analyse du coût de communication et de calcul 93
3.3.3. Analyse de la sécurité 94
3.3.4. Travaux connexes 94
IV. RESULTATS ET PERSPECTIVES 95
4.1. Résultats 95
4.1.1. Résultats de l'application du chiffrement de paillier 95
4.2. Résultats expérimentaux 96

Conclusion

Bibliographie

Introduction Générale

Introduction Générale

L'Homme a soif de connaître son avenir et de comprendre ce qui l'entoure. À cette fin, de nombreux instruments de mesure ont été inventés. Actuellement, ces instruments devenus très précis génèrent de grandes quantités de données. Pour mieux appréhender le monde qui l'entoure, l'Homme utilise ces données pour modéliser son environnement. En analysant les modèles et en *simulant* leur évolution, il devient possible de comprendre des phénomènes et systèmes complexes ou d'en inventer de nouveaux, et même de *prévoir* leurs possibles évolutions.

Cependant, les instruments de mesure, très précis, peuvent générer quotidiennement de grandes masses de données. Les modèles deviennent également de plus en plus complexes et prennent en compte un nombre de paramètres sans cesse grandissant. Simuler l'évolution de tels modèles demande donc une puissance de calcul et de stockage de plus en plus grande. Face à cette demande croissante de puissance, les *grilles de calcul* apparaissent de plus en plus comme *la solution* de demain.

Seulement une grille de calcul est un instrument particulièrement complexe car il est composé de milliers de machines souvent hétérogènes et réparties géographiquement dans des institutions qui mettent en commun leurs ressources.

Aujourd'hui, les entreprises ont à leur disposition une masse de données importante. En effet, les faibles coûts des machines en termes de stockage et de puissance ont encouragé les sociétés à accumuler toujours plus d'informations. Cependant, alors que la quantité de données à traiter augmente énormément le volume d'informations fournies aux utilisateurs n'augmente lui que très peu. Ces réservoirs de connaissance doivent être explorés afin d'en comprendre le sens et de déceler les relations entre données, des modèles expliquant leur comportement.

Le terme de Data Mining est souvent employé pour désigner l'ensemble des outils permettant à l'utilisateur d'accéder aux données de l'entreprise, de les analyser. Nous restreindrons ici le terme de Data Mining aux outils ayant pour objet de générer des informations riches à partir des données de l'entreprise et de découvrir des modèles implicites dans les données. Ils peuvent permettre par exemple à un magasin de dégager des profils de client et des achats types et de prévoir ainsi les ventes futures.

Au cours de ce travail nous aurons l'occasion de parler tout d'abord au cours du premier chapitre des grilles de calcules et de ses différentes architectures, ensuite au cours du deuxième chapitre nous verrons les différentes méthodes d'extraction de données ce qui nous mènera vers le troisième chapitre qui parle de la première contrainte rencontrée au cours de l'extraction des données qui est la préservation de la vie privée et de la confidentialité ainsi que ses multiples techniques. Nous verrons notamment en détail trois algorithmes connus de PPDM(Privacy Preserving DataMining) qui sont ceux de Vaidya,

DNSP(Distributed Non-private Scalar Product) et PSPP (Private Scalar Product Protocol) au cours du dernier chapitre de l'étude expérimentale.

CHAPITRE I

Grilles de Calcul

I. Introduction :

Les applications modernes exigent des capacités de calcul et de stockage qui dépassent celles d'un individu ou d'une institution [1,2]. En plus de cette exigence, l'accès à l'information doit pouvoir être accessible n'importe quand, n'importe où et à partir de dispositifs physiques très diversifiés (informatique) [3]. Pour répondre à de telles exigences, il est donc nécessaire de rechercher d'autres types d'architectures différentes des architectures classiques (séquentielles, parallèles, distribuées et mobiles). Ainsi, l'accroissement constant de la puissance des ressources et l'immense réserve disponible via Internet, ainsi que la banalisation du haut débit et des technologies réseaux offrent l'opportunité de globaliser l'accès aux ressources informatiques largement distribuées dans le monde, pour l'exécution d'applications de très grandes tailles [1, 2, 4]. C'est cette opportunité qui a permis aux grilles de calcul et de données d'émerger comme un important domaine du calcul parallèle et distribué, qui se distingue par son orientation vers le partage de ressources hétérogènes à grande échelle [1, 2, 4].

Les grilles de calcul sont de plus en plus utilisées, aussi bien dans le monde de la recherche que dans celui de l'industrie. Il s'agit en effet d'un instrument puissant, permettant de traiter des problèmes difficiles faisant appel à de nombreuses données de taille importante. Cependant, la conception d'applications pour ce type de système est complexe. Les applications sont composées de processus distribués géographiquement s'exécutant sur les nœuds appartenant aux grilles. Lors de l'exécution d'une application distribuée, les processus qui la composent coopèrent notamment en partageant des données. De plus, de nouvelles ressources peuvent être ajoutées à tout moment, de même que des ressources présentes peuvent disparaître (pannes, arrêts volontaires).

1. Le concept de grille de calcul :

Le concept de grille de calcul (Grid Computing) a été introduit, pour la première fois en 1999, par Foster et Kesselmann [5] qui l'ont défini par analogie à un réseau électrique. Il n'existe pas actuellement de consensus sur la définition du concept de grille de calcul et l'on trouve plusieurs définitions de ce concept dans la littérature [6, 7, 8]. Ainsi, Plaszczak et Wellner [8] définissent une grille de calcul comme une technologie permettant la virtualisation de ressources pour répondre à leur demande et a leur partage entre plusieurs organisations. La société IBM définit une grille de calcul comme une méthode utilisant des protocoles standards pour faciliter les accès aux applications et aux données, à la puissance de calcul et à la capacité de stockage entre réseaux de machines a travers le réseau Internet [2]. Buyya, dans [6], définit une grille comme un système parallèle et distribué qui permet le partage, la sélection et l'agrégation de ressources dynamiques géographiquement distribuées. Chacune de ces ressources a ses propres caractéristiques, ses propres performances, ses propres politiques de gestion, d'accès, etc. Pour le laboratoire européen CERN (Conseil Européen pour la Recherche Nucléaire), une grille représente un service pour le partage de la puissance de calcul et les capacités de stockage d'un ensemble de machines à travers le réseau Internet [9]. Malgré cette pluralité de définitions, celle de Ian Foster [24ghalem] semble actuellement faire l'unanimité dans le domaine des grilles. Cette définition présente une grille comme une infrastructure virtuelle constituée d'un ensemble

de ressources hétérogènes, coordonnées et garantissant des qualités de service non-triviales (voir Figure 1.1).

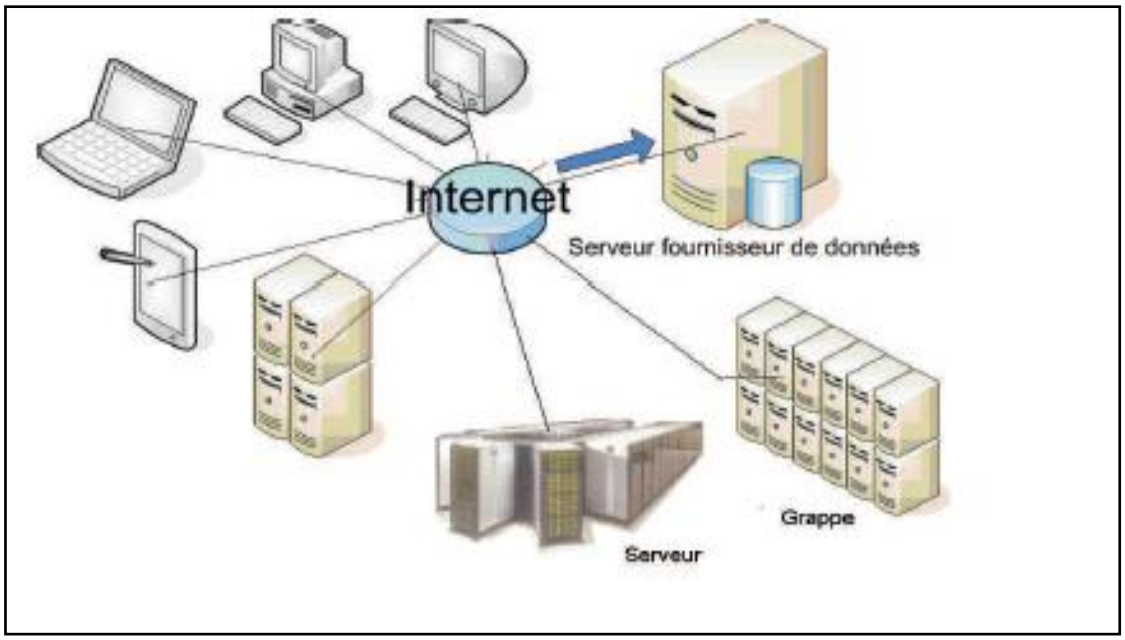


Figure 1.1 : exemple d'une grille informatique

Les grilles de calcul sont une architecture en plein développement. Elles consistent en un réseau d'ordinateurs faiblement couplés et ont pour but d'offrir une très grande puissance de calcul à leurs utilisateurs de la façon la plus transparente possible. Ces ordinateurs peuvent être des supercalculateurs, des clusters ou des stations de travail ordinaires. Ils sont reliés par un réseau à très grande échelle, le plus souvent l'internet. De ce fait, les grilles sont plus un ensemble de logiciels permettant de répartir et d'exécuter des applications sur un parc de machines hétérogènes et placées sous des autorités administratives différentes.

2. Les principales caractéristiques d'une grille :

Le calcul sur grille consiste à fédérer des ressources de calcul et de stockage géographiquement réparties, en vue de permettre leur utilisation de manière transparente pour tout client de la grille. Nous allons maintenant examiner les caractéristiques principales d'une grille de calcul et des ressources qui la composent [10,11] :

- *Hétérogénéité :* Une grille héberge des ressources matérielles et logicielles très différentes les unes des autres. Cette diversité doit être masquée à l'utilisateur de la grille, qui doit pouvoir accéder à ces ressources de manière totalement transparente. Ceci suppose la mise en place de protocoles de communication standardisés, et impose des contraintes de portabilité de code.
- *Partage de ressources :* Les ressources d'une grille peuvent appartenir à différentes organisations. Celles-ci peuvent choisir de partager l'ensemble des

ressources de la grille entre elles, et également d'autoriser d'autres organisations à y accéder. Cela permet de mieux répartir la charge de travail, et d'exploiter les ressources sous-utilisées, promouvant ainsi l'efficacité et la réduction des coûts d'exploitation.

- *Multiplicité des domaines d'administration :* Chaque organisation peut mettre en place des politiques de gestion et de sécurité différentes, en termes d'accès au réseau, d'authentification ou bien encore de confidentialité. Les ressources doivent néanmoins être accessibles et utilisables par tous les clients de la grille. De plus, il est indispensable de fournir des méthodes d'administration particulières aux personnes en charge de la gestion globale de la grille.
- *Passage à l'échelle :* Le nombre de ressources constituant une grille peut varier d'une dizaine à plusieurs milliers. Ce problème de dimensionnement pose de nouvelles contraintes sur les applications et les algorithmes de gestion des ressources. Notons que le passage à l'échelle est également désigné par le terme de scalabilité.
- *Accès transparent, consistant et pervasif :* La grille doit être vue comme un unique ordinateur virtuel. La complexité de la plate-forme sous-jacente est masquée à l'utilisateur. De fait, la grille doit gérer les défaillances éventuelles des ressources, et s'adapter en permanence à l'environnement dynamique qui la constitue. L'accès aux ressources disponibles doit être garanti à l'utilisateur à tout instant.

II. *Architecture d'une grille :*

1. Topologies des grilles de calcul :

Il existe 3 grands types de topologie de grille de calculs :

- L'intragrille :

C'est la plus simple des grilles. Elle est composée d'un ensemble relativement simple de ressources et de services appartenant à une organisation unique. Ses principales caractéristiques sont la présence d'un réseau d'interconnexion performant et haut-débit, d'un domaine de sécurité unique et maîtrisé par les administrateurs de l'organisation et d'un ensemble relativement statique et homogène de ressources.

- L'extragrille :

Une extragrille agrège plusieurs intragrilles. Les principales caractéristiques d'une telle grille sont la présence d'un réseau d'interconnexion hétérogène haut et bas débit (LAN / WAN), de plusieurs domaines de sécurité distincts, et d'un ensemble plus ou moins dynamique de ressources.

- L'intergrille :

Une Intergrille consiste à agréger les grilles de multiples organisations, en une seule grille. Les principales caractéristiques d'une telle grille sont la présence d'un réseau d'interconnexion très hétérogène haut et bas débit (LAN / WAN), de plusieurs domaines de sécurité distincts et ayant parfois des politiques de sécurité différentes et même

contradictoires, et d'un ensemble très dynamique de ressources. Il s'agit la bien sur du modèle le plus pointu à mettre en place !

Bien que chaque projet de Grid Computing ait sa propre architecture logique, une architecture générale en couches est cependant admise pour expliquer certains concepts fondamentaux des grilles.

L'architecture d'une grille est souvent représentée comme un empilement de plusieurs couches (voir Figure 1.2), dont chacune assure une fonction spécifique [1,6]. La figure ci-dessous illustre une telle architecture :

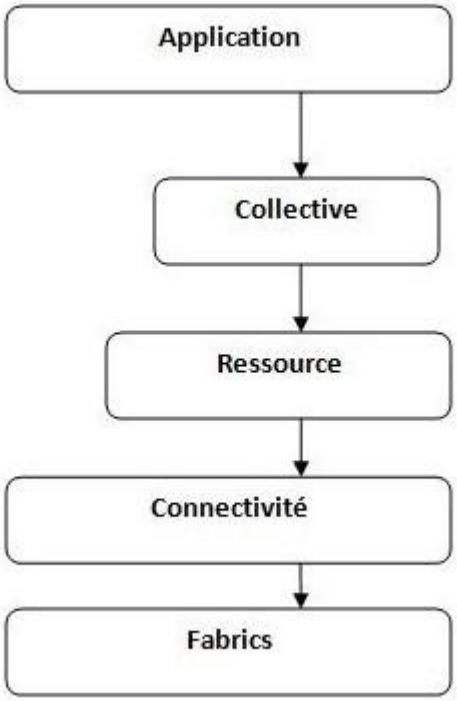


Figure 1.2 : architecture en couches d'une grille

- *La couche Physique (Fabric) :* désigne l'ensemble de l'infrastructure physique d'une grille (ordinateurs, systèmes de stockage, bases de données, etc.). Ces ressources (physiques et logiques) implémentent un certain nombre de mécanismes de demandes d'informations permettant de les découvrir, de connaitre leurs états, leurs capacités et les mécanismes de gestion correspondants.
- *La couche Connectivité (Connectivity) :* implémente les principaux protocoles de communication et de sécurité nécessaires au bon fonctionnement d'une grille.
- *La couche Ressource (Resource) :* utilise les services des deux couches précédentes pour collecter les informations sur les ressources, pour les surveiller, pour les contrôler et pour gérer une éventuelle facturation quant a leur utilisation, au cas où elles sont payantes.
- *La couche Collectif (Collective) :* se charge de la coordination des ressources. Par opposition à la couche Ressource, qui traite les ressources de manière individuelle, la

couche Collectif permet de coordonner l'ensemble de ces ressources. Les services et les protocoles décrits dans cette couche ne sont pas associés à une ressource spécifique mais a un ensemble de ressources. C'est ainsi que cette couche est responsable de l'ordonnancement et de la co-allocation des ressources, lorsqu'un utilisateur fait appel a plusieurs ressources simultanément. Elle est chargée également de la réplication des données.

- *La couche Application :* représente l'ensemble des applications des utilisateurs qui seront exécutées sur une grille, afin de stocker et de récupérer des données ou pour y effectuer des calculs. Dans de telles situations, les différentes couches d'une grille seront alors sollicitées. Par exemple, les couches Collectif et Ressource seront en charge de la recherche des ressources pouvant héberger l'exécution d'une application. Une fois celles-ci trouvées, la couche Connectivité sera utilisée pour une authentification et, enfin, la couche Fabrique sera en charge de l'accès physique à ces ressources.

En plus de la structure définie ci-dessus, il existe d'autres approches de description d'une grille en couches [2]. Une première approche met l'emphase sur les constituants d'une grille du point de vue infrastructure (voir Figure 1.3) [7, 1], alors qu'une deuxième mettra l'accent sur les services offerts par chaque couche (voir Figure 1.4) [4,2].

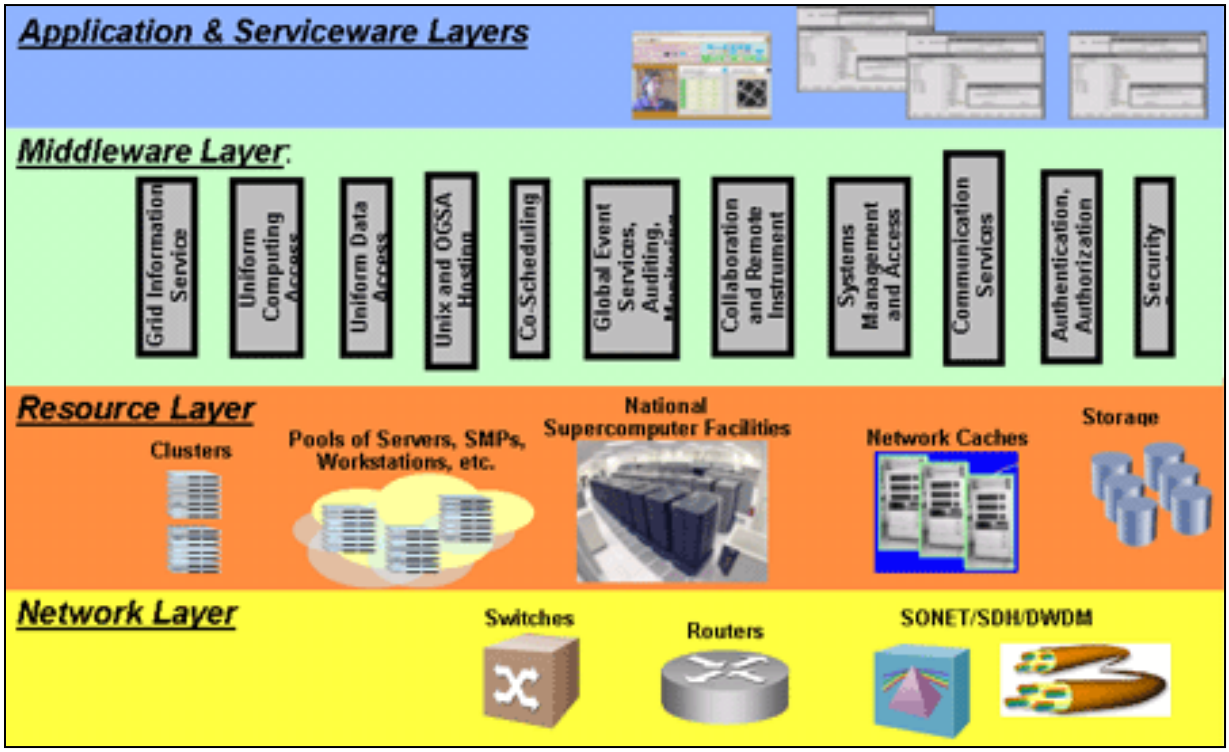


Figure 1.3 : Constituants des couches d'une grille

Application	Application de l'utilisateur
Services Collectifs	Courtage des ressources, surveillance et diagnostic
Protocoles de ressources et de connectivité	Vérification de l'identité des utilisateurs et des ressources
Fabrique	Ressources telles que ordinateurs, systèmes de mémoire, catalogues de données électroniques

Figure 1.4 : Fonctionnalités des couches d'une grille

Une grille doit intégrer un intergiciel (middleware) qui regroupe l'ensemble des services logiciels pour sa mise en œuvre [6,2]. Comme tout middleware, son principal rôle est d'assurer une interface entre les différentes couches de la grille. Un intergiciel comprend notamment les fonctions suivantes [2, 10] :

- Le partage et l'allocation des différentes ressources d'une grille ;
- L'exécution, l'ordonnancement des travaux ainsi que l'administration d'une grille ;
- L'ensemble des procédures de sécurisation d'une grille ;
- Les outils collaboratifs permettant aux divers acteurs de travailler ensemble ;
- Les outils d'évaluation des performances et de mesure de qualité de service ;
- Les outils de développement et les interfaces utilisateurs pour le déploiement d'applications.

2. Les bibliothèques de développement :

Les bibliothèques de développement correspondent en fait à des boîtes à outils open source pour la mise en place d'une grille ou la construction d'outils de grilles. Il ne s'agit nullement de produits prêt à l'emploi.

Ce jeu de protocoles ouverts permet de réaliser les actions élémentaires nécessaires à une grille de calculs tels qu'un système d'authentification, un système de recherche des ressources disponibles et un système de mobilisation de ressources pour l'exécution d'un programme.

La boîte à outil la plus utilisée à ce jour est sans aucun doute le « Globus Toolkit ».

Il s'agit d'un outil open source qui tourne sur des plateformes LINUX, ou Solaris, ou .Net (à partir de la dernière version du Globus Toolkit 3 (GT3); entièrement refondé et basé sur du Java qui n a plus rien à voir avec le GT2). Le Globus Toolkit™, du projet éponyme du Laboratoire d'Argonne, est devenu une architecture de référence.

Trois services peuvent être considérés comme le noyau fondamental de Globus :

- *Le Globus Sécurity Interface :*

C'est une API visant à la mise en place de diverses fonctions de sécurité distribuée. Elle concerne l'authentification des utilisateurs et la garantie d'intégrité et de confidentialité des messages.

- *Le Métacomputing Directory Service :*

Il fournit les informations sur la grille dans un format unique et avec un protocole standard.

- *Le Globus Ressource Access Manager :*

Il permet d'exécuter un programme lorsque le serveur et les clients sont dûment authentifiés et informés quant aux ressources disponibles.

III. Architecture OGSA :

1. Origine de l'architecture OGSA :

L'OGSA est l'architecture de services de Grille. Elle est née du changement de paradigme dans le monde du génie logiciel c'est-à-dire du passage de la vision orientée-système vers la vision orientée-service et plus précisément de service à état [4]. Cette approche, mieux connue sous le nom de Service Oriented Architecture (SOA), définit une architecture où les services sont à la base des communications. L'OGSA est issue du modèle d'architecture qui avait été proposé par deux chercheurs de Californie, Ian Foster et Carl Kesselman [11]. L'idée était qu'une fois les ressources informatiques virtualisées, il est envisageable de mutualiser toutes ou partie de ces ressources pour générer des services. Initialement le service rendu se limitait au calcul intensif gourmand en ressource de traitement et de stockage. L'OGSA est une série de spécifications techniques par lesquelles on définit une infrastructure pour intégrer et gérer les services à l'intérieur d'une organisation virtuelle, distribuée et dynamique [12].

2. Modèle d'architecture OGSA :

La figure suivante est une représentation synthétique des différentes notions essentielles introduites par OGSA. Il est à noter toutefois que, bien que la sécurité occupe une place importante, elle a volontairement été occultée ici afin de se focaliser uniquement ici sur l'aspect lié à la gestion des services. Sur le bas du diagramme, les ressources sont représentées sous forme d'unités qui fournissent de la capacité de traitement (1), stockage (2) et qui sont couplées à travers une infrastructure de réseau TCP/ IP. L'intérieur du rectangle en pointillé (4) au milieu délimite le champ d'OGSA. En premier lieu, les ressources physiques sont virtualisées (5). Toutes les unités de ressource sont additionnées pour former une capacité globale de traitement et de stockage (6). Dans un deuxième temps des

conteneurs de service (7) sont crées dynamiquement avec une portion de ressource réifiée (8).

Les instances de service sont exécutées dans ces conteneurs (9). L'état des services grille (10) est ainsi géré de cette manière. Une communauté virtuelle (11) est composée d'un ou plusieurs utilisateurs appelé membre (12). Un même utilisateur peut être membre de plusieurs communautés virtuelles (13). Les membres d'une communauté virtuelle peuvent accéder et modifier l'état des services (15) propres au conteneur de leur communauté virtuelle selon les droits qui leurs sont attribués (14). Il est à noter que la terminologie employée ici est tirée de la spécification OGSI (Open Grid Service Infrastructure) [OGSI]. Récemment, les concepts OGSI ont été repris un à un avec une nouvelle terminologie issue du groupe de normalisation WSRF (Web service Resource framework) [WSRF]. Ainsi, la notion de service instance correspond à WS- Ressources dans la nouvelle terminologie et la notion d'identifiant de service correspond à WS-Addressing.

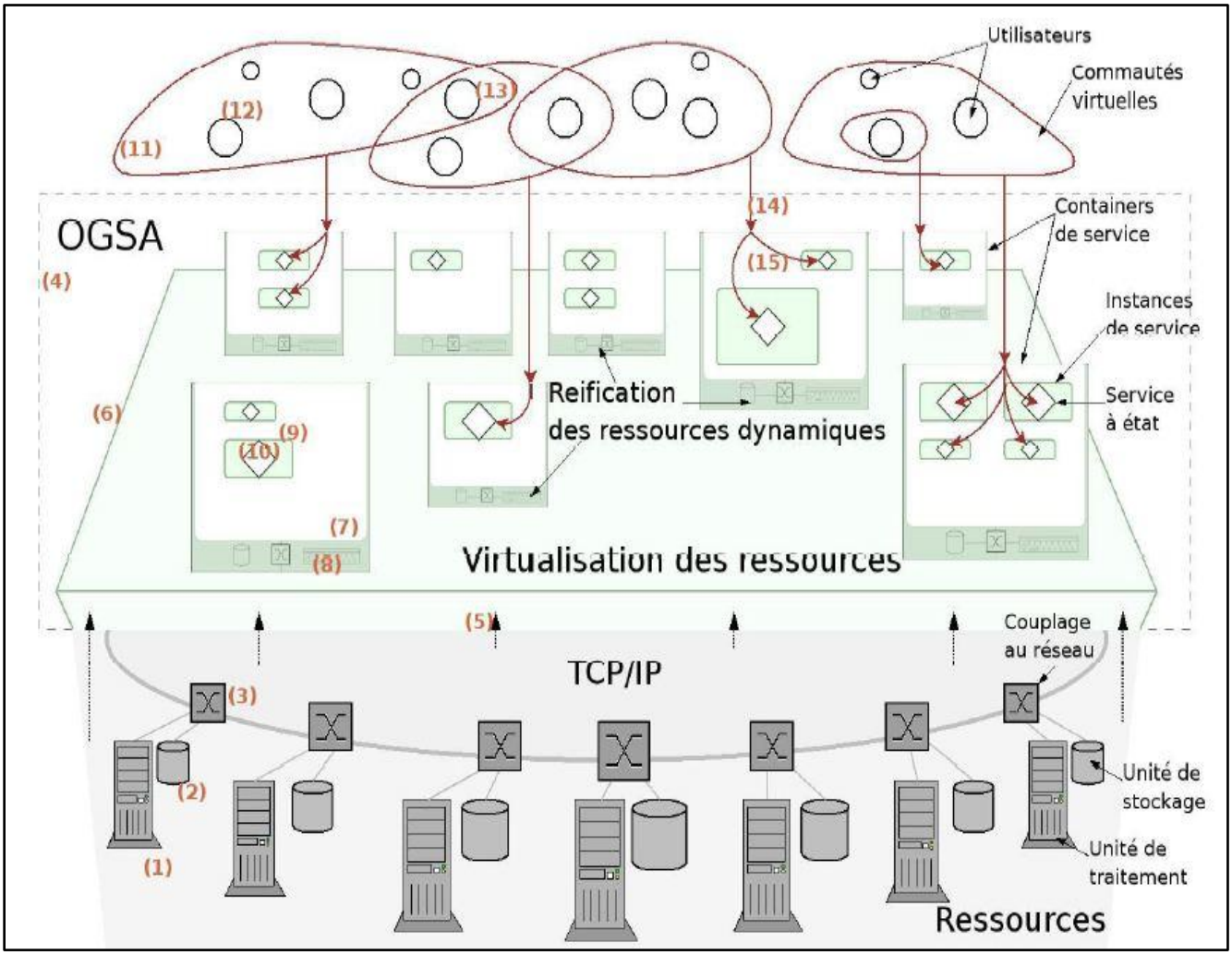


Figure 1.5 : Architecture OGSA

IV. Evolution des grilles

Les grilles de calcul se sont développées au début des années 2000, dès que l'évolution technologique et des tarifs ont rendu abordables les réseaux informatiques académiques à très haute vitesse, supérieure à 1 gigaoctet par seconde. En effet, une telle vitesse permet d'envoyer des données distantes dans un processeur à un rythme voisin de celui de traitement interne des données. Cela permet donc de traiter des données stockées loin avec une efficacité comparable à un traitement local. Ce découplage entre localisation des ressources de calcul et de stockage est une caractéristique essentielle des grilles de calcul.

Les réseaux de communication sont devenus un élément indispensable dans le monde informatique moderne [4, 12, 8]. Ils permettent le partage d'informations et la collaboration entre les personnes et leurs applications (voir Figure 1.6). Depuis la fin des années 90, le développement des réseaux a permis à la technologie des grilles informatiques d'évoluer de manière considérable.

Ainsi, nous pouvons distinguer trois générations des grilles selon leurs apparitions chronologiques dans le temps [4, 12, 13].

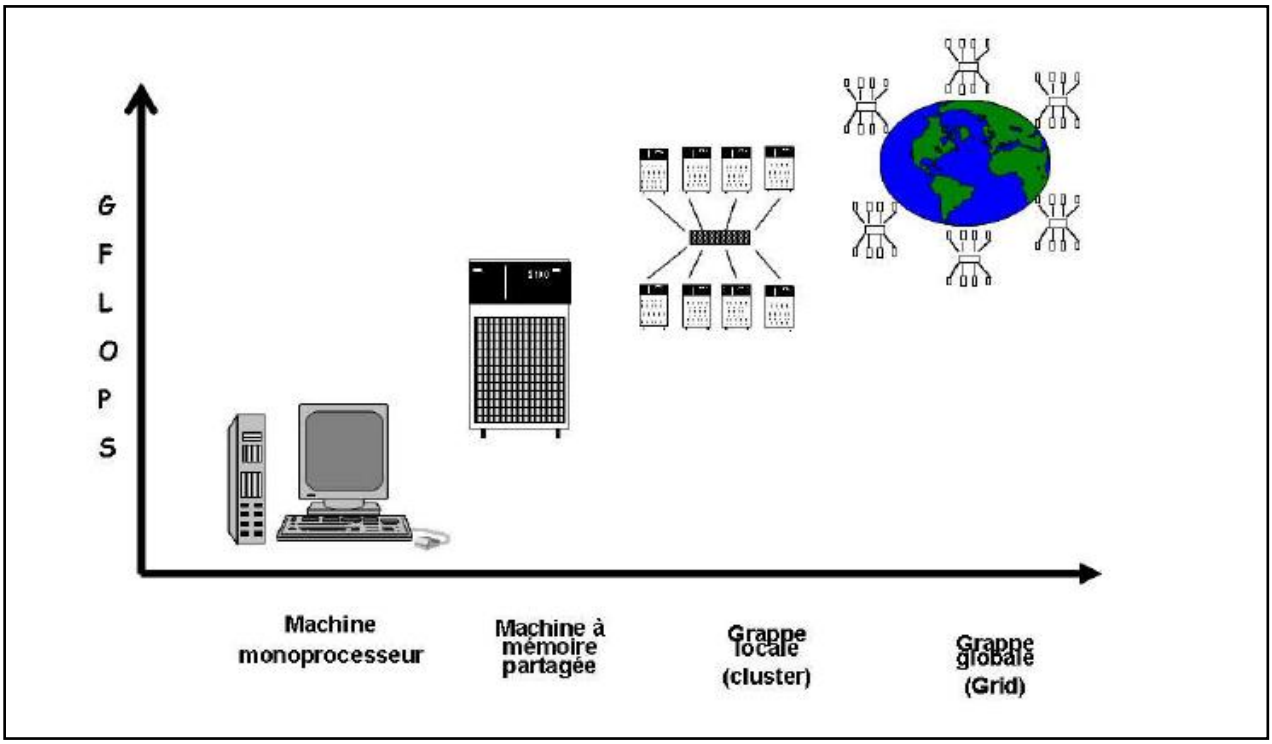


Figure 1.6 : Evolution des grilles

PREMIERE GENERATION :

Le Metacomputing, qui remonte à la fin des années 90, est l'ancêtre immédiat des grilles. Ce terme, appelé aussi Métacalcul, a été défini dans le cadre du projet CASA de la NCSA [2], pour d'écrire le concept de passage à l'échelle massif de l'agrégation des ressources d'un grand nombre de machines individuellement connectées à Internet. Depuis,

le metacomputing a été utilisé pour désigner des projets d'interconnexion de gros centres informatiques aux Etats-Unis, visant à fédérer la puissance de multiples supercalculateurs [4], pour faire face à des problèmes nécessitant de grandes puissances de calcul, comme dans le cas des projets FAFNER [5] et I-WAY [2].

DEUXIEME GENERATION :

Les évolutions technologiques réalisées dans différents domaines, tels que les réseaux, les systèmes Peer-to-Peer, le haut débit et l'adoption de normes informatiques internationales, a permis aux grilles de calcul d'être perçues comme une infrastructure distribuée viable et sure à l'échelle de la planète, qui peut supporter diverses applications qui exigent de fortes densités de calcul et de données.

Du point de vue gestion de ressources, la deuxième génération a apporté un outil essentiel, appelé courtier de ressources (resources broker) [2, 14] schématisé par la Figure 1.7.

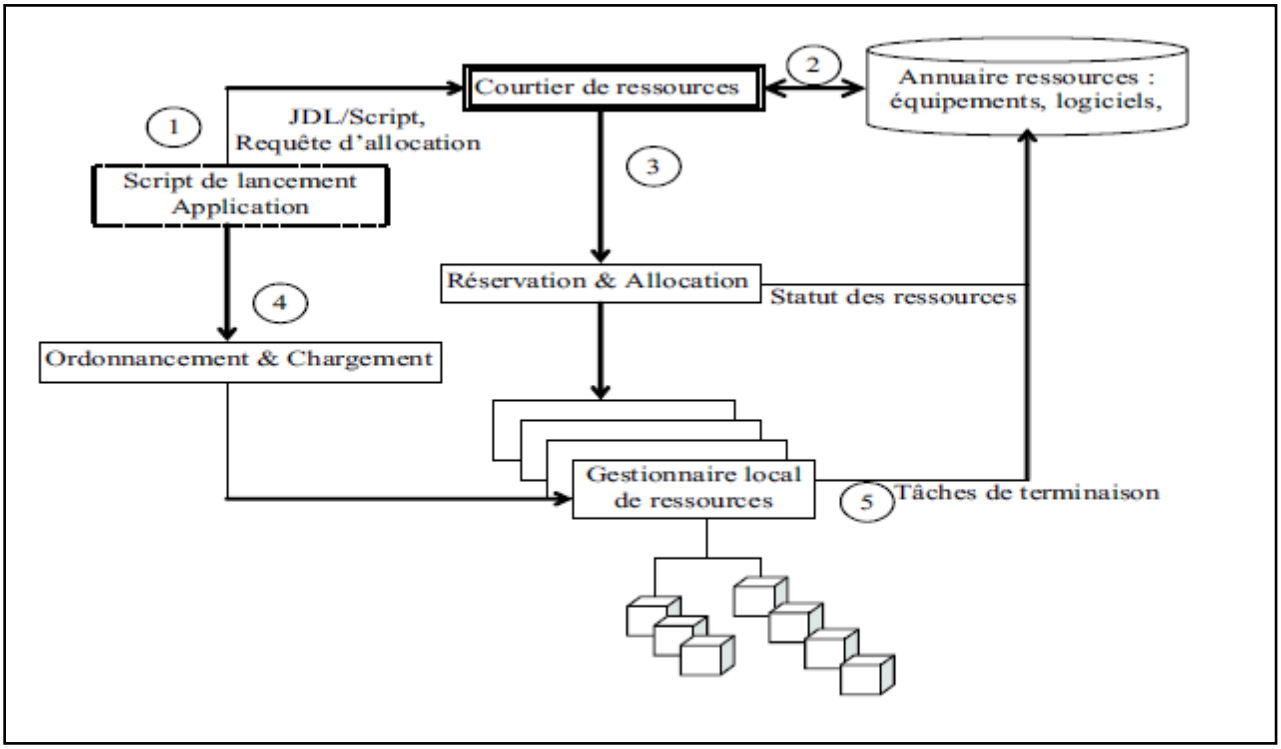


Figure 1.7 : Courtier de ressources

Ce gestionnaire dispose de l'état de l'ensemble des ressources offertes par une grille, ce qui lui permet de faire des allocations à travers des scripts de lancement de travaux, voire de les réserver préalablement, avant ordonnancement puis exécution. La deuxième génération se caractérise par trois principaux aspects [4, 2] :

1. *Hétérogénéité :* Une grille implique une multitude de ressources de nature hétérogène en terme de matériels et de logiciels ;

2. *Passage à l'échelle :* Cet aspect définit la capacité pour une grille de calcul à rester performante malgré une montée en charge des ressources utilisées et du nombre d'utilisateurs. Le maintien de cette performance pose bien évidemment de nouvelles problématiques d'un point de vue gestion des ressources et des utilisateurs ;

3. *Dynamicité :* Les grilles de calcul sont des systèmes à large échelle et dynamiques. La nature dynamique des grilles est à la fois un atout et un défi. Un atout, car elle permet d'ajouter des ressources au fur et à mesure de leur disponibilité et/ou des besoins ou d'en retirer pour effectuer des opérations de maintenance par exemple. Un défi, car pour le concepteur système cet aspect dynamique soulève de nombreux problèmes de gestion [15].

TROISIEME GENERATION :

Les principales caractéristiques de la troisième génération des grilles de calcul peuvent se résumer comme suit [4] :

1. Meilleure sécurité d'accès aux grilles et aux ressources par l'utilisation de certificats électroniques;

2. Une gestion des fichiers adaptée à la prise en charge de données massives sur grille, avec découpage et regroupement en entités logiques pouvant être répliquées en cas de nécessité pour des traitements parallèles ;

3. La généralisation de la notion d'organisation virtuelle [1] permettant de définir des sous ensembles de ressources et leurs modalités d'usage par des communautés d'utilisateurs ;

4. L'introduction de modèles économiques [11] pour l'allocation de ressources ;

5. Le passage d'un mode accès à des ressources d'une grille vers la fourniture de services de grilles (web et grid services) ;

6. L'interopérabilité via une normalisation des services web appliqués aux grilles, tel qu'OGSA (Open Grid Services Architecture) et WSRF (Web Services Resource Framework) [2]

7. L'utilisation de services réseaux, tels que des protocoles plus performants pour le multicast, les transferts massifs de données, l'utilisation de primitives de communication issues de la bibliothèque MPI (Message Passing Interface) [2], l'allocation de bande passante à la demande, etc. ;

8. L'utilisation d'approches basées sur les SLA (Service Level Agreement) [4] au niveau réseau et au niveau grille ;

9. Une meilleure prise en compte des besoins industriels : administration simplifiée, sécurité, facturation des services et business models, interopérabilité des grilles, fiabilisation des services, connexion à la demande à une grille.

V. *Les différents types de grilles :*

Il n'est parfois pas évident de repérer les systèmes de grilles informatiques dans notre monde. Pourtant, il en existe un certain nombre que nous pouvons classer en trois catégories [4] :

1. Les grilles d'information :

C'est sans aucun doute la première incarnation évidente du concept de grille. Elle permet le partage de l'information à travers un réseau. L'exemple le plus caractéristique de ce type de grille est le Web [4], qui permet l'accès à de grandes masses de données disséminées à travers le monde.

2. Les grilles de stockage :

Cette catégorie de grille permet le partage de données externalisées entre plusieurs nœuds. Les cas les plus représentatifs de cette catégorie sont les réseaux d'échange Peer-to-Peer [4, 16]. Les grilles appartenant à cette catégorie permettent l'accès à des données (fichiers, flux, etc.) via un réseau de sites (ou serveurs) qui contiennent et partagent un index. Les données sont référencées pour optimiser les recherches à travers un moteur de recherche. Les fichiers peuvent se trouver sur des nœuds différents du réseau et en différents points du globe. Comme exemple de cette catégorie de grilles, nous pouvons citer notamment le réseau GNUTELLA [2] ou encore KaZaA [4, 2].

3. Les grilles de calcules :

Cette troisième catégorie est l'incarnation des grilles dans le monde informatique, dans le sens où les premières grilles mises en place avaient pour objectif de répondre à des exigences en puissance de calcul, qui ne pouvaient pas être satisfaites par les ressources appartenant à des individus ou à des institutions. Les premières utilisations des grilles de cette catégorie ont consisté à récupérer les heures d'inactivité des processeurs à travers le

monde, via le réseau Internet. Cette approche s'est ensuite généralisée pour mettre en place le principe de regroupement des capacités de calcul de milliers, voire de millions d'ordinateurs, pour répondre à un intérêt commun. Comme exemple pratique de ce principe, nous pouvons signaler la participation d'IBM avec son World Grid Computing[2], qui est un organisme mettant à la disposition des laboratoires de recherche une très grande puissance de calcul de l'ordre du *TeraFlops*.

VI. Modèles de déploiement :

Les applications de calcul distribué ou calcul en grille (Grid Computing) relèvent de la technologie P2P dans la mesure où les ordinateurs qui composent une grille n'adoptent pas une structure centralisée. Chaque client logiciel collecte auprès d'un nœud de répartition la tâche à accomplir, ainsi que la ou les adresses des machines auxquelles il doit renvoyer les résultats. Exigeant une méthode de calcul particulièrement complexe, ces programmes sont limités, pour l'instant, à des applications scientifiques de calcul intensif.

Cependant des variantes existent aussi bien niveau des architectures P2P et donc des grilles qui peuvent adopter un mode plus ou moins centralisé.

1. Modèles centralisés :

1.1. Architecture centralisée pour les grilles de stockage : le peer-to-peer assisté

Le réseau peer to peer le plus connu du grand public est sans doute Napster. Son originalité réside dans le fait qu'il utilise une architecture centralisée ce qui a contribué à son succès mais aussi à sa perte.

Sur le papier, un tel dispositif représente actuellement la solution la plus confortable pour échanger des fichiers dans une communauté (musique, DVD...). Mais dans la réalité, ce type d'architecture exige un tel investissement en ressource que les services restent rarement de bonne qualité (lenteur, disponibilité...). Soit ils sont saturés, soit ils sont limités en termes d'utilisateurs simultanés autorisés.

Concrètement, dans toute architecture centralisée, un dispositif exclusivement serveur se charge de mettre en relation directe tous les utilisateurs connectés. L'intérêt de cette technique réside dans l'indexation centralisée de tous les répertoires et intitulés de fichiers partagés par les abonnés sur le réseau. En général, la mise à jour de cette base s'effectue en temps réel, dès qu'un nouvel utilisateur se connecte ou quitte le service.

Cela fonctionne avec le client comme avec un moteur de recherche classique : vous lancez une requête en inscrivant un mot-clé. Vous obtiendrez une liste d'utilisateurs actuellement connectés au service et dont les fichiers partagés correspondent au terme recherché. Dès lors, il suffit de cliquer sur un des intitulés de lien pour vous connecter directement sur la machine correspondante et entamer le transfert. Dans ces conditions, à aucun moment les fichiers se retrouvent stockés.

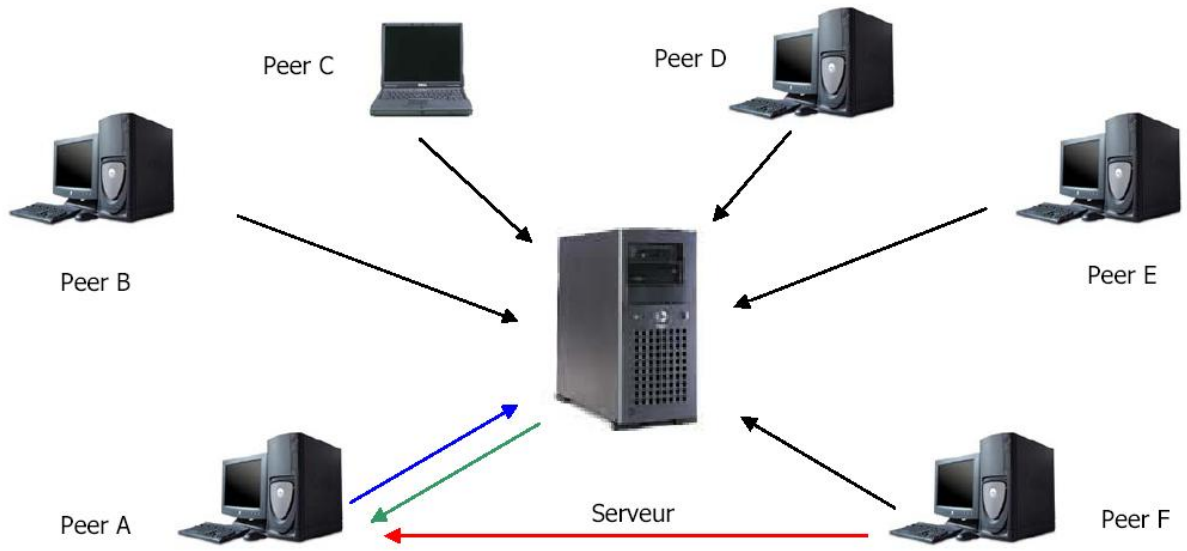


Figure 1.8 : architecture centralisée : le peer-to-peer assisté

- → *Un utilisateur recherche un fichier ressource en envoyant une requête au serveur central.*
- → *Le serveur central répond et transmet la liste des ordinateurs utilisateurs proposant le fichier demandé.*
- → *L'utilisateur télécharge le fichier directement à partir d'un des ordinateurs renseignés par le serveur.*

1.2. Modèle client/serveur pour les grilles de calcul :

1.2.1. L'Internet Computing : le projet SETI

Les premières technologies de grilles de calcul sont apparues il y a quelques années sous le nom de **Calcul Global**. L'application la plus célèbre est SETI@home pour la recherche de signaux extra terrestres. Le projet du décrypthon a utilisé également ce type de grille pour établir la carte des 500 000 protéines du vivant. Entrent dans la même classe, les méthodes utilisant des calculateurs répartis pour casser des codes de cryptographie (RSA-155). Le principe est le suivant : partant du constat que le taux d'utilisation des ressources de l'Internet est très faible (nuit, week end, travail interactif), la problématique est alors d'être capable d'utiliser la masse des ressources inactives pour des traitements aujourd'hui inaccessibles. L'idée directrice du travail est que ces ressources doivent être accessibles instantanément comme l'électricité aux divers appareils "branchés" en permanence sur l'Internet directement ou via des "réseaux ambiants" : PC, téléphone, palm, portable, véhicule, etc. Pour cela un serveur distribue des données à des ordinateurs clients ; chaque ordinateur récupère une donnée et exécute un programme de traitement et envoie le résultat au serveur ; le serveur rassemble les résultats, relance de nouvelles données vers les ordinateurs et en tire les informations pertinentes.

Un des points clefs est la conception d'un protocole automatique (plug&run) de découverte/localisation/acquisition de ressources et services disponibles dans ce réseau global. Enfin, les applications distribuées sur ces ensembles de ressources doivent tolérer les déconnexions intempestives, ce qui nécessite des modèles de programmation et des supports exécutifs spécifiques.

Pour le mode d'exécution du logiciel client, le plus répandu reste celui de l'économiseur d'écran comme *SETI@home* (lequel dispose aussi d'un modèle en tâche de fond).

Les projets de calcul global tels que *seti@home*, Entropia, ou COSM offrent essentiellement les capacités de calcul des ordinateurs volontaires, souvent des PC, à des applications à parallélisme trivial, c'est-à-dire sans communications entre les PC participants. L'utilisabilité de ces systèmes est très précaire et l'accès aux ressources très limité. C'est le système de calcul global qui lance et gère les exécutions et seul le propriétaire du système peut soumettre des tâches à exécuter. Ces systèmes sont donc très fermés et les participants n'ont que la possibilité d'offrir leur(s) ressource(s) de calcul. D'un autre coté, les systèmes Pair à Pair comme Napster, Gnutella ou Freenet proposent d'exploiter les ressources de stockage de PC volontaires. L'accès est généralement autorisé à tous les PC participants.

Cependant, aucun système Pair à Pair ne propose une infrastructure permettant de lancer des calculs sur les PC participants et c'est une des directives de recherche développée au niveau des grilles de calcul.

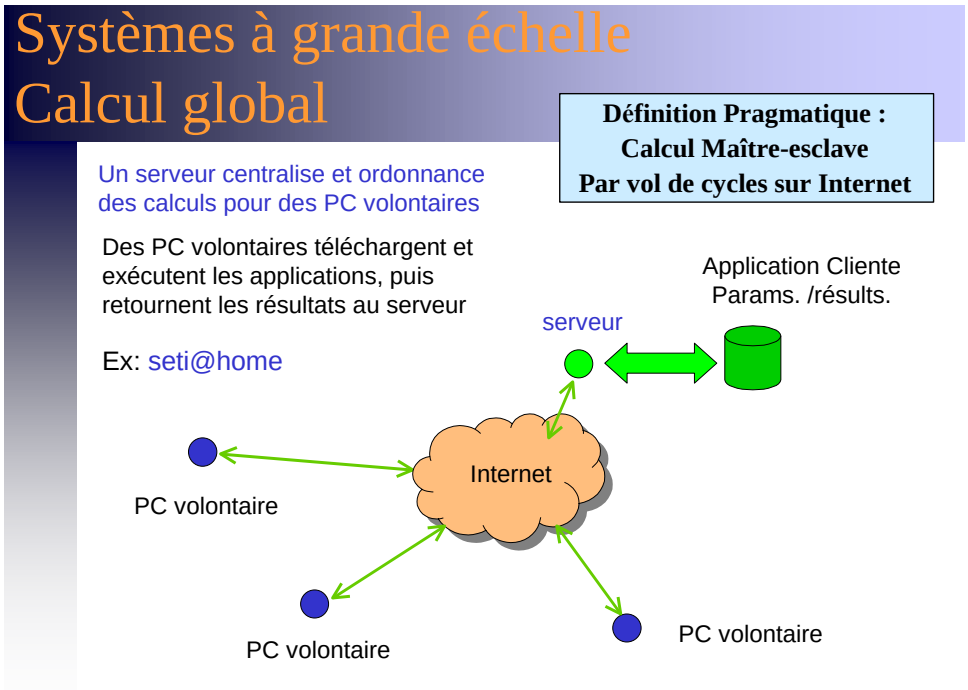


Figure 1.9 : calcul global

1.2.2 Metacomputing : le projet NetSolve

Le projet NetSolve a été développé à l'Université du Tennessee et au Laboratoire National d'Oak Ridge (*Oak Ridge National Laboratory*). La première version publique de

NetSolve a été publiée en mars 1997 [CAS 97]. Il a été initialement conçu dans le but de permettre une utilisation partagée de programmes de calcul scientifique s'exécutant sur des ordinateurs dédiés, NetSolve est basé sur le modèle Client/Serveur. Le client ne connaît pas la localisation de la machine sensée exécuter son programme.

Un système NetSolve est composé d'un ensemble de services pouvant être appelés par des machines clientes. Chaque service est hébergé et exécuté au niveau d'un ou de plusieurs nœuds serveurs du réseau. Ces services sont répertoriés dans des « bases de données » réparties appelées *Agent NetSolve*. Un service est représenté par son nom, par son emplacement (adresses des machines qui l'exécutent) ainsi que par d'autres informations détaillées dans le paragraphe suivant. Un même service peut être inscrit auprès de plusieurs Agents. Par ailleurs, un service peut exister sous plusieurs versions sur plusieurs machines et être répertorié par le même Agent. Dans ce cas, plusieurs programmes représentent le même service.

Pour pouvoir exécuter un service NetSolve, une machine cliente doit être inscrite auprès d'un Agent. Ceci signifie que le client connaît l'adresse de l'Agent. Le service demandé doit être catalogué auprès du même Agent. A la réception d'une requête client (demande d'exécution d'un service), l'Agent NetSolve consulte sa base de données pour déterminer les machines susceptibles de répondre à la requête, calcule leur charge (*workload)* et retient la moins chargée comme serveur « élu ». Le calcul de la charge d'un serveur est détaillé dans le paragraphe qui suit. Notons enfin qu'un Agent NetSolve ne dispose que d'une vue limitée et statique de la totalité du système, sa base de données ne contient, en général, qu'un sous ensemble des services disponibles. La figure 1.10 illustre l'architecture de NetSolve [17]

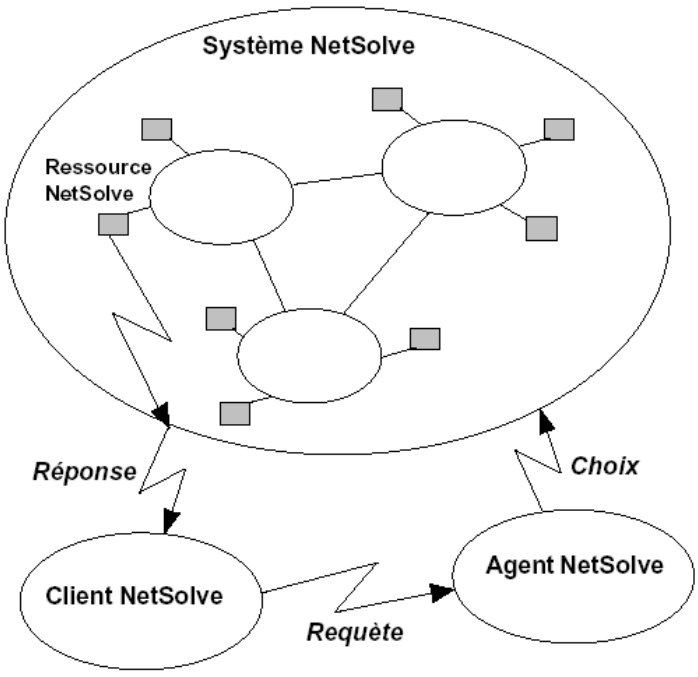


Figure 1.10 : L'architecture de NetSolve

L'ajout et/ou la suppression d'un service au niveau d'un Agent se fait manuellement. L'appel d'un service NetSolve peut se faire interactivement ou à partir d'un programme. Des librairies de routines sont disponibles à cet effet pour plusieurs langages tels que C, Fortran, JAVA, MATLAB, MATHEMATICA.

Ces appels peuvent être bloquants (attente de retour du résultat) ou non bloquants. Le même service peut être exécuté en parallèle sur des données différentes : c'est le *Process Farming* Ceci revient à exécuter en parallèle les (ou une partie des) programmes représentant le service concerné. Chaque programme s'exécute sur son propre serveur. La gestion de la charge des serveurs permet un déploiement optimisé du modèle SPMD (*Single Program Multiple Data*).

2. Modèles décentralisés ou distribués :

2.1. Architecture décentralisée pour les grilles de stockage : le pur peer to peer

L'architecture centralisée pose des problèmes de sécurité, robustesse, et de limitation de la bande passante. Les problèmes sont directement issus de l'utilisation de serveurs dont le seul but est de posséder l'annuaire des clients.

Si on désire supprimer les serveurs centraux il faut donc trouver le moyen de constituer un annuaire sur chaque client, puis de les faire communiquer. C'est sur ces mécanismes que sont basés les réseaux Peer to Peer décentralisés (figure 1.11). Il n'y a donc plus de serveurs centraux, ce sont tous les éléments du réseau qui vont jouer ce rôle. Chaque machine dans ses rôles est identique à une autre, c'est pour cela que l'on appelle ces types de réseaux **pur peer to peer**.

Un grand avantage de ce nouveau type de réseaux, est en théorie le total anonymat qu'il procure. En effet en évitant de communiquer avec une machine centralisant les demandes et les annuaires, on évite les problèmes de récupération des données utilisateurs.

Exemple de réseau peer-to-peer de ce type accessible via Internet : Gnutella (http://www.gnutella.com)

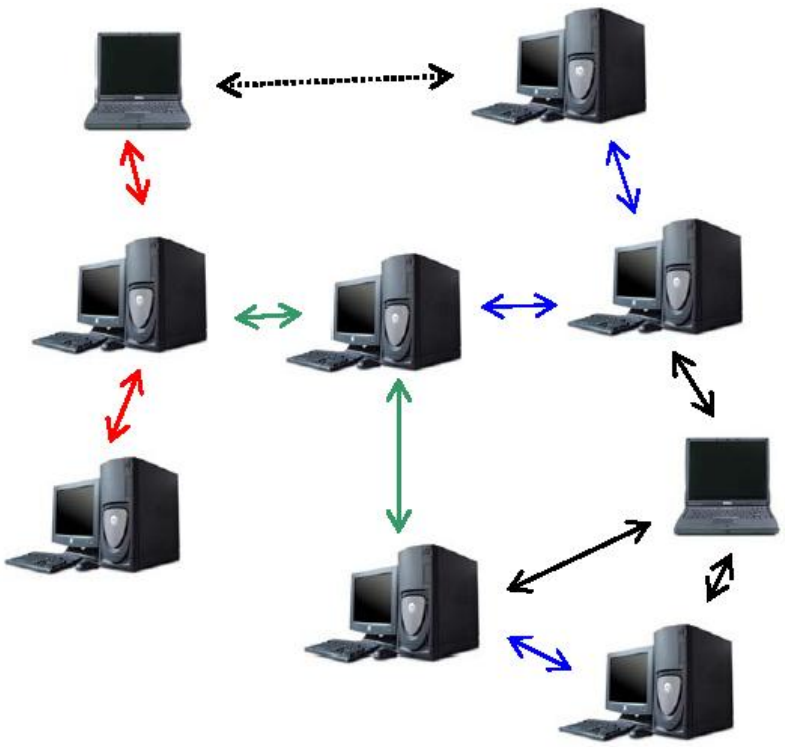


Figure 1.11 : Architecture décentralisée : le pur peer-to-peer

2.2. Vers une architecture distribuée pour les grilles de calcul :

De véritables grilles informatiques, faisant coopérer leurs différents nœuds de façon transparente, commode et sur un pied d'égalité, n'existent que sous forme de prototypes et sont encore loin d'être utilisées en production.

Aux Etats-Unis, par exemple, le système Globus est développé pour relier ensemble des supercalculateurs américains et construire ainsi un calculateur parallèle encore plus puissant.

Dans ce cas, il ne s'agit plus de technologies client-serveur puisque tous les ordinateurs peuvent lancer des calculs et utiliser la puissance de tous les autres. Les applications traitées sont d'une plus grande difficulté de mise en œuvre. Les projets européens DataGrid et EuroGrid s'appuient sur ce type de technologie.

3. Modèles hybrides :

3.1. Un modèle hybride pour les grilles de stockage : les réseaux Super Nœuds

Le modèle super nœud a pour but d'utiliser les avantages des 2 types de réseaux (centralisé et décentralisé). En effet sa structure permet de diminuer le nombre de connexions sur chaque serveur, et ainsi d'éviter les problèmes de bandes passantes.

D'autre part le réseau de serveurs utilise un mécanisme issu des réseaux décentralisés pour tenir à jour un annuaire client et un index des fichiers à partir des informations provenant des autres serveurs. Un serveur peut donc proposer à n'importe quel client toutes les informations contenues sur le réseau.

Le réseau n'est plus pollué par les trames de broadcast. Mais la contrepartie est que l'anonymat n'est plus assuré.

Pour exemple le réseau e-Donkey [18] est né en septembre 2000 mais c'est au cours de l'année 2001 qu'il a connu son véritable essor. Par rapport à ses prédécesseurs, e-Donkey fonctionne dans un mode décentralisé avec une multitude de serveurs : l'architecture est proche du super noeud. Au niveau applicatif e-Mule remplace peu à peu e-Donkey chez les clients Windows grâce à ses nombreuses fonctionnalités comme une gestion des recherches plus poussées et une gestion automatique des files d'attente et des priorités. [19]

3.2. Une architecture hybride pour les grilles de calcul : XtremWeb

XtremWeb est un système P2P destiné au calcul de haute performance. Ce système appartient à la famille des « Desktop Grid ». L'architecture 3 tiers d'XtremWeb (figure 1.12) considère : un Client qui soumet des requêtes et collecte des résultats, un Coordinateur (le serveur) qui se charge de traduire les requêtes client en tâches lancées sur des Workers et des Workers qui exécutent les tâches et retournent leurs résultats vers le Coordinateur. Le système est complètement tolérant aux pannes Client, Coordinateur et Worker. L'architecture actuelle est centralisée.

Le système est un modèle d'interaction entre le modèle du Global Computing et du Peer to Peer (modèles d'interaction). XtremWeb ne vise pas à construire une plate-forme avec des millions d'utilisateurs (SETI, Napster, Gnutella, Freenet, etc.). XtremWeb est beaucoup plus un environnement pour construire des clusters par dessus Internet ou à l'intérieur d'un réseau propriétaire multi-sites.

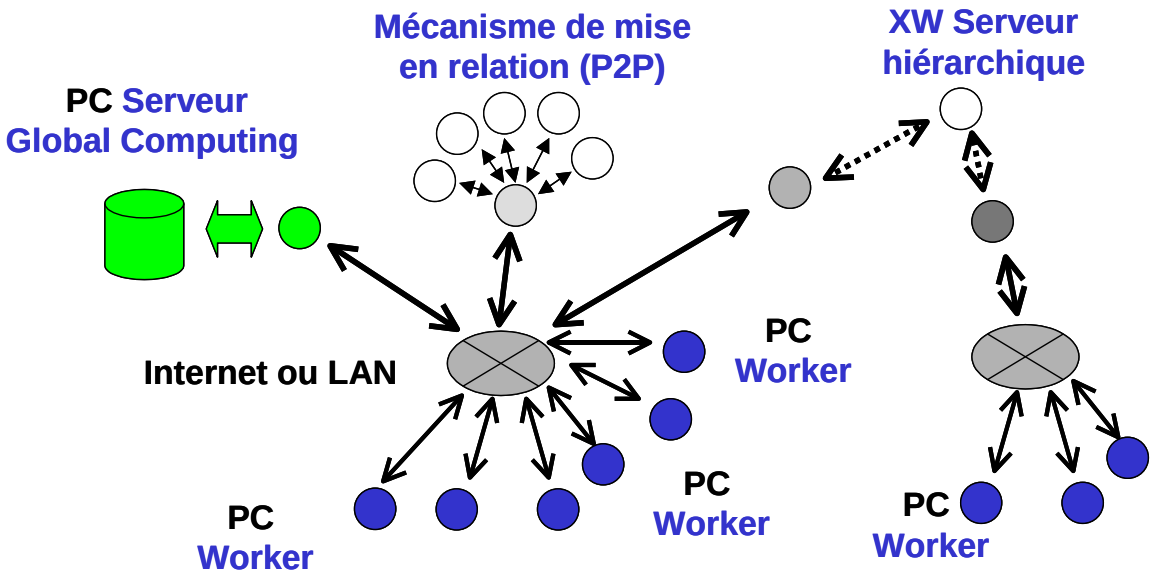


Figure 1.12 : L'architecture d'XtremWeb avec ces 3 entités : client/server/worker

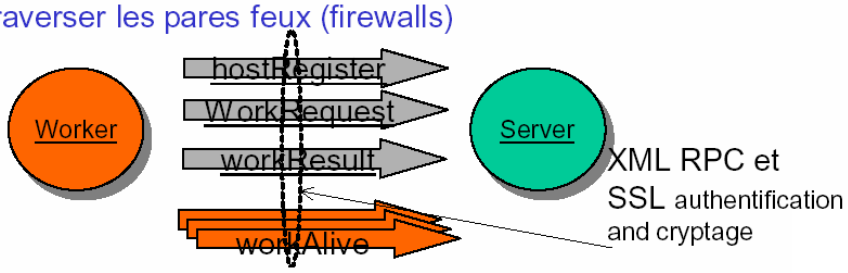
- ***Le Worker :*** C'est l'entité qui exécute des calculs. Lorsqu'une phase de sous utilisation système est ressentie, le PC s'enregistre auprès du serveur comme étant apte à lancer des calculs.
  Toutefois, l'application Java worker.jar doit être lancée sur chacun des PCs destinés à être utilisés sur le parc de Grid Computing. Les applications ciblées pour être exécuté par les workers XtremWeb doivent être qualifiées et enregistrées dans la base de données du serveur.
  Ainsi, une application compilée sous Windows ne fonctionnera qu'avec des workers Windows. Si aucun worker Windows n'est susceptible de lancer le calcul, alors le job sera mis en file d'attente.

Applications → Binaire (codes CHP en Fortran ou C)
→ Java (codes récents, codes objet)

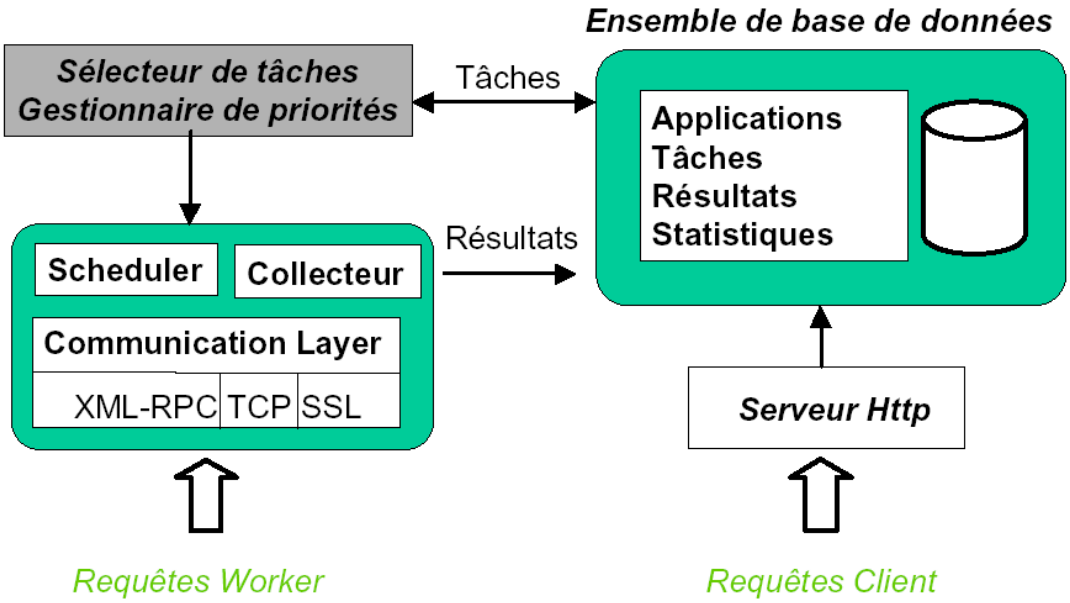
Systèmes d'exploitation → Linux
→ Windows

Mode d'opération → Screen saver
→ contrôle à distance du serveur (on/off)

Protocol: traverser les pares feux (firewalls)

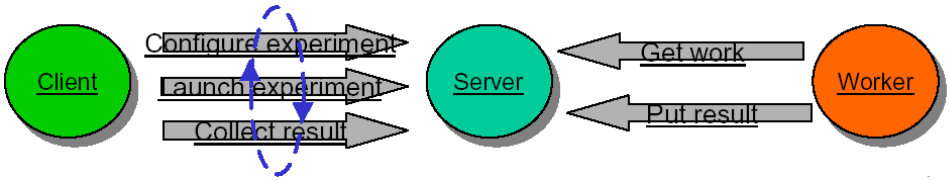


- *Le serveur :* Comme on peut le constater, le serveur XtremWeb utilise un serveur Http (Apache), un serveur Mysql, un scheduler ainsi que des librairies utilisées pour la sécurité des communications. C'est à lui qu'incombe le monitoring des ressources, le lancement de jobs, et la récupération des résultats.



- *Le client :* En fait, un client à l'instant t était un worker à l'instant (t-1). Nous pouvons donc assimiler un client à un worker qui lance un job.
  Le client aura le pouvoir de soumettre un job (si l'application relative au job est qualifiée et enregistrée au sein de la base de données du serveur XtremWeb), de suivre son déroulement et de récupérer son résultat du job.

Une API Java	→ soumission de tâches → collection de résultats → contrôle du serveur
Applications	→ Multi-paramêtres, sac de tâches → Maître-esclave (iteratif), EP
Contrôle du serveur	→ configuration de la plate-forme (nombre de Workers, type de Workers, etc.)



VII. Principaux enjeux des grilles :

Les grilles de calcul tendent à devenir une technologie de besoins de type Business-to-Business [2] et regroupe des personnes ayant un intérêt commun par l'agrégation et l'unification de leurs ressources [10]. Ce regroupement impose un certain nombre d'enjeux que les grilles de calcul doivent être en mesure de satisfaire. Ces enjeux peuvent se résumer sous la forme de cinq critères à satisfaire [2, 9] :

1. Partage de ressources :

Ce critère permet d'accéder à une grille pour partager l'utilisation de ressources distantes. Cette capacité de partage implique plus qu'un simple transfert de fichiers, dans la mesure où elle requiert un accès direct à différents logiciels, différents ordinateurs, différentes données, différents équipements, etc. Malgré leur mise en commun, les ressources d'une grille restent la propriété des personnes ou des institutions auxquelles elles appartiennent. Ceci signifie qu'elles relèvent de domaines administratifs différents, qu'elles exécutent des logiciels différents et qu'elles sont régies par des politiques de sécurité et de contrôle d'accès également différentes.

Accès sécurisé :

Une conséquence du partage des ressources est le contrôle des accès à ces ressources pour garantir un certain nombre de règles de sécurité [4, 2] :

- *Politique concernant l'accès :* les fournisseurs et les utilisateurs des ressources doivent définir clairement et soigneusement ce qui est partagé, qui est autorisé à partager des ressources ainsi que les conditions dans lesquelles aura lieu ce partage ;
- *Authentification :* afin d'éviter toute utilisation abusive et non autorisée de ressources partagées, il est indispensable de définir un mécanisme permettant d'établir, de manière précise, l'identité de l'utilisateur d'une ressource ;
- *Autorisation :* en plus d'une authentification, il faut également définir un mécanisme permettant de déterminer si une opération est conforme aux règles de partage qui ont été définies.

2. Utilisation efficace des ressources :

Ce troisième critère permet d'optimiser l'utilisation des ressources de la grille, même si leur nombre et leurs capacités dépassent les besoins des utilisateurs. A cet effet, il est indispensable non seulement d'assurer que les ressources sont utilisées de manière efficace (exploitation de leurs potentialités) mais également que la charge de travail est bien équilibrée entre les ressources, afin d'éviter que des ressources soient surchargées alors que d'autres sont sous-utilisées ou complètement libres.

3. Abolition de la distance :

Les connexions à haut débit entre ordinateurs rendent possible la mise en place d'une grille à l'échelle mondiale [2]. Il s'agit en fait de donner l'impression que la distance qui sépare un utilisateur des ressources qu'il veut utiliser est pratiquement nulle. Ceci nécessite bien évidemment d'autres avancées au niveau des réseaux de communication.

4. Mise en place de normes :

Les grilles de calcul ayant tendance à se généraliser, il devient important de standardiser une architecture pour les grilles. Cette standardisation permettra de contribuer, de manière significative, à la résolution des problèmes d'hétérogénéité qui se posent au niveau des grilles [10, 15]. Beaucoup de travaux sont menés actuellement dans ce sens, et des normes spécifiques aux grilles sont en cours d'élaboration par un organisme de

normalisation, appelé GGF (Global Grid Forum) [2]. Fédérant plus de 5000 chercheurs et praticiens individuels, cet organisme représente une force significative en matière d'édition de normes et d'élaboration d'éléments permettant le travail en commun. Actuellement, une norme, connue sous le nom OGSA (Open Grid Services Architecture) [2], a été mise en place et elle est considérée comme une référence clé pour les projets d'élaboration de grilles.

VIII. Domaines d'application des grilles :

De part les potentialités qu'elle offre, une grille est un excellent moyen de partager des données et de la puissance de calcul. Ce principe de mise en commun des ressources est utilisé dans une multitude de projets industriels et scientifiques [2]. Les domaines d'utilisation des grilles étant très diversifiés, nous pouvons quand même les regrouper en cinq grandes classes d'applications dans lesquelles les grilles peuvent être d'un apport très substantiel.

1. Calcul distribué :

Les applications de calcul distribué sont évidemment d'excellentes candidates pour être exploitées sur une grille. Elles bénéficient ainsi d'un nombre beaucoup plus important de ressources de calcul leur permettant de résoudre des problèmes qui leur étaient auparavant inaccessibles [4, 5]. Pour certains types d'applications relevant de domaines tels que la physique nucléaire, la recherche de séquence d'ADN, il est évidemment illusoire d'espérer disposer de la puissance de calcul sur un seul et même site. L'utilisation d'une grille se révèle dès lors indispensable [2]. Parmi les principaux défis que doit relever une architecture de type grille pour de telles applications, nous pouvons citer [10] :

- L'ordonnancement à grande échelle de travaux, de jobs et de processus ;
- La souplesse des algorithmes et protocoles qui doivent être capables de gérer un nombre de nœuds pouvant allé de la dizaine à des centaines voire des dizaines de milliers de machines ;
- La tolérance des algorithmes aux temps de latence inhérents à la taille d'une grille ;
- La possibilité d'atteindre et de maintenir un haut niveau de performance dans une infrastructure très hétérogène.

2. Calcul à haut débit :

Contrairement aux calculs distribués, les applications de calcul à haut débit nécessitent la résolution d'un très grand nombre de calculs de manière indépendante et parfois coordonnée.

Les ordinateurs, lorsqu'ils sont inutilisés, peuvent ainsi contribuer à agréger une puissance de calcul pour résoudre des applications très exigeantes en matière de puissance de calcul. Le projet SETI@home [2] est un exemple d'infrastructure qui a permis d'agréger une puissance de calcul très importante uniquement à partir des périodes d'inactivité des processeurs de simples ordinateurs de type PC.

3. Calcul à la demande :

Ce type d'applications nécessite de la puissance de calcul pour des besoins ponctuels. Plutôt que d'acheter des ressources qui ne seront utilisées qu'occasionnellement ou pour un objectif bien précis, il serait plus judicieux de pouvoir utiliser, en cas de besoin, les ressources d'autres utilisateurs notamment pendant la période de leur inactivité [7].

Pour rendre cette nouvelle forme d'utilisation des ressources possible, il est primordial de développer des systèmes performants et sécurisés d'utilisation des ressources, même dans le cas ou elles sont payantes [2].

4. Traitement massif de données :

Dans ce type d'applications, le but est d'extraire de nouvelles informations à partir de grandes bases de données géographiquement distribuées (fouilles de données, bio-informatique, météorologie, etc.). Généralement, ces types de traitement sont également de grands consommateurs de puissance de calcul et de bande passante [2]. Les principaux problèmes à résoudre, pour ce type d'application, sont l'acheminement et la configuration deux de données très importants à travers les éléments d'une grille.

5. Travail collaboratif :

Les grilles de calcul permettent non seulement de regrouper (virtuellement) des ressources informatiques hétérogènes, mais aussi de fédérer des milliers de personnes réparties sur des entreprises et institutions différentes, dans le but d'un travail collaboratif [11]. Dans ce contexte, les grilles, de part leurs caractéristiques, constituent une véritable plate-forme de travail collaboratif à moindre coût.

IX. Avantages et inconvénients des grilles de calcul :

1. Les avantages :

Les avantages d'utiliser une telle architecture sont multiples et indéniables. Nous pouvons citer les exemples suivants :

- *Utilisation des ressources inutilisées la nuit :*

Le grid computing est un concept au fort potentiel, dont l'idée est de faire en sorte que toute la puissance de calcul des PCs inutilisés soit utilisée. De nos jours, les ordinateurs restent souvent inutilisés pendant de longues périodes, leur processeur n'étant que rarement utilisé à 100%.

Avec cette technologie, les moments d'inactivité de centaines ou de milliers d'ordinateurs et de serveurs peuvent être utilisés et vendus à quiconque ayant besoin d'une puissance de calculs massive.

Prenons le cas d'une grande banque, dont les ordinateurs ne fonctionnent pratiquement pas la nuit. Leur puissance de calcul peut être utilisée par une autre société située à l'autre bout du monde (bénéficiant du décalage horaire) pour traiter différentes tâches. En contrepartie, la banque pourrait avoir accès à des capacités supplémentaires lorsqu'elle en a besoin.

- *Basé sur une architecture sûre de type client/serveur :*

Le grid computing doit reposer sur une architecture bien précise et très sûre, en l'occurrence, c'est l'architecture client serveur qui a été choisie. Cette architecture qui est également utilisée dans de nombreux protocoles réseaux, a du être adaptée en fonction des besoins spécifiques de cette technologie.

- *Meilleure rentabilisation du matériel :*

Il est évident qu'il y a une sous utilisation des machines, et le grid computing présente la solution idéale, d'un point de vue économique pour les entreprises et d'un point de vue pratique pour les utilisateurs, pour rentabiliser les ressources.

- *Allègement des charges des calculateurs :*

Les entreprises qui possèdent déjà des supercalculateurs, pourront ainsi alléger les charges de ces derniers en lançant des calculs sur une architecture basée sur le grid computing. Mais il ne faut pas se voiler la face, les technologies de grille ne sont pas encore arrivées à un niveau de maturité suffisant pour s'imposer systématiquement et remplacer les supercalculateurs.

2. Les inconvénients :

Derrière cette idée de mutualiser les temps processeurs inutilisés pour exécuter des calculs distribués se cachent tout de même un certain nombre d'inconvénients et de contraintes :

- Les applications doivent être qualifiées et qualifiables pour tourner sur une telle architecture.
- Le faible niveau de sécurité dissuade les entreprises de se lancer dans un tel projet.
- La lenteur des temps d'accès interdit de rebondir en quelques fractions de secondes vers une autre opération.
- Ce genre de système distribué possède une faible tolérance aux pannes.
- Un changement possible des méthodes de travail pour le lancement des calculs rend réticent les futurs utilisateurs.
- Cette technologie étant très récente, les développeurs et les intégrateurs de systèmes de grilles n'ont pas une vision post-installation assez pertinente pour pouvoir tirer des conclusions sur cette technologie.

Ces différentes faiblesses sont essentiellement dues la cohabitation de plusieurs standards incompatibles s'appuyant sur des langages et des protocoles différents.

X. Conclusion :

Les besoins en puissance de calcul pour la recherche scientifique fondamentale dépassent souvent les possibilités qu'offre la technologie. Il faut donc inventer constamment de nouvelles solutions pour permettre à la recherche de disposer d'outils adaptés. Et la performance des processeurs n'évoluant pas suffisamment vite par rapport aux besoins, l'axe de progrès en la matière passe par la multiplication des unités de traitements. La grille de calcul reprend l'idée qu'une application lourde puisse être découpée en petites tâches isolées, confiées à des ordinateurs différents au travers d'un réseau. L'aspect économique est particulièrement séduisant puisqu'il s'agit d'utiliser la puissance de calcul et les espaces de stockage inutilisés des ordinateurs d'un parc informatique. Mais avant de voir se généraliser les grilles de calcul, il faudra que les mentalités évoluent encore de manière significative. Pour les décideurs, cette technologie n'est pas encore suffisamment mature et les besoins de l'industrie peuvent encore être satisfaits avec la technologie actuelle.

L'un des objectifs essentiels des grilles de calcul est de donner l'impression, à un utilisateur, d'utiliser un super-calculateur virtuel qui permet de connecter une très large collection de systèmes hétérogènes à travers le réseau Internet. Ainsi, une grille englobe un certain nombre de ressources et de services qui sont nécessaires à des applications qui exigent une forte puissance de calcul, ou à celles qui exigent une masse très importante de données distribuées. Ces ressources et ces services sont utilisés à travers un middleware qui assure une transparence totale de la

Dans les grilles de calcul, le partage de données est complexe. Cela est principalement dû à :

(i) la grande échelle ; et, (ii) la nature dynamique des grilles de calcul.

Ces deux caractéristiques rendent impossible une connaissance globale de l'état d'une grille, car le nombre de nœuds peut être très important, et de plus, les nœuds sont susceptibles de quitter ou de rejoindre la grille à tout instant.

Dans ce chapitre, nous avons mis en évidence les principales caractéristiques des grilles selon leurs types, leurs domaines d'utilisation ainsi que les différentes problématiques qu'elles posent pour leur mise en œuvre.

CHAPITRE II

Data Mining et Confidentialité

I. Introduction :

Les technologies de data mining permettent, grâce aux processus d'intelligence artificielle, de traiter des masses gigantesques de données afin d'en extraire l'information cruciale (connaissance), celle qui sera déterminante pour une prise de décision efficace. Une connaissance est définie par un ensemble de relations (règles, phénomènes, exceptions, tendances…) entre les données.

Le data mining est apparu au début des années 1990. Cette émergence est le résultat de la combinaison de plusieurs facteurs à la fois technologiques, économiques et même sociopolitiques. Les volumes gigantesques de données constituent, dès lors, des mines d'informations stratégiques aussi bien pour les décideurs que pour les utilisateurs.

Le data mining est l'un des maillons de la chaine de traitement pour la découverte de connaissances à partir de données plus connu sous le nom d'Extraction de Connaissances à partir de Données (ECD). Sous forme imagée, on pourrait dire que l'ECD est un véhicule dont le data mining est le moteur [23].

Les nouvelles technologies de l'information permettent à la fois un stockage de larges volumes de données et une contribution à leur croissance exponentielle. Ceci a pour conséquence de créer une disproportion entre les volumes de données et les moyens matériels et humains pour les traiter. Face à ces problèmes, les technologies du datamining permettent de traiter ces ensembles de données en vue d'extraire de la connaissance, qui sera déterminante pour la prise de décisions efficaces.

Cependant l'application de ces technologies sur des données à caractère personnel permet de définir des profils d'individus, de prédire leurs comportements et d'agir en conséquence. Or, une telle connaissance se heurte aux droits des personnes qui veulent préserver les données relatives à leur vie privée.

Le problème de confidentialité peut être adapté à de nombreux domaines, comme par exemple l'analyse de transactions financières, l'analyse de comportements sur des sites de e-commerce, etc.

Dans ce début de chapitre, nous allons donner les concepts fondamentaux de data mining et du processus ECD. Ensuite nous parlerons de la confidentialité ainsi que la nécessité de la préserver la vie privée.

II. Le processus ECD:

Le data mining est une étape d'une chaine de traitement plus complète, le processus d'Extraction de Connaissances à partir de Données (ECD). Qu'est ce que l'ECD ? Quels sont ses étapes ?

1. Définition :

L'*extraction de Connaissances à partir de Données (ECD)* désigne tout le cycle de découverte d'informations ou de connaissances dans les bases de données. Il regroupe donc toutes les opérations à effectuer pour extraire de l'information de ces données [22].

2. Etapes du processus ECD :

L'ECD est un processus complexe qui se déroule suivant une suite d'opérations. Selon [6], l'ECD peut être vu comme un processus en sept étapes :

1. Positionnement du problème ;

2. Collecte et sélection des données ;

3. Nettoyage des données ;

4. Actions sur les attributs ;

5. Construction du modèle ;

6. Evaluation des résultats ;

7. Intégration de la connaissance.

2.1. Positionnement du problème :

Il s'agit de comprendre le contexte de la recherche pour donner une signification logique aux variables de sa problématique. Dans cette étape, toutes les variables susceptibles d'expliquer le phénomène étudié sont identifiées.

Une première approche consiste à reformuler le problème pour qu'il puisse être traité par les techniques et les outils de modélisation (par exemple modéliser le problème sous forme d'une fonction mathématique).

2.2. Collecte et sélection des données :

On procède d'abord par une collecte de données. Ces dernières se trouvant généralement dans des bases de données ou dans des datawarehouses. Les données collectées, ne faisant état d'aucun critère de sélection préalable, nécessitent habituellement une phase de sélection pour n'en garder que les données dont on a besoin.

Une sélection optimale des données nécessite souvent l'aide d'experts du domaine pour déterminer les facteurs, ou variables, les plus aptes à décrire la problématique.

Dans le cas où les experts ne sont pas disponibles, une recherche des variables les plus déterminantes est effectuée par des techniques d'analyse (comme la régression, les réseaux de neurones…).

2.3. Nettoyage des données :

Une faible qualité de données (erreurs de saisie, champs nuls, valeurs aberrantes…) impose généralement une phase de nettoyage de celles-ci. Cette étape a pour objectif de corriger ou de contourner les inexactitudes ou les erreurs de données.

On commence par rechercher et éliminer les valeurs aberrantes. Ensuite, on vise à gérer les données manquantes par l'une des méthodes suivantes :

- Exclure les enregistrements incomplets ;

- Remplacer les valeurs manquantes (généralement par la moyenne) ;

- Gérer les valeurs manquantes (lorsque l'absence de donnée est acceptable du point de vue de la performance).

2.4. Actions sur les attributs :

Une fois que les variables sont pertinentes et que les données sont fiables, une transformation éventuelle s'impose pour les préparer au travail d'analyse. Il s'agit d'intervenir sur les valeurs des variables pour qu'elles soient mieux exploitables par les algorithmes de traitement. Voici des exemples d'actions sur les variables parmi les plus usuelles :

Modification de l'unité de mesure :

Afin d'éviter certaines disproportions, il est recommandé de procéder à une ''normalisation'' des distributions. Dont l'objectif est d'obtenir des ordres de grandeurs comparables pour chaque attribut. Pour un échantillon de données numériques, la normalisation consiste à soustraire, à chaque valeur, la valeur moyenne de l'échantillon et à diviser cette différence par l'écart-type de l'échantillon.

Transformation des dates :

Les dates absolues ont généralement beaucoup moins de ''valeur'', du point de vue d'un travail de modélisation, que des fréquences ou des écarts entre dates. Ainsi, on calculera par exemple l'ancienneté d'un étudiant par la différence entre la date d'entrée et la date actuelle.

Modification des données géographiques en coordonnées :

Les techniques de data mining ont généralement des difficultés à appréhender des codes postaux ou des adresses. Une approche habile consiste à joindre des coordonnées de longitude et de latitude (géocodage).

2.5. Construction du modèle (data mining) :

Il s'agit de la phase la plus souvent décrite sous le terme de ''data mining'' et qui repose, pour partie, sur une recherche exploratoire, dépourvue de préjugés concernant les relations entre les données.

Les techniques de fouille de données cherchent à construire un ''modèle''. Ce modèle doit rendre compte des relations liant la description d'une situation à un résultat. La formulation d'un modèle peut prendre plusieurs formes [26] :

- Une formulation à base d'équations (*réseaux de neurones*, *techniques de régression*);

- Une formulation à base d'analyse logique de la forme ''Si A et B alors C'' (*arbres de décision*, *règles d'associations*) ;

- Une formulation par les techniques de projection des données dans un espace plus ou moins structuré du type ''x, y ı A'' (*analyse factorielle*).

La construction du modèle se fait sur les données d'apprentissage qui doivent être distinctes des données de test. Les données d'apprentissage et de test sont généralement issues à partir de la même base de données mais elles ne comprennent pas les mêmes données.

L'apprentissage prend généralement 70% à 80% des enregistrements, la base de test étant constituée du reste [26].

Les modèles construits de manière totalement automatique, sont particulièrement sensibles à la qualité des données. Aussi, une interaction avec l'utilisateur est parfois requise pour guider et améliorer le raisonnement au fur et à mesure de la construction du modèle.

Cette interactivité contribue, également, à bâtir des modèles parfois moins performants mais souvent plus réalistes.

2.6. Evaluation du résultat :

Cette étape permet d'estimer la fiabilité du modèle c'est-à-dire sa qualité à déterminer correctement les valeurs, qu'il est censé avoir apprises, sur des cas nouveaux.

A l'issue de la construction du modèle, il est possible d'en tester la pertinence sur l'ensemble des tests. En général, la performance d'un modèle s'apprécie au travers une matrice qui compare la situation réelle et la situation prévue par le modèle. La qualité globale du modèle est la portion de prédictions exactes par rapport aux nombre total de prédictions. Un niveau de prédiction de 80% est jugé moyen [26].

2.7. Intégration de la connaissance :

La connaissance n'est rien tant qu'elle n'est pas convertie en décision puis en action. Cette étape est essentielle puisque il s'agit de la transition du domaine des études au domaine opérationnel.

Dans cette phase finale, il est opportun de dresser un bilan du déroulement des étapes précédentes. Ce bilan sert à améliorer l'existant en termes de données et de collecte de ces données ; par exemple, la faible qualité constatée de données conduit à revoir les processus d'alimentation de la base.

Les étapes du processus ECD peuvent être intégrées dans trois grandes étapes :

- L'étape de prétraitement : Elle regroupe la collecte, la sélection, le nettoyage des données et les actions sur les variables ;

- L'étape de data mining proprement dit ;

- L'étape de post-traitement : Elle regroupe l'évaluation des résultats et l'intégration de la connaissance. (Figure 1)

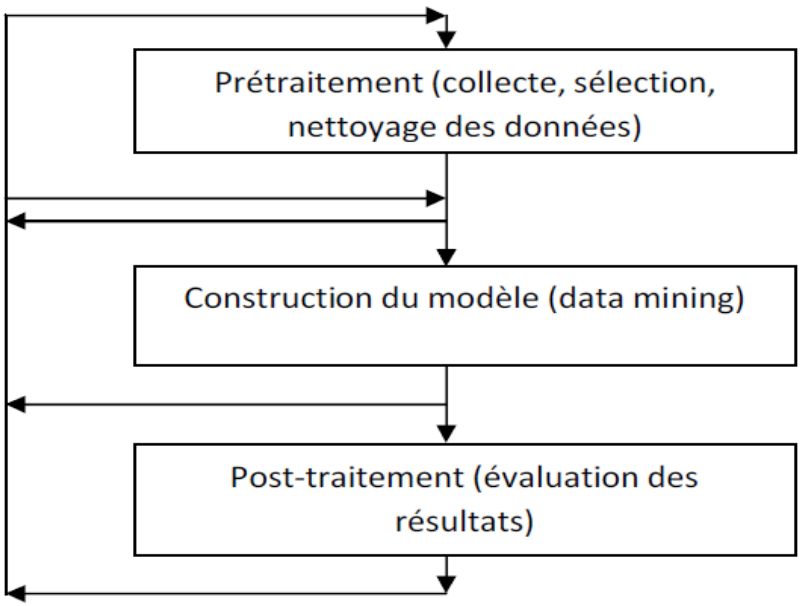


Figure 1 : Les étapes du processus ECD

III. Le data mining :

Après avoir énuméré les différentes étapes du processus ECD, nous nous intéressons dans ce qui suit particulièrement à l'étape de data mining qui constitue le cœur du processus ECD. Qu'est ce que le data mining ? A quoi sert-il ?

1. Définition :

Le data mining, ou fouille de données, est l'une des étapes du processus ECD (*voir 2*).

Cette étape regroupe l'ensemble des méthodes et techniques destinées à l'exploration des bases de données de façon automatique, ou semi-automatique, dans l'optique de détecter des règles, des associations, des tendances nouvelles et imprévisibles, des structures particulières restituant l'essentiel de l'information utile [28].

Selon [29], il s'agit d'un processus de sélection, exploration, modification et modélisation de grandes bases de données afin de découvrir des relations entre les données jusqu'alors inconnues.

2. Principe :

Le data mining adopte une démarche sans a priori (approche pragmatique) et essaie ainsi de faire ''émerger, à partir de données brutes, des inférences que l'expérimentateur peut ne pas soupçonner, et dont il aura à valider la pertinence à l'aide des techniques et des algorithmes'' [20].

3. Tâches de data mining :

Le data mining est un ensemble de techniques complémentaires dédiées à différentes tâches. Selon [21], ces techniques sont partagées, principalement, entre la classification automatique (supervisée et non supervisée) et la recherche d'associations.

3.1. La classification automatique supervisée :

La classification automatique supervisée consiste à examiner les caractéristiques d'un objet nouvellement présenté afin de l'affecter à une classe d'un ensemble prédéfini [21]. Le modèle généré permet, en outre, de prédire ou estimer la valeur manquante ou erronée en utilisant le modèle de classification comme référence [12].

3.2. La classification automatique non supervisée :

C'est une tâche qui vise à identifier des ensembles d'éléments qui partagent certaines similarités. Elle se distingue de la classification automatique supervisée par le fait qu'elle ne se base pas sur des classes prédéfinies [21].

Le principe général de la classification automatique supervisée, également appelée *clustering*, repose sur le regroupement des objets de telle manière que ceux qui appartiennent à la même classe soient fortement similaires entre eux et fortement dissimilaires avec les objets qui appartiennent aux autres classes. Cette tâche précède, généralement, la phase de *clustering*.

3.3. Les règles d'association :

C'est une tâche qui permet de découvrir les rapports de lien qui peuvent exister dans une base de données. Ces liens sont généralement exprimés sous la forme '' A -> B '' qui signifie que la présence de A implique la présence de B (avec une certaine probabilité).

Exemple : Un étudiant qui réussi en mathématiques réussira en algorithmique dans 80% des cas.

4. Techniques de data mining :

Les techniques de data mining diffèrent en fonction des besoins de l'utilisateur, entre autres la tâche à effectuer. Chacune des tâches citées ci-dessus regroupe une multitude d'algorithmes pour construire le modèle auquel elle est associée.

Selon [23], les dix algorithmes les plus populaires dans le domaine de data mining sont, dans l'ordre: *C4.5*, *k-Means*, *SVM*, *Apriori*, *EM*, *PageRank*, *AdaBoost*, *k-NN*, *Naive Bayes* et *CART*. Ces algorithmes sont classifiés, selon la tâche associée, comme suit :

- Algorithmes de construction des arbres de décision : *CART*, *C4.5* ;

- Algorithmes de classification non supervisée : *k-Means*, *EM* ;

- Algorithmes de classification supervisée : *k-NN*, *SVM*, *Naive Bayes* et *AdaBoost* ;

- Algorithme d'extraction des règles d'association : *Apriori* ;

- Algorithme de classification automatique des pages Web : *PageRank*.

On présente ici une brève récapitulation des algorithmes répandus de fouille de données et une étude des applications du Data Mining Distribué au domaine de la confidentialité ; on fournit un certain détail sur les modèles de partitionnement des données.

Le Data Mining (Fouille de données) est considérée, dans sa globalité, comme l'analyse d'ensembles de données à la recherche de relations possibles (méconnues auparavant) entre ces données mais aussi de récapituler la connaissance véhiculée par les données de manière originale et compréhensible par propriétaire [61]. Il y a diverses fonctionnalités concernant la fouille de données.

[62] les aborde en citant en premier lieu la "caractérisation de données" qui est, selon lui, la récapitulation des caractéristiques générales d'une classe cible contenant des données.

La "discrimination de données", est quand à elle, une comparaison entre les caractéristiques générales des éléments de données d'une classe cible.

"L'analyse d'association" est la découverte des règles d'association montrant les conditions de type attribut-valeur qui possèdent la caractéristique d'occurer ensemble dans un ensemble de données.

La "Classification", selon le même auteur est le processus consistant à trouver un ensemble de modèles (ou de fonctions) qui décrivent et distinguent entre classes de données ou encore concepts, dans le but de pouvoir employer le modèle pour prévoir la classe des objets dont la classe est inconnue. Ce modèle dérivé peut être représenté dans diverses formes, telles que les règles de classification, les arbres de décision, les formules mathématiques, réseaux de neurones.

A la différence de la classification et la prévision, qui analysent les éléments des objets de données marqués par classe, le "clustering" lui, analyse des éléments de données sans avoir à consulter une étiquette connue auparavant de la classe.[62] donne un examen détaillé de ces diverses fonctionnalités. Enfin, [63] ;[61] ont abordé les algorithmes liés au concept d'extraction de connaissances.

4.1. Fouille Distribuée (Distributed Data Mining)

Contrairement au modèle centralisé, le modèle distribué de fouille de données(DDM) part avec l'idée que les sources de données sont distribuées à travers de multiples sites. Les algorithmes développés dans ce contexte abordent le problème en cherchant à extraire de façon efficace des résultats à partir de toutes les données réparties à travers ces sources distribuées. Sachant que le souci majeur est l'efficacité (et rien d'autre), la plupart des algorithmes développés jusqu'ici ne prennent pas le critère de sécurité en considération.

Une approche simple pour la fouille de données à travers des sources multiples et qui, en même temps, ne partage pas des données peut être envisagée en exécutant les outils de fouille de données existants au niveau de chaque site de manière indépendante et en combinant les résultats [64] ;[65] ;[66]. Cependant, ceci va échouer à l'idée de donner des résultats globalement valides. La cause de la disparité entre les résultats locaux et globaux inclut :

- Des valeurs concernant la même entité peuvent être réparties à travers des sources. La fouille de données locale ne pourra pas détecter des corrélations entre sites.

- Le même article peut être reproduit au niveau de plusieurs sites et serait par la conséquence, surévalué à la fin (la phase de résultats).

- Les données d'un site sont susceptibles de venir d'une population homogène. Des distinctions géographiques ou démographiques importantes au sein de cette population ne peuvent pas être vues en ayant recours à un seul site.

[67] propose une méthode pour les données horizontalement divisées en y ajoutant une classification distribuée. Des approches comme ceux de [64] ;[65] ;[66] de méta-apprentissage ont été développées dans ce contexte.

Elles partent classificateur par apprentissage local au niveau de chaque site en vu de générer un classificateur global. Ceci pourrait protéger les différentes entités, mais il reste à démontrer que les classificateurs individuels (locaux) ne révèlent pas l'information privée.

D'autres algorithmes ont été développés qui divisent les données en sous-ensembles [67]. Selon [68] ;[69], les algorithmes parallèles de fouille de données peuvent également servir de point de départ.

Avec des données distribuées, la manière avec laquelle les données sont distribuées joue également un rôle important dans la définition du problème, du au fait que les données peuvent être divisées en différentes parties, soit verticalement ou horizontalement.

4.2. Approches Proposées

Dans nos jours, la collecte de données est omniprésente. La collecte de connaissance à partir de ces données est une tâche. Si les données sont extraites d'un seul site, le data mining ne pose pas vraiment un risque de confidentialité en plus ; n'importe quelle entité possédant un accès aux données à cet emplacement aura l'information spécifique qu'il désire. Le contrôle d'une telle utilisation n'est pas du domaine du PPDM.

4.2.1. Modèle de Partitionnement des données:

Avant de formuler des solutions, il est nécessaire de d'abord modéliser les différentes manières dont des données sont distribuées dans le monde réel. A savoir les deux méthodes de base : Horizontale (distribution homogène) et verticale (distribution hétérogène).Comme exemple, prenons le cas de la figure 3 qui représente une B.D.D d'un hôpital et celle d'une compagnie d'assurances ; les deux collectent des informations relatives aux malades ; Alors

que la Figure 2 montre le cas de deux banques qui collectent des informations de la carte de crédit du client.

L'ensemble de données *D´(E; I)*, où *E* est l'ensemble d'entités pour lequel l'information est collectée. I est l'ensemble caractéristique qui est collecté. Nous supposons qu'il existe *k* différents sites : *P1; ....; Pk*

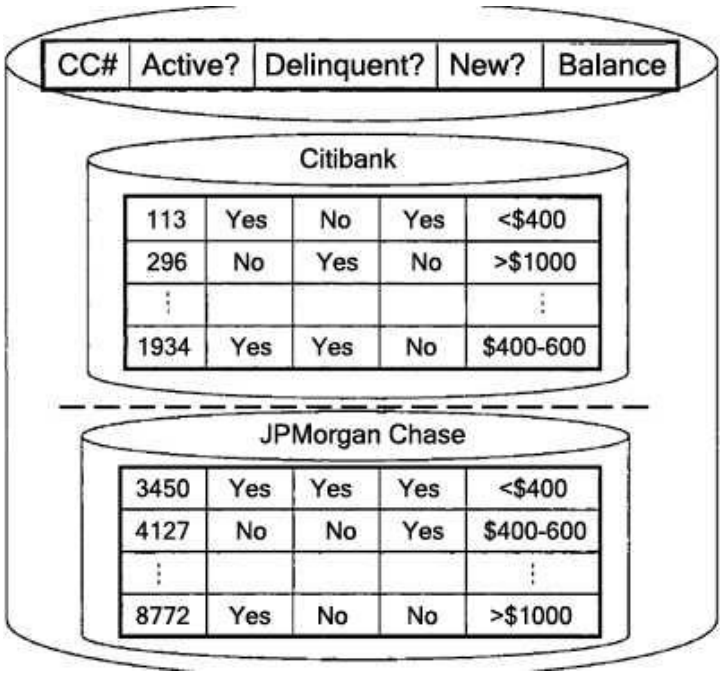


Figure 2: Partitionnement Horizontal : Distribution Homogène

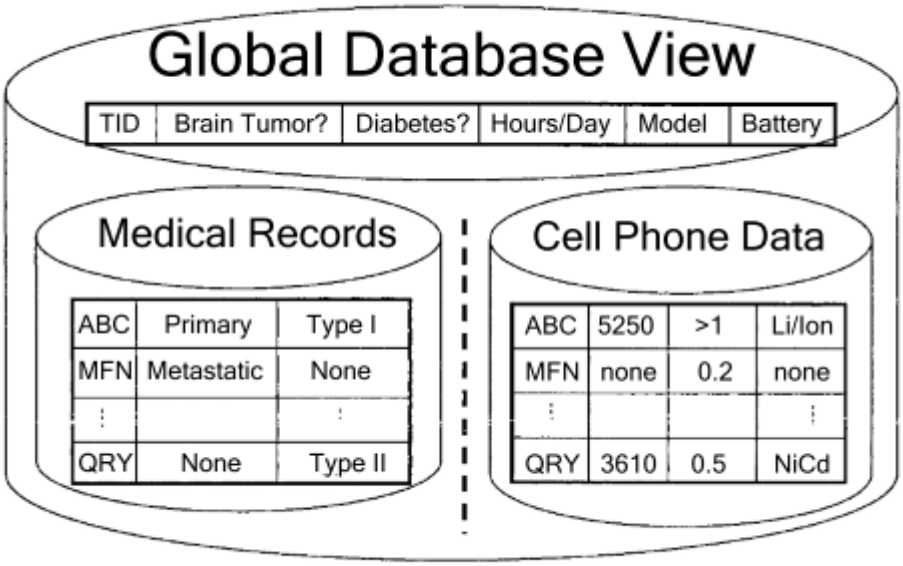


Figure 3: Partitionnement Vertical : Distribution Hétérogène

4.2.2. Perturbation des données:

Cette approche de PPDM est basée sur la perturbation des données à l'origine, fournissant par la suite l'ensemble perturbé de données comme entrée à l'algorithme de fouille de données. Les propriétés de préservation de la confidentialité sont un résultat de la perturbation : Des valeurs (privées) identifiables ne sont pas indiquées.

Celles qui ne le sont pas seront divulguées. Par exemple si une compagnie souhaite fouiller des données partant d'un aperçu de valeurs d'informations confidentielles. Alors qu'il se peut que certains n'aimerait toutefois pas que ces données soient en diffusion mais qu'elles soient perturbées. Si un attribut est continu, une simple méthode de perturbation consiste à ajouter le bruit, produit à partir d'une distribution de probabilité spécifique. Soit X un attribut et pour un individu $X = x$, où x est une valeur réelle. r étant un nombre aléatoirement tiré d'une distribution normale de moyenne 0 et de variance 1. Au lieu de révéler x, l'individu indique $x+r$. Des méthodes plus compliquées peuvent être conçues.[70] a proposé la méthode aléatoire de réponse pour manipuler des questions sensibles de confidentialité dans l'aperçu(survey). Supposons qu'un attribut Y avec deux valeurs (oui ou non) est intéressant lors de l'aperçu. L'attribut cependant est privé et un individu qui participe à l'aperçu n'est pas disposé à le révéler. Donc, en posant les deux questions suivantes :

1. J'ai l'attribut Y.

2. Je n'ai pas l'attribut Y.

L'individu va donner une réponse de oui ou non mais on ne sait pas à quelle question il a répondu. [71] a employé cette technique pour établir la préservation de la confidentialité dans les arbres de décision.

[72] a exploré des arrangements plus sophistiqués pour ajouter le bruit. Les données randomisées ou bruyantes préservent la confidentialité individuelle, mais elles posent problème à la fouille de données. Deux questions cruciales sont à se poser :

Comment extraire de l'information à partir de données randomisés et à quel point les résultats basés sur des données randomisées sont ils comparables aux résultats probables des données d'origine?

Le degré de randomisation dépend de la quantité d'information qu'une source de données veut préserver, ou combien d'information permet-elle aux autres d'apprendre.

[68] précisé une chose importante : la randomisation arbitraire n'est pas sûre. Bien que randomisées les données peuvent paraître très différentes des données originales ; un adversaire peut tirer profit des propriétés telles que les corrélations et les modèles contenus dans les données d'origines dans le but de se rapprocher le plus de leurs valeurs. Par exemple, supposer que des données contiennent un attribut dont les valeurs sont une constante. Basé sur les données randomisées, un analyste peut apprendre ce fait assez facilement, qui a immédiatement comme conséquence une brèche dans la confidentialité.

5. Domaines d'application :

Le data mining est une spécialité transverse, elle regroupe un ensemble de théories et d'algorithmes ouverts à tout domaine susceptible de drainer une masse importante de données. Parmi ces domaines, on cite :

- *Sociétés commerciales :* L'étude d'appétence dans les sociétés commerciales vise à concentrer les mailings et le phoning sur les clients susceptibles de répondre favorablement;

- *Grandes surfaces :* L'analyse du ticket de la supérette permet de déterminer les articles souvent achetés simultanément et organiser les promotions en conséquence ;

- *Secteur bancaire :* L'objectif premier des banques est de réduire le risque de prêts bancaires ; la création de modèles à partir des caractéristiques des clients permet de discriminer les clients à risque élevé ;

- *Secteur d'assurances :* Dans le secteur d'assurances, le data mining est utilisé pour la détection de fraudes et leur anticipation ;

- *Secteur médical :* Dans le secteur médical, le data mining est naturellement répandu. On trouve par exemple la détermination de groupes de patients susceptibles d'être soumis à des protocoles thérapeutiques déterminés. Les études sur les associations de médicament permettent, entre autres, de révéler les anomalies de prescription.

- *Secteur de l'éducation :* Le data mining est employé dans les établissements scolaires pour améliorer la qualité d'enseignement. Par exemple, répartir les élèves ayant une grande capacité d'assimilation dans la même classe.

6. Difficultés de data mining :

La mise en œuvre de data mining rencontre trois difficultés principales [24] et [25] :

Qualité des données : Les statistiques ont montré que 60% à 70% du temps de travail dans un projet de data mining est consacré au prétraitement des données (sélection, correction, transcodage, chargement…), ce qui montre que le temps de préparation est un inconvénient majeur qui influe sur le temps global du projet.

Choix des algorithmes et de l'itinéraire du travail : Pour pouvoir répondre aux questions qui se posent, les algorithmes doivent être choisis en fonction du problème traité. Il faut que l'expert en datamining soit aussi un animateur et possède des qualités que l'on trouve rarement ensemble chez la même personne : rigueur dans la méthode, ouverture et chaleur humaine dans la communication.

Evaluation des résultats : Avant de procéder au déploiement final du modèle, il est important de l'évaluer plus complètement et de passer en revue toutes les différentes étapes exécutées pour construire ce modèle. Ceci permettra d'être certain qu'il permet d'atteindre les objectifs fixés. Lors de la définition du problème, un objectif principal est de déterminer s'il y a un aspect important du problème à résoudre qui n'a pas été suffisamment considéré. A la fin de cette phase, une décision sur l'utilisation des résultats fournis par les outils de data mining devrait être prise. Ce qui suit est un état de l'art des principaux algorithmes de data mining.

IV. Un état de l'art des principaux algorithmes de data mining :

1. Introduction :

Le data mining offre une très grande variété de techniques et d'algorithmes de fouille de données. Ces algorithmes ont des origines diverses ; Certains sont issus des statistiques (régression…), d'autres de l'intelligence artificielle (réseaux de neurones, arbres de décision…), certains encore s'inspirent de la théorie de l'évolution (algorithmes génétiques…) ou de l'éthologie (colonies d'abeilles…). Cette combinaison de technologies facilite la résolution, la compréhension, la modélisation et l'anticipation des problèmes.

Dans ce deuxième chapitre, on se propose de donner les bases mathématiques, les algorithmes et les programmes de calcul pour les principales tâches de data mining : la classification non supervisée, la classification supervisée et l'extraction des règles d'association.

Chaque tâche comporte un recueil d'algorithmes parmi les plus courants. Pour chaque algorithme, on donne le principe de fonctionnement, les caractéristiques, les avantages, les inconvénients et éventuellement un exemple d'application.

Les données sont considérées comme étant l'ensemble des enregistrements de la base de données. Chaque enregistrement représente un objet et les champs de l'enregistrement définissent les attributs de l'objet.

2. La classification automatique non supervisée :

2.1. Introduction :

La classification est une tâche appliquée dans la vie courante. Elle est utilisée pour expliquer les nouveaux phénomènes rencontrés en les comparants avec des concepts et des phénomènes connus et en essayant de rapprocher les caractéristiques le plus possible.

C'est un sujet de recherche actif qui se place au cœur de data mining, ce qui justifie pleinement l'intérêt qui lui est porté.

Les techniques de classification sont réparties essentiellement en deux approches : la classification supervisée et la classification non supervisée (appelée aussi ''clustering'').

L'objectif le plus simple de clustering est de répartir une population d'objets en groupes d'observations homogènes (classes), chaque classe étant bien différenciée des autres, pour former une partition.

Le clustering est un sujet abordé par plusieurs métiers. Ce fait lui donne l'avantage d'être utilisé par divers domaines d'application qui peuvent être catalogués suivant le but recherché selon trois grands types [16] :

- *Extractions des connaissances :* Dans ce type d'applications, on cherche généralement à donner une description sémantique aux groupes constitués afin de donner un sens à l'information dont on dispose ;

- *Réduction des bases de données :* L'objectif visé est de réduire l'espace des données sur lequel on travaille en espérant que le problème qu'on cherche à résoudre devient beaucoup plus simple ;

- *Etude de comportement (profiling) :* Le but est de détecter des sous populations qui partagent des caractéristiques proches, ce qui leurs donne un certain comportement commun. Les systèmes de décision se basent souvent sur l'étude de comportement afin d'entreprendre des actions convenables à chacune des sous population.

Dans ce qui suit, nous allons présenter les principaux algorithmes de clustering parmi les plus représentatifs tout en essayant de bien cerner les points forts et les faiblesses de chaque algorithme.

2.2. Définition :

Le clustering est une approche de classification dont les classes existent a posteriori. Au départ, on dispose d'un ensemble d'objets non étiquetés (dont la classe est inconnue). A partir de ces objets, l'idée est de parvenir à détecter des objets ''similaires'' afin de les regrouper dans des classes [16].

Selon [17], le clustering consiste à diviser une population d'objets en sous-ensembles d'objets appelés classes pour que tous les objets dans une même classe soient similaires et les objets de classes distinctes soient dissimilaires.

Parmi les objectifs que le clustering doit permettre d'atteindre, citons la réduction des données et la prédiction basée sur les groupes.

2.3. Terminologie :

Avant d'entamer le processus de clustering, il est indispensable de définir quelques concepts:

2.3.1. Représentation d'une classe :

C'est la description de l'ensemble des objets constituant la classe. Cette description, caractérisée par un ensemble de paramètres, permet de définir la forme et la taille de la classe dans un espace de données.

Dans la plupart des algorithmes de classification, la description d'une classe est simplement une combinaison linéaire des objets (centroide), ou par un axe médian (médoide) [18].

Le centroide :

Un centroide d'une classe *C* est un vecteur où chacune de ses composantes correspond à la moyenne arithmétique (quand elle existe) des objets de la classe *C* [18].

Le médoide :

On prend k objets parmi les objets de la classe. Ces k éléments sont centraux vis-à-vis de la classe, c'est-à-dire qu'ils sont proches du centre géométrique de la classe [18]. Ils

permettent de représenter la classe non pas par un objet unique, qui n'a pas toujours d'existence réelle, mais plutôt par un noyau médian censé définir au mieux la classe.

2.3.2. La notion de proximité (distance) :

Une mesure de distance (appelée aussi mesure de similarité ou de proximité) exprime une similarité ou une dis-similarité : plus deux objets se rassemblent, plus leur similarité est grande et plus deux objets s'éloignent, plus leur similarité est petite.

La mesure de distance peut s'appliquer entre deux objets ou entre deux classes.

Distance entre deux objets :

Cette mesure peut être exprimée par une formule qui génère un nombre positif reflétant la proximité ou l'éloignement entre deux objets.

Selon [11], pour qu'une fonction de deux objets $d(x, y)$ soit une vraie métrique de distance, elle doit remplir les conditions suivantes :

- $d(x, y) = 0$ si et seulement si $x = y$

- $d(x, y) \geq 0$ quels que soient x et y

- $d(x, y) = d(y, x)$. (Symétrie)

- $d(x, y) \leq d(x, z) + d(z, y)$. (Inégalité triangulaire)

Les mesures de distance entre objets dépendent notamment du type de données. Parmi les mesures qui traitent des données numériques très populaires, on trouve la distance de Minkowski [19], notée *dm,* entre deux objets *x, y* définis chacun par ses attributs ;

$x = (a1, a2, .., an)$, $y = (b1, b2, .., bn)$:

Distance entre deux classes :

Les mesures inter-classe les plus populaires dans la littérature sont : le lien simple (*single link)*, le lien moyen (*average link)* et le lien complet (*complete link)* [20].

La distance D entre 2 classes $C1$, $C2$ notée D ($C1$, $C2$) est calculée pour toute paire d'objets, un objet de chaque classe. Une opération (Minimum pour le lien simple, Moyenne pour le lien moyen et Maximum pour le lien complet) est ensuite appliquée pour chaque paire d'objets.

D ($C1$, $C2$) = Opération { $d(x, y) \mid x \in C1, y \in C2$ } [20].

Tel que : $d(x,y)$ = distance entre les deux objet x et y.

2.3.3. L'inertie :

L'inertie d'une population d'objets est la moyenne des carrés des distances des individus à leur centre de gravité. Cette entité mesure l'homogénéité des objets de la population.

L'inertie intra-classe est la somme des inerties des classes. Plus cette entité est petite, plus les objets de la même classe sont homogènes.

L'inertie inter-classe est la moyenne (pondérée par le nombre de classes) des carrées des distances des centres des classes au centre global. Plus cette entité est importante, plus les classes sont éloignées [21].

Un bon clustering possède deux propriétés :

- Inertie intra-classe (variance des objets dans la même classe) minimale ;

- Inertie inter-classe (variance des centres des classes) maximale.

2.4. Les algorithmes de clustering :

La majorité des algorithmes de clustering peuvent être répartis en trois grandes familles [12], [22] et [23] :

- *Les algorithmes de partitionnement :* Ils adoptent une recherche itérative des classes jusqu'à l'optimisation d'un critère d'arrêt (par exemple atteindre une certaine inertie).

- *Les algorithmes hiérarchiques :* Ils sont descendants si, à partir d'une seule classe, ils cherchent à établir une partition par division ; ou ascendants s'ils cherchent à former des classes plus grandes par fusion de classes jusqu'à la satisfaction d'un critère d'arrêt.

- *Les algorithmes à base de densité :* Les classes sont formées selon le voisinage des objets et le niveau de densité de chaque objet.

2.4.1. Les algorithmes de partitionnement :

Leur principe général est de démarrer à partir d'un certain nombre de classes qui sont partitionnés d'une manière aléatoire en effectuant une ''redistribution'' des objets ou en essayant d'identifier les classes comme étant des régions très peuplées jusqu'à la rencontre d'un critère d'arrêt.

Les algorithmes de partitionnement que nous allons développer sont k-Means, k-Medoïds, PAM, CLARA et CLARANS.

2.4.1.1. L'algorithme k-Means :

L'algorithme k-Means (ou l'algorithme des centres mobiles) a été introduit par J.MacQuenn et mis en œuvre sous sa forme actuelle par E.Forgy [12]. Il est le plus utilisé dans les applications scientifiques et industrielles car il est le plus simple et le plus rapide.

Dans cet algorithme, chaque classe est représentée par la moyenne (centroide). k-Means est un algorithme itératif. Il commence avec un ensemble de *k* individus de référence choisis de façon aléatoire. Les individus de données sont ainsi partitionnés dans *k* classes ; un individu appartient à une classe si le centre de cette classe est le plus proche de lui (en

terme de distance). La mise à jour des centroides et l'affectation des individus de données aux classes sont réalisées pendant les itérations successives [24].

La *Figure* ***4*** montre un exemple de déroulement de l'algorithme k-Means sur un nuage d'objets bidimensionnels, avec $k = 3$.

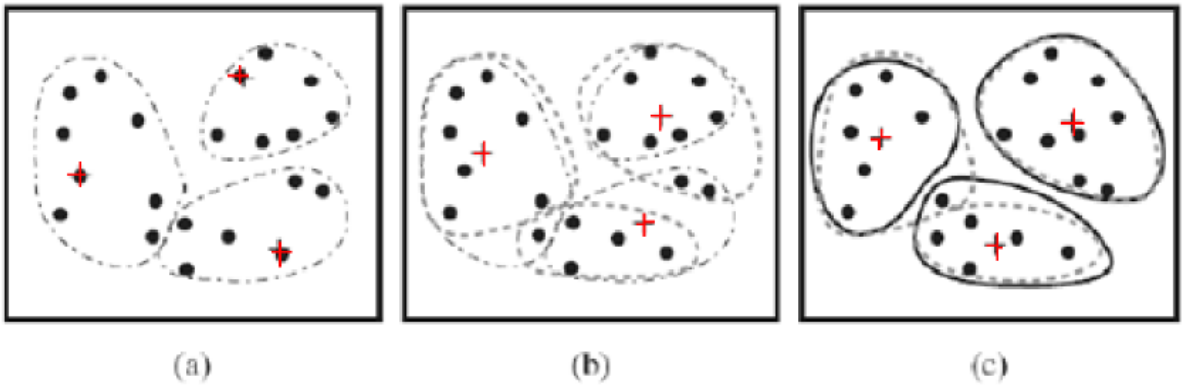


Figure 4.a **:** Choix aléatoire de trois centre initiaux (objets) marqués par (+) et affectation de chaque objet restant au centre le plus proche ;

Figure4.*b* : Calcul du nouveau centre pour chaque classe (moyenne des objets de la classe) et redistribution des objets selon les nouveaux centres ;

Figure 4.*c* : Le processus est répété jusqu'à stabilité des classes.

k-Means vise donc à minimiser la variance intra-classe. Comme avantages de cet algorithme, on cite [24] :

- Il s'adapte bien pour des populations de tailles importantes ;
- Il est relativement efficace ; si n le nombre d'objets, k le nombre de classes et t le nombre d'itération, l'algorithme converge, généralement, avec k et t suffisamment inférieur à n ($k, t << n$) ;
- Il est indépendant de l'ordre d'arrivée des données.

Parmi les inconvénients de cet algorithme, on cite [24] :

- Il est applicable seulement dans le cas où la moyenne des objets est définie ;
- Le nombre de classes k, doit être spécifié à priori ;
- Il converge souvent vers un optimum local ;
- Il est sensible aux objets isolés (bruits) ;

- Il n'est pas adapté pour découvrir des classes avec des structures non-convexes, et des classes de tailles et formes différentes.

2.4.1.2. L'algorithme k-Medoids :

Il est introduit par Kaufman et Rousseew [12]. L'esquisse de l'algorithme k-Medoids ressemble à celle de k-Means sauf que, contrairement à l'algorithme k-Means où la classe est représentée par une valeur moyenne, le centroide, dans l'algorithme k-Medoids, une classe est représentée par un de ses objets prédominants, le médoide.

L'algorithme k-Medoids utilise une fonction objective qui définit la distance moyenne entre un objet et le médoide.

Soient :

k : le nombre de classes défini à priori,

P= {p1, p2, … , pn}, population de *n* objets,

M = {m1, m2, …, mk}, l'ensemble des médoides des *k* classes,

C = {C1, C2, … , Ck}, l'ensemble des *k* classes ;

d(pij, mi) : distance d'un objet pij à son médoide mi, avec pij, mi ı Ci pour j=1,|Ci|

|Ci| étant le nombre d'objets de la classe Ci.

La *Figure 5* est une illustration du déroulement de l'algorithme k-Medoids sur un nuage d'objets bidimensionnels avec *k* = 3.

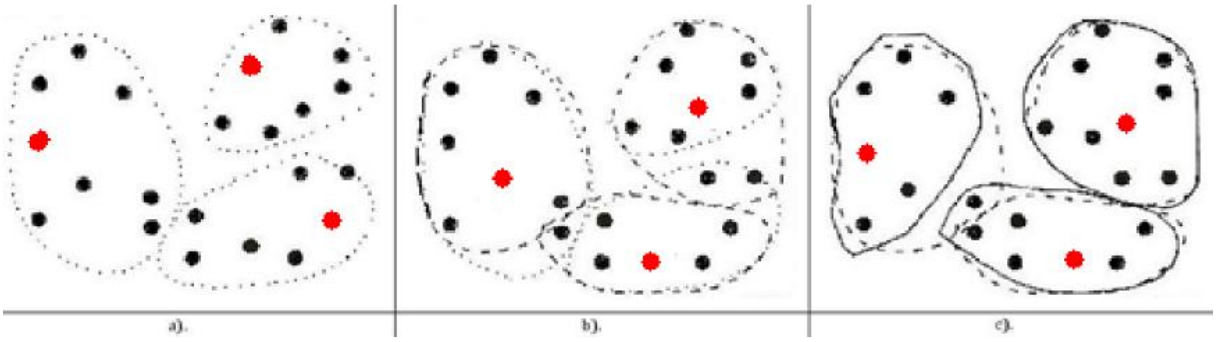


Figure 5.a **:** Choix des trois centres initiaux marqués par (●) et affectation de chaque objet restant au centre le plus proche ;

Figure 5.b **:** Calcul des nouveaux médoides pour chaque classe et redistribution des objets selon les nouveaux médoides (Le médoide étant l'objet le plus proche du centroide de la classe) ;

Figure 5.c **:** Le processus est répété jusqu'à stabilité des classes.

Comme avantages de cet algorithme par rapport à k-Means, on cite :

- Il s'adapte à n'importe quel type de données ;
- Il est insensible aux objets isolés.

Pour améliorer la qualité de clustering de L'algorithme k-Medoids, de nombreuses variantes s'y sont succédées, entre autres les algorithmes PAM [25], CLARA [12] et CLARANS [17].

Ces algorithmes adoptent le même principe de représentation par médoide et utilisent la même fonction objective donnée ci-dessus.

2.4.1.3. L'algorithme PAM :

PAM (*Partitioning around Medoids*) a été développée par Kaufman et Rousseeuw [16]. L'idée de cet algorithme consiste à commencer avec un ensemble de *k* médoides puis échanger le rôle entre un objet médoide et un non-médoide si cela permet de réduire la distance globale, ce qui revient à minimiser la fonction objectif (*voir 2.4.1.2*) [25].

L'inconvénient majeur de cet algorithme est son coût total de calcul, il est d'une complexité quadratique de l'ordre de O(*k*.(*n*-*k*)2) avec *n* le nombre d'objets et *k* le nombre de classes pour chaque itération, ceci le rend non adaptable pour une population importante d'objets.

2.4.1.4. L'algorithme CLARA :

L'algorithme CLARA (*Clustering LARge Application*) a été mise en œuvre par Kaufman et Rousseeuw dans le but de réduire le coût de calcul de PAM [12] et [17].

Cet algorithme travaille sur des échantillons. On prend une petite partie d'objets (échantillon). Ensuite, les *k* médoides sont déterminés en appliquant PAM sur cet échantillon. Si cet échantillon est choisi d'une manière aléatoire, alors il représente bien tous les objets, donc les médoides sont similaires à ceux qui sont créés à partir de tous les objets [17].

L'algorithme est, généralement, exécuté sur plusieurs échantillons pour obtenir le meilleur résultat. Les auteurs ont indiqué, suite à leurs expérimentations, que la taille d'un échantillon de (40+2*k*), *k* étant le nombre de classes, donne un bon résultat [16].

Le principal inconvénient de l'algorithme CLARA, outre les difficultés d'échantillonnage, est que si un objet qui serait le meilleur médoide n'apparait dans aucun échantillon, alors la meilleure solution ne pourra jamais être atteinte.

2.4.1.5. L'algorithme CLARANS :

CLARANS (*Clustering Large Application based on RANdomized Search*) a été mis en place par Raymon Ng et Jiawei Han [12] et [17].

Les auteurs de cet algorithme ont utilisé un graphe pour résoudre le problème de recherche des *k* médoides. Chaque nœud dans le graphe est un ensemble de *k* objets médoides et possède *k*.(*n*-*k*) voisins. Un nœud voisin s'obtient en échangeant les rôles entre le médoide et un non-médoide.

Une fois le graphe construit, PAM parcourt tous les nœuds voisins à la recherche de la solution, ce qui rend le cout de traitement trop cher. CLARA, à son tour, ne parcourt qu'un sous-graphe de l'ensemble des voisins, il travaille donc sur une région localisée.

CLARANS est une combinaison des deux concepts, il choisit aléatoirement un nombre de nœuds voisins (ce nombre est donné en paramètre) et se déplace d'un nœud au nœud adjacent qui minimise la fonction objectif *(voir 2.4.1.2).*

L'algorithme s'arrête quand aucun nœud adjacent ne peut optimiser la fonction objective.

Cet algorithme de clustering a tous les avantages de PAM et de CLARA avec une complexité linéaire par rapport au nombre d'objets. D'ailleurs, l'expérimentation des auteurs a montré qu'il fournit toujours les meilleurs résultats par rapport au PAM et CLARA dans tous les cas [26].

2.4.2. Les algorithmes hiérarchiques :

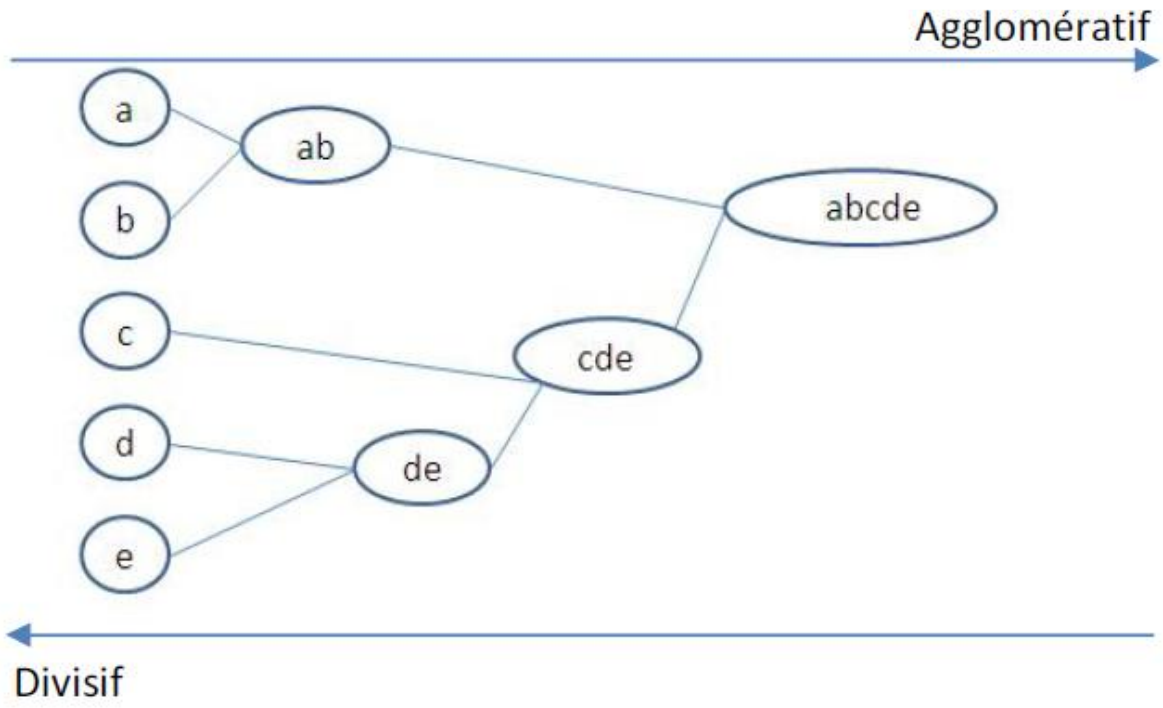
Ces algorithmes construisent les classes graduellement sous une forme hiérarchique, autrement dit, un arbre des classes appelé ''dendrogramme'' (Figure 6).

Ils sont divisés en 2 types : Agglomératifs (ascendants) et divisifs (descendants).

En agglomération, on commence par considérer chaque objet comme une classe et on essaye de fusionner deux ou plusieurs classes appropriées (selon une similarité) pour former une nouvelle classe. Le processus est répété jusqu'à atteindre un critère d'arrêt.

En division, tous les objets sont au début considérés comme une seule classe, on divise successivement les classes en classes plus raffinées selon une similarité. Le processus est répété jusqu'à atteindre un critère d'arrêt.

Le critère d'arrêt peut être le nombre de classes désiré, le nombre minimum (ou maximum) d'objets dans chaque classe, le nombre d'itérations, l'inertie...etc.



Il y a de nombreux algorithmes hiérarchiques qui sont proposés dans la littérature, les plus connus sont : BIRCH pour le groupement agglomératif et DIANA pour le groupement divisif.

2.4.2.1. L'algorithme BIRCH:

BIRCH (*Balanced Iterative Reducing and Clustering using Hierarchies*) est un algorithme de clustering hiérarchique agglomératif. Il a été Introduit par Zhang, Ramakrishan et Livny [12] et [17].

La communauté du domaine de classification trouve que BIRCH est l'un des meilleurs algorithmes pouvant traiter de gros jeux de données [17].

L'idée principale est que la classification est effectuée sur un ensemble d'objets compactée et organisée en une structure d'arbre équilibré appelé CF_Tree (*Clustering Feature Tree*) de taille limitée, au lieu qu'elle s'effectue sur l'ensemble des objets original. CF_Tree est une structure hiérarchique ou chaque niveau représente une phase de clustering.

Parmi les avantages de BIRCH cités dans [7], on trouve :

- La capacité de traiter de grands volumes de données ;
- La structure CF_Tree peut être ajustée à l'espace mémoire disponible ;

Cependant, certaines faiblesses ont été constatées telles que :

- L'algorithme ne considère que les données numériques (dans sa version originale) ;
- Il est fortement sensible à l'ordre d'arrivée des données.

2.4.2.2. L'algorithme DIANA :

DIANA est un algorithme hiérarchique qui procède par division. Son fonctionnement repose sur un principe simple. Au départ, tous les objets appartiennent à un seul et unique groupe.

Un critère de partitionnement est, ensuite, utilisé pour diviser le groupe en deux sous-groupes et ensuite pour diviser chaque sous-groupe récursivement jusqu'à ce que chaque groupe contienne un seul élément.

2.4.3. Les algorithmes basés sur la densité :

Ces algorithmes considèrent les classes comme étant des régions denses dans l'espace d'objets. Un objet de l'espace est dense si le nombre de ses voisins dépasse un certain seuil fixé à priori. Ils essayent alors d'identifier les classes en se basant sur la densité des objets dans une région. Les objets sont alors groupés non pas sur la base d'une distance mais sur la base de la densité de voisinage excède une certaine limite.

Parmi les algorithmes les plus connus dans cette catégorie, nous citons DBSCAN et OPTICS [12], [23].

Avant de développer les algorithmes, il faut d'abord définir quelques concepts [27] :

Soient :

X : ensemble d'objets ;

ε : rayon maximum de voisinage ;

MinPts : nombre minimum d'objets dans le voisinage d'un point.

Le voisinage d'un objet *p* par rapport à ε : Vε (*p*) = {*q* ϵ X / distance (*p*, *q*) ≤ ε }.

Un objet *p* est ''directement densité-accessible'' à partir de *q* (Figure 7) si :

- *p* ı Vε (*q*)
- |Vε (*q*)| > MinPts (q est dit noyau)

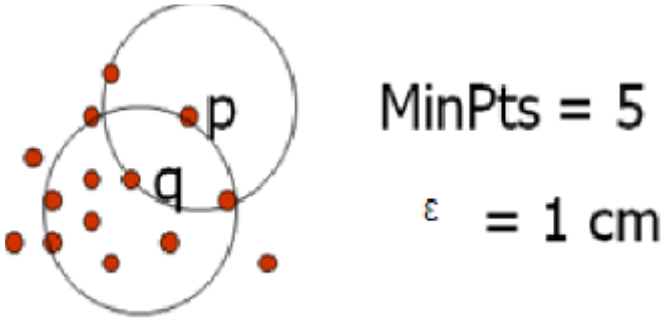


Figure 7 : *Illustration du concept ''directement densité-accessible'' [27]*

Un objet *p* est ‘‘accessible’’ à partir de *q* s’il existe *p1, …, pn, p1 = q, pn = p* tel que *pi+1* est directement densité-accessible à partir de *pi* pour 1 < *i* < *n*. (Figure 8)

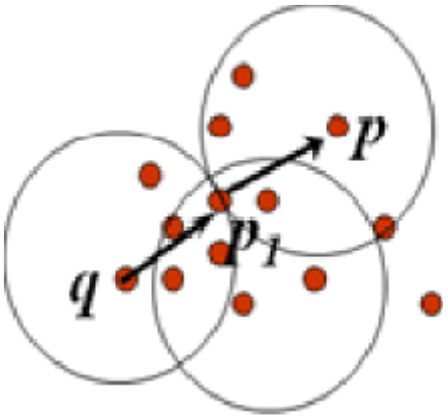


Figure 8 : *Illustration du concept ‘‘accessible’’ [27]*

Un objet *p* est ‘‘connecté’’ à *q* s’il existe un objet *o* tel que *p* et *q* soient accessibles à partir de *o* (Figure 9).

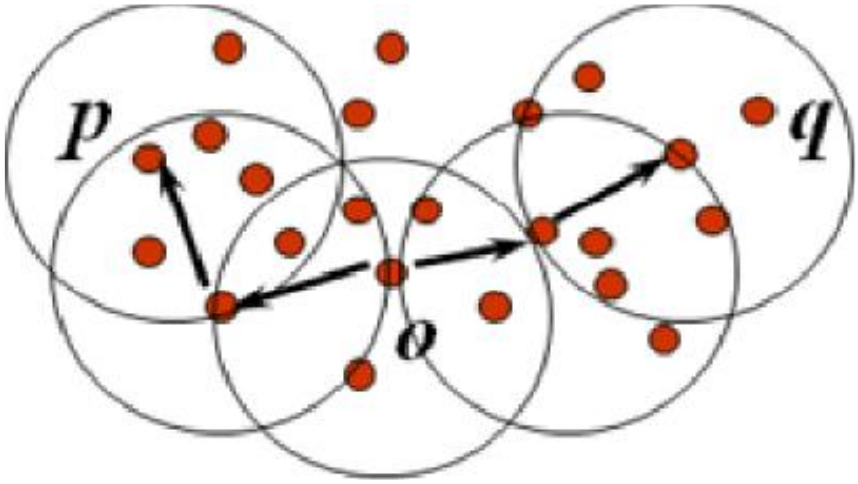


Figure 9 : *Illustration du concept ‘‘connecté’’ [27]*

Une classe est l’ensemble maximal des objets connectés [27].

2.4.3.1. L’algorithme DBSCAN:

DBSCAN (*Density Based Spatial Clustering of Application with Noise*) est introduit par Ester, Kriegel, Sander et Xu [12]. La démarche générale de DBSCAN se résume comme suit [27] :

1. Sélectionner aléatoirement un objet *p* de l’ensemble d’objets *X*.
2. Déterminer l'ensemble *E* des objets accessible depuis *p* dans *X*.
3. Si *p* est un noyau, alors *E* est une classe.

4. Sinon sélectionner un autre objet et aller en (2).

5. S'arrêter lorsque tous les objets ont été sélectionnés.

Comme avantages de DBSCAN, on cite [17] :

- Il peut chercher les classes de forme arbitraire ;
- Il est insensible à l'ordre d'arrivée des objets ;
- Il est incrémental car un nouvel objet qui arrive peut affecter certains voisinages.

Parmi les inconvénients de cet algorithme, on cite [17] :

- Il est difficile à préciser le rayon de voisinage ε ;
- La performance peut être diminuée pour les données de haute dimension.

2.4.3.2. L'algorithme OPTICS :

L'algorithme OPTICS a été introduit par Ankerst, Breunig, Kriegel et Sandar pour pallier au premier inconvénient de l'algorithme DBSCAN. Il consiste à effectuer un clustering DBSCAN pour plusieurs valeurs du rayon de voisinage de manière simultanée [12]. L'algorithme procède de la manière suivante :

1. Commencer par créer une liste ordonnée de différents rayons ;

2. Appliquer ensuite DBSCAN, en parallèle, à cette liste ;

3. Choisir la plus petite distance qui assure la plus forte densité.

2.4.4. Autres algorithmes :

2.4.4.1. Les algorithmes basés sur la grille :

Le principe de cette famille d'algorithmes est qu'on divise l'espace de données en cellules (fusion des régions jugées denses, selon la valeur d'un seuil fixé à priori, les régions jugées peu denses permettent d'établir des frontières.). Donc ce type d'algorithme est conçu pour des données spatiales. Une cellule peut être un cube, une région ou un hyper rectangle. En fait, elle est un produit cartésien de sous intervalles d'attributs de données.

Avec une telle représentation des données, au lieu de faire la classification dans l'espace de données, on la fait dans l'espace spatial en utilisant des informations statistiques des objets dans la cellule. Les algorithmes les plus connus dans cette catégorie sont STING, CLIQUE et WaveCluster [23] et [17].

2.4.4.2. Les algorithmes basés sur les statistiques :

Ces algorithmes démarrent du principe que les données de départ ont été générées suivant une certaine loi de distribution (normale, uniforme, exponentielle,...)

Le principe général est d'utiliser le modèle statistique permettant d'approcher au mieux cette distribution pour réaliser le groupement.

Parmi les systèmes de classification basés sur des modèles statistiques, on peut trouver COBWEB (Fisher), CLASSIT (Gennari, Langley et Fisher) et AutoCass (Cheeseman et Stutz) [12].

2.5. Conclusion :

Nous avons constaté à travers cette étude des algorithmes de clustering qu'il en existe plusieurs. Ces algorithmes se distinguent principalement par :

- Le type des attributs manipulés (k-Means par exemple ne peut traiter que des attributs numériques) ;

- *La complexité de calcul :* un algorithme de grande complexité (PAM par exemple) a un coût de calcul élevé et s'adapte mal aux cas d'un jeu de données important ;

- *Le critère d'arrêt :* certains algorithmes exigent un critère de qualité pour l'arrêt, comme le cas des algorithmes de partitionnement.

Le choix d'un algorithme approprié dépend fortement du contexte de son application, la nature des données et les ressources disponibles. Une analyse attentive des données aide à bien choisir le meilleur algorithme à partir du moment qu'il n'existe pas un algorithme qui peut répondre à toutes les demandes.

3. La classification automatique supervisée :

3.1. Introduction :

La classification supervisée est une tâche largement appliquée dans la vie courante. En effet, il existe une multitude de problèmes qui entrent dans ce cadre, parmi lesquels on trouve la reconnaissance des caractères manuscrits, la reconnaissance des paroles, la catégorisation des textes, la détection des spams, l'aide au diagnostic médical, la bioinformatique…etc.

A la différence de la classification non supervisée où les groupes d'objets sont découverts à posteriori, les techniques de classification supervisée s'appliquent lorsqu'on veut rattacher un nouvel objet (observation) à une classe qui est choisie parmi un ensemble de classes connues. Cette étape suit, généralement, l'étape de clustering.

La plupart des algorithmes de classification tentent de trouver un modèle (une fonction mathématique) qui explique le lien entre les données d'entrée et les classes de sortie. Un ensemble d'apprentissage (ensemble de classes) est donc utilisé par l'algorithme. Cette méthode de raisonnement est appelée inductive car on induit la connaissance (le modèle) à partir des données d'entrée (objets à classifier) et des sorties (leurs classes). Grâce à ce modèle, on peut alors prédire les classes de nouvelles données. Le modèle est bon s'il permet de bien prédire. Un exemple de cette catégorie d'algorithmes est le réseau de neurones.

Une autre approche qui n'est pas moins intéressante, c'est le raisonnement à partir des cas. Ces algorithmes ne cherchent pas à calculer le modèle mais à trouver, pour l'objet à classifier, un ou plusieurs cas similaires déjà résolus pour en déduire la classe. Les arbres de décision et l'algorithme k-NN adoptent ce principe.

Dans ce qui suit, nous allons décrire et détailler les principaux algorithmes de classification automatique supervisée.

3.2. Problématique :

On dispose d'un ensemble *X*, comportant *N* données étiquetées (dont la classe est connue). Chaque donnée *x* est caractérisée, par *P* attributs et par sa classe *Ci ϵ C, C* étant l'ensemble des classes.

Le problème consiste alors, en s'appuyant sur l'ensemble *X* = { (*xi; Ci*)*; i ϵ {1..N}* }, à prédire la classe de toute nouvelle donnée *x*.

3.3. Les algorithmes de classification :

Parmi les algorithmes de classification supervisée les plus populaires dans la littérature [24], on trouve : les arbres de décision, l'algorithme k-NN et les réseaux de neurones.

3.3.1. Les arbres de décision :

Un arbre de décision est, comme son nom l'indique, un outil d'aide à la décision qui permet de classifier une population d'individus selon les valeurs de leurs attributs. C'est une représentation graphique de la procédure de classification où :

- Une feuille indique une classe ;

- Un nœud spécifie un test que doit subir un certain attribut ;

- Chaque branche sortant de ce nœud correspond à une valeur possible de l'attribut en question (Figure 10).

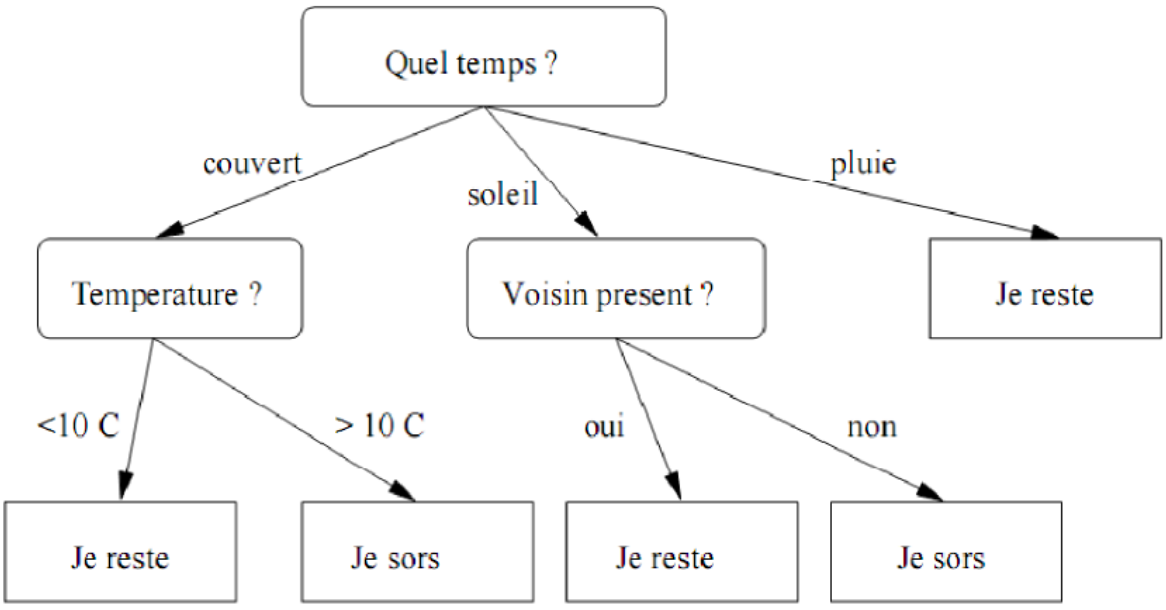


Figure 10 : *Arbre de décision [28]*

Pour classifier un nouvel objet, on suit le chemin partant de la racine (noeud initial) à une feuille en effectuant les différents tests d'attributs à chaque noeud. L'arbre permet d'émettre des prédictions sur les données par réduction, niveau par niveau, du domaine de solutions.

La démarche générale de construction de l'arbre de décision consiste en deux étapes:

- Construction de l'arbre à partir des données (apprentissage) ;

- Elagage de l'arbre dans le but d'alléger l'arbre résultant souvent volumineux.

Construction de l'arbre :

Il existe une grande variété d'algorithmes pour construire des arbres de décision ; quelques uns des plus répandus portent les noms de ID3 (*Inductive Decision-tree*) introduit par Quinlan et amélioré pour devenir C4.5 [29], CART (*Classification And Regression Trees*, introduit par Breiman et al [30] et CHAID (*Chi-squared Automatic Interaction Detection*) [11].

Le principe général démarre d'un arbre vide et procède à la construction de l'arbre de manière inductive (à partir des données) et récursive en commençant par l'ensemble des objets tout entier. Si tous les objets sont de même classes, une feuille est créée avec le nom de la classe. Sinon, l'ensemble d'objets est partagé en sous-ensembles selon la valeur d'un certain attribut qui subissent le même traitement.

Afin d'avoir un arbre de décision concis et suffisant, il ne suffit pas de traiter les attributs séquentiellement. Au contraire, toute la richesse des arbres de décision consiste à choisir judicieusement les attributs d'éclatement pour aboutir, par le chemin le plus court et nécessaire, au plus grand nombre d'objets de la même classe.

Le choix des attributs peut se faire par plusieurs techniques, entre autres :

- Entropie (ID3, C4.5) [29] ;

- Indice de Gini (CART) [29] ;

- Table de Khi-2 (CHAID) [24].

Elagage de l'arbre :

L'opération d'élagage de l'arbre se fait en deux phases : Le pré-élagage et le post-élagage.

Le pré-élagage consiste à fixer un critère d'arrêt qui permet de stopper la construction de l'arbre lors de la phase de construction.

Le post-élagage est un traitement qui intervient après la construction entière de l'arbre. Il consiste à supprimer les sous-arbres qui n'améliorent pas l'erreur de classification.

Les arbres de décision constituent un moyen très efficace de classification, et ce pour les avantages qu'elles présentent. Parmi ses avantages, on peut citer [30] [5] :

- Facilité à manipuler des données catégoriques ;
- Traitement facile des variables d'amplitudes très différentes ;
- La classe associée à chaque individu peut être justifiée ;
- Les attributs apparaissant dans l'arbre sont des attributs pertinents pour le problème de classification considéré ;

Ces méthodes présentent tout de même des inconvénients dont les plus importants sont :

- La sensibilité au bruit et aux points aberrants ;
- La sensibilité au nombre de classes (plus le nombre de classes est grand plus les performances diminuent) ;
- Le besoin de refaire l'apprentissage si les données évoluent dans le temps.

3.3.2. L'algorithme k-NN :

k-NN (*k Nearest Neigbours*) est un algorithme de raisonnement à partir de cas c'est-à-dire prendre des décisions en recherchant un ou plusieurs cas similaires déjà résolus. La décision consiste à chercher les *k* échantillons les plus voisins de l'objet et de l'affecter à la classe qui est la plus représentative dans ces *k* échantillons (''*dis-moi qui sont tes amis, et je te dirais qui tu es*'').

L'approche la plus simple est de rechercher le cas le plus similaire et de prendre la même décision, on parle de 1-NN. Si cette approche peut fournir des résultats acceptables sur des problèmes simples pour lesquels les objets sont bien répartis en groupes denses de même classe, en règle générale, il faut considérer un nombre de voisin plus important pour obtenir de bons résultats [5].

L'algorithme k-NN a les avantages de :

- Ne pas refaire l'apprentissage lors de l'introduction de nouveaux attributs ;
- Traiter tout type de données avec un nombre d'attributs élevé ;
- Fournir des résultats clairs.

Cependant, le coût de classification peut devenir cher car :

- D'une part, le temps pour calculer les voisins peut être prohibitif ;
- D'autre part, il faut toujours stocker le modèle durant toute l'opération de classification.

3.3.3. Les réseaux de neurones :

Un réseau de neurones est un système composé de plusieurs unités de calcul simples (nœuds) fonctionnant en parallèle, dont la fonction est déterminée par la structure du réseau et l'opération effectuée par les nœuds.

Le principe de fonctionnement est le suivant : On dispose initialement d'une base de connaissances constituée de couples de données (entrées / sorties) et on souhaite utiliser cette base de données pour entrainer un algorithme à reproduire les associations constatées entre les entrées et les sorties de l'échantillon.

L'exemple le plus simple de réseau de neurones est souvent donné par le ''perceptron multicouches'' qui est un cas particulier de réseau de neurones (*Figure **12***).

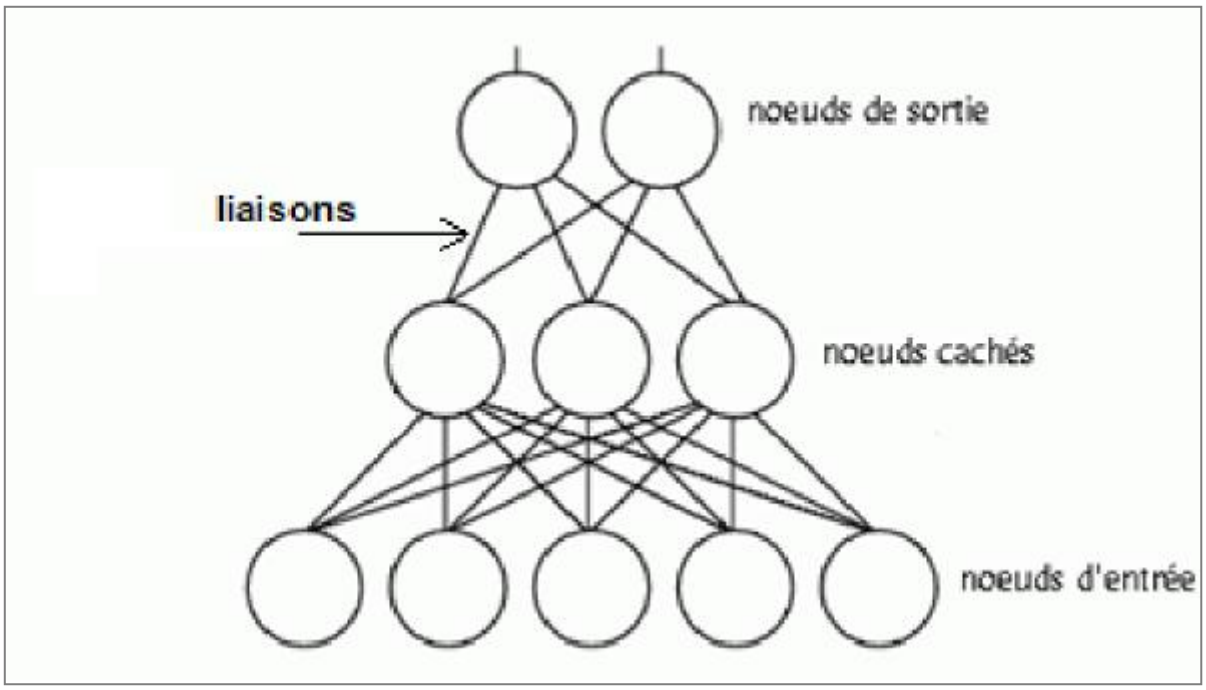


Figure 12 : *Perceptron à trois couches (schéma type) [28]*

Pour un réseau de neurones avec *N* nœuds d'entrée, notées C(1),...,C(N) , et N poids affectées aux liaisons et notés w(1),...,w(N) l'entrée d'un nœud de la couche suivante sera généralement une somme pondérée des valeur de sortie des neurones précédents :

X = w(1)*C(1) + w(2)*C(2) + w(3)*C(3) + ... + w(N)*C(N)

Les poids sont des paramètres adaptatifs, dont la valeur est à déterminer en fonction du problème via un algorithme d'apprentissage (propagation, rétro-propagation…) [30].

Les réseaux de neurones peuvent être utilisés pour effectuer une classification supervisée floue de la manière suivante : chaque nœud d'entrée correspond à un attribut de l'objet (Autant de nœuds d'entrée que d'attributs). On peut prendre un neurone de sortie par classe; la valeur de sortie est la valeur de la fonction d'appartenance (probabilité que l'objet appartienne à cette classe) [30].

V. La Confidentialité :

La manière de faire les affaires a complètement changée par l'avancée de la technologie de l'information, passant de transactions offline aux transactions en temps réel.

L'activité de la plupart des personnes aujourd'hui dépend de la disponibilité de cette information online, la banque online par exemple pour leurs tâches quotidiennes. En changeant la manière de concevoir cette transaction, les gens peuvent obtenir plus.

C'est pour cela que nous pouvons affirmer que nous sommes dans l'ère d'information, parfois même en surcharge même.

Forcé par les demandes des clients ne cessant de croître et le marché concurrentiel dans lequel évoluent les entreprises, celles-ci sentent la nécessité de rassembler le plus d'informations possibles à partir des personnes. Ceci peut les aider à atteindre l'efficacité dans leurs organisations. Mais, d'un autre coté, le souci de l'intimité est une raison principale qui pousserait les gens à ne pas vouloir se risquer à des échanges (commerce, achat . . .) online, mais qui seraient disposés à le faire si leurs intimité leurs soit assurée.

Il est important d'assurer la protection des données et leur intimité (Privacy) dans tout système d'information, spécialement dans les applications d'aujourd'hui basées sur le WEB essentiellement. Il y va de soi de différencier entre l'intimité, la confidentialité et la sécurité. [Gotlieb 1995] différencie entre trois notions que sont : l'intimité, la confidentialité et la sécurité:

Intimité (Privacy) est un concept social, culturel et légal, chacun de ces trois aspects qui peuvent varier d'un pays à l'autre.

Confidentiality est une responsabilité en terme de gestion : elle concerne les problèmes qui traitent la manière gérer les données par les règles qui à la fois satisfassent les managers des banques de données et les personnes à partir desquels les données sont sous-traitées.

Security est un problème technique. Il se focalise sur la façon dont les règles d'accès aux données établies par l'organisme qui stocke les données peuvent être forcées, par l'utilisation des mots de passe, cryptographie, et les techniques du genre.

Les internautes veulent et espèrent à la confidentialité dans ce qu'ils entreprennent, quand ils échangent des données électroniques (Transactions électroniques). La notion de confidentialité devient une caractéristique importante dans tous les aspects de la vie d'une société moderne particulièrement quand elle vient à la révélation d'informations confidentielles. Comme expliqué dans la section précédente, la confidentialité est un problème concernant la révélation des informations confidentielles, qui se rapporte généralement à l'information personnelle. La protection de l'intimité est nécessaire pour éviter l'utilisation inadéquate et les utilisations illégales des informations personnelles.

La confidentialité est à l'économie de l'information du siècle prochain ce que la protection des consommateurs et les soucis environnementaux ont été à la société industrielle du 20ème siècle [Gleick 1996]. Les nouvelles technologies dans le monde entier ont affecté différents aspects de la manière d'aborder les informations personnelles dans des champs d'application variés tel que la sécurité, commerce....Il y a eu beaucoup d'efforts de définir la confidentialité. Cette difficulté d'arriver à une simple définition de la confidentialité a eu comme conséquence des approches multidimensionnelles.

Elle peut être définie dans divers contextes tels que le domaine légal et celui économique. Dans un contexte légal, l'intimité est en grande partie synonyme du droit d'être seul (solitude) [S.Warren 2008]. Du point de vue et contexte économique, l'intimité est définie comme la dissimulation des informations utiles ayant une valeur économique dans la transaction (Stigler, 1980). Beaucoup de personnes ont proposé différents points de vue concernant la confidentialité.

Kang [Kang 1998] affirme que l'intimité (confidentialité) concerne le contrôle du flux d'information personnelle dans toutes les étapes du traitement - acquisition, traitement, révélation, et utilisation.

[Goldberg,Wagner 1997] définit la confidentialité comme la capacité de contrôler la collecte, la conservation et la distribution des informations.

Burgoon, et al.[Burgoon 1989] définit la confidentialité comme " capacité de contrôler et limiter l'accès physique, interactionnel, psychologique et informationnel " appliquée à l'individu ou à un groupe d'individus. La confidentialité est définie comme le droit des individus de déterminer par eux-mêmes quand, comment et dans quelle mesure leurs informations seront communiquées et de quelle manière.

Al-Fedaghi, [Al-Fedaghi 2005a] cite la différence entre le secret et la confidentialité. Il déclare que l'information " confidentielle " se rapporte généralement à n'importe quelle information qui doit être maintenue confidentielle, pour qui la divulgation exige le consentement de son propriétaire. Cela implique la protection des secrets d'autres personnes par le contrôle de l'accès aux informations et aussi le contrôle de la divulgation en suivant un certain accord de principe (agreement).On peut en conclure donc de toutes ces définitions que le respect de l'intimité est à la base de toute application de la confidentialité.

1. Définition :

Une définition du dictionnaire standard de la confidentialité qui a trait aux données est la «liberté de l'intrusion non autorisée» [58]. En ce qui concerne la préservation de la vie privée dans le data mining, cela va nous éclairer. Si les utilisateurs ont donné l'autorisation d'utiliser les données de la tâche particulière de data mining, alors il n'y a pas de problème de confidentialité.

Toutefois, la deuxième partie est plus difficile: Si l'utilisation n'est pas autorisée, c'est ce qui constitue l'utilisation d'«intrusion»?

Une norme commune parmi la plupart des lois sur la confidentialité (par exemple, les lignes directrices de la vie privée de la Communauté européenne [26] ou des lois de santé américaine [40]) est que la confidentialité s'applique uniquement aux "données identifiables individuellement". Combinant intrusion et individuellement identifiables conduit à une norme pour juger la préservation de la vie privée dans le data mining: Une technique d'extraction des données préservant la vie privée doit s'assurer que toute information divulguée :

Ne peut pas être attribuée à un particulier;

Ne constitue pas une intrusion.

Définitions formelles de ces deux articles sont un défi ouvert. À un extrême, on pourrait supposer que les données qui ne nous donnent pas la connaissance exacte au sujet d'un individu spécifique répondent à ces critères. À l'autre extrême, toute amélioration de nos connaissances sur une personne peut être considérée comme une intrusion. Ce dernier est particulièrement susceptible de causer un problème pour l'exploration de données, car l'objectif est d'améliorer notre connaissance. Même si l'objectif est souvent des groupes d'individus, avoir plus de connaissances sur un groupe augmente nos connaissances sur les individus dans le groupe. Cela signifie que nous devons mesurer à la fois les connaissances acquises et notre capacité à les relier à une personne en particulier, et de déterminer si ceux-ci dépassent les seuils.

Pour pouvoir relier la confidentialité au domaine de la préservation de la vie privée dans l'extraction des données, c'est très important de rappeler que toutes les divulgations à partir de l'extraction des données doivent être prises en compte. Cela inclus les divulgations des ensembles de données qui ont été modifiées ou mélangées garantir la confidentialité, les communications entre chaque parties qui participent au processus d'extraction et la révélation des résultats de l'extraction des données.

PPDM (Privacy Preserving Data Mining) est un domaine émergeant qui étudie comment les algorithmes de datamining affectent la protection de la vie privée et essaye de trouver et d'analyser de nouveaux algorithmes qui vont respecter cette contrainte de vie privée.

Comme scénario illustratif de ce problème de confidentialité, nous pouvons citer l'exemple de deux compagnies de bioinformatique. Chaque compagnie possède une base de données gigantesque constituée de mesures collectées à partir d'expériences effectuées dans leurs laboratoires. Les deux compagnies sont prêtes à coopérer pour réaliser une tâche d'apprentissage d'intérêt commun, mais aucune d'elles ne souhaite communiquer sa base de données en clair. Comment peuvent-elles alors atteindre cet objectif sans divulguer aucune information sensible ?

2. Types de Confidentialités:

Quelques auteurs ont classé l'intimité en deux catégories. [Tavani 1999] différencie entre la confidentialité de l'information et celle de sa communication. D'autres différencient entre problèmes de confidentialité d'une l'information ou d'une base de données, et ceux de la confidentialité des communications en plaçant dans la dernière (3ème) catégorie l'ensemble de soucis de confidentialité relatifs aux technologies telles que la surveillance électronique, chiffrement (cryptage), email, téléphonie numérique...

Les problèmes de protection des données sont ceux liées à l'information personnelle enregistrée dans une base de données. Plusieurs types confidentialité ont été abordés en littérature comprenant la confidentialité physique et la confidentialité informationnelle.

Al-Fedaghi, [Al-Fedaghi 2007] a séparé la confidentialité en deux catégories ;

1) confidentialité des informations personnelles (individuelles)

2) confidentialité des informations non personnelles. Cette dernière peut appartenir à des catégorisations de la confidentialité qui incluent ces types de confidentialité comme un genre de confidentialité physique, en analogie avec la confidentialité des informations personnelles qui elles, impliquent des actes sur les informations personnelles.

De cette classification, on note qu'il y a deux types de confidentialité; confidentialité de l'information et confidentialité de non-information. La confidentialité de non information peut être une partie de confidentialité physique ou confidentialité de communications.

D'autres auteurs comme [Clarke 1999] donnent plusieurs dimensions à la confidentialité:

- ***La confidentialité de la personne:*** parfois désignée sous le nom " intimité corporelle". Cela concerne l'intégrité du corps de l'individu. Des exemples peuvent être cités comme la transfusion sanguine sans consentement de l'individu, l'extraction forcée d'échantillons de liquides corporels et de tissu du corps de la personne, l'immunisation forcée, la stérilisation forcée ...

- **La confidentialité du comportement individuel:** Ceci est associé à tous les aspects du comportement, mais particulièrement aux sujets sensibles, tels que les préférences sexuelles et les habitudes, les activités politiques et pratiques religieuses, dans les lieux privés et publics. Il inclut ce qui est désigné parfois sous le nom de " media privacy ";

- **La confidentialité des communications personnelles:** Les individus cherchent leurs intérêts derrière leurs capacités de communiquer entre eux, et ce par divers moyens, sans surveillance courante de leurs communications par d'autres personnes ou organismes. Ceci inclut ce qui est désigné parfois sous le nom de la "confidentialité d'interception (Interception Privacy) ";

- L'intimité des données personnelles: Les individus exigent que les données portant sur eux-mêmes ne devraient pas être disponibles de façon automatique et servir ainsi à d'autres personnes et organismes, et ce, même dans le cas où les données sont possédées par une autre partie, l'individu (la personne) doit pouvoir exercer un degré substantiel de contrôle sur ces données et leurs utilisation. Ceci est désigné parfois par l'appellation " confidentialité des données " et " confidentialité de l'information ".

2.1. Information Personnelle

Les données ont leurs importance dans n'importe quelle transaction ; dans le cas des transactions off-line ou celles on-line. Malheureusement, un nombre réduit de personnes prennent compte et comprennent l'importance de leurs informations personnelles.

Quelques unes de ces informations personnelles peuvent être classifiées comme sensibles et nécessitent d'être considéré comme cite informations privées. D'autres sont sensibles mais il n'y a pas besoin de les considérer comme privées. Il y a en tout quatre types de données impliqués dans le traitement des données [Guarda,Zannone 2008] :

1. Données personnelles : toutes données qui peuvent être employées pour identifier une personne telle que son nom, adresse, numéro de téléphone...

2. Données sensibles : n'importe quelles données qui révèlent des informations sur l'origine raciale ou ethnique, religieuse, philosophique ou autre convictions politiques, appartenance à des parties, et aussi les données personnelles qui révèlent l'état de santé telle que l'historique de santé.

3. Données d'identification : données personnelles qui permettent l'identification directe du sujet (individu) tel que l'ADN, numéro de carte d'identité, . . .

4. Données anonymes : toutes données qui ne peuvent pas n'être associées à un sujet identifié ou identifiable tel que le genre, la type de maladie, . . .

De la classification ci-dessus, les trois premiers types de données peuvent être considérés comme informations privées. Les informations privées sont les informations personnelles qui exigent une protection. Cela est du aux risques qui pourraient résulter de leurs révélations, changements, ou destructions. Ces informations personnelles et privées devraient être protégées pour assurer l'aspect privacy.

Les données personnelles vont subir une modification qui les ramènera au stade d'information. Il y a de diverses définitions concernant l'information personnelle. En 1993, elle a été définit comme n'importe quelle information qui est liée à l'individu. Elles sont souvent interprétées comme toute information individuelle possédée par une personne, telle que les blocs- notes de calendrier, les adresses de contacts et des amis d'un individu,. . ..

Bergman et al. [Bergman 2004] les a définies dans un contexte de management des informations personnelles (gestion des informations personnelles) ; il s'agit du stockage, l'organisation, et la recherche par un individu donné, de ces informations pour son usage personnel. On définit l'information personnelle comme n'importe quelle information liée à l'individu. Elle est toute expression linguistique qui référencie la personne [Al-Fedaghi 2005].Il y'en a trois types de ces références :

- Zero Personal Information : ne possède aucun référent individuel.

-Informations Personnelles Atomiques (Atomic Personal Information) : elle possède un référent simple.

- Les informations personnelles composées (Compound Personal Information) : elle peut avoir plus d'un référent.

Dans environnement à base de web, l'information personnelle est révélée par le propriétaire des données et employé par les différents organismes. Ces derniers vont effectuer le rassemblement, la sauvegarde, et la manipulation de cette information dans le but d'accomplir leurs besoins. D'un point de vue système d'information, la confidentialité de l'information peut servir à protéger les individus de l'abus de leurs données dont ils pourront faire l'objet, l'accès non autorisé à une partie ou la totalité des données, la modification des informations, ou tout ce qui pourrait atteinte au propriétaire de cette information. L'information joue donc un rôle fondamental dans le domaine de la confidentialité du au fait qu'elle fait l'objet de rassemblement, manipulation, sauvegarde, et déploiement selon des besoins bien précis. Clarke [Clarke 1999] définit la confidentialité comme suit : l'intérêt que les individus ont à exercer "un espace personnel", exempté d'interférences venant d'autres individus et organismes.

2.2. Catégories d'Informations Personnelles

Il y a deux types d'informations personnelles ;

1) information personnelle privée et

2) information personnelle non-privée. L'information personnelle privée se rapporte à tout individu identifiable de manière unique, et possédée par une personne, alors que l'information personnelle non-privée, elle ne se rapporte pas à la personne identifiable de manière unique. Les informations privées représentent toute information qui contient des informations privées sur l'individu lui-même. L'information non privée est toute information possédée par une personne mais non considérée comme information privée, telles que les calendriers, les cartes géographiques, …

Les informations personnelles doivent être gardées par le propriétaire des données lui-même ou bien il doit être en mesure de contrôler leurs révélations afin d'assurer que son intimité ne soit pas remise en cause. Mais, dans des applications WEB, cette information devrait être révélée afin de permettre l'accomplissement des transactions.

Bien que ces informations personnelles fassent l'objet de révélation à autrui, normalement, pour des raisons de sécurité et de confidentialité, elles ne devraient pas être consultées par des utilisateurs non autorisés à le faire. Pour cette raison, il y a trois principes doivent qui doivent être considérés :

1. L'information personnelle ne devrait pas être accessible par des utilisateurs non autorisés.

2. Seul une information personnelle requise est mise à disponibilité d'autrui.

3. l'information personnelle ne peut en aucun cas être diffusée à ceux qui n'ont en pas besoin.

2.3. Confidentialité des Informations Personnelles

La confidentialité des informations personnelles est la " réclamation faite par un individu, faisant par laquelle valoir son droit de contrôler les termes sous lesquelles son information personnelle (identifiable à cet individu là) est saisie (collectée), révélée et utilisée. ".Cette définition vient du principe de fourniture et celui de l'usage de l'information personnelle (principes d'IITF (Information Infrastructure Task Force) aux Etats Unis. On a introduit cinq principes de base dans le but de satisfaire la confidentialité de l'information :

1. L'acquisition et la conservation : traite quelles informations concernant les individus sont rassemblées et pour combien de temps cette information est gardée par une organisation.

2. L'intégrité: est concernée par le fait de ne maintenir des informations concernant des individus que si celles-ci sont correctes, complètes, et à jour.

3. L'agrégation et la dérivation des données : permet de s'assurer que toute agrégation ou dérivation opérée par une organisation sur sa propre information est nécessaire pour qu'elle puisse effectuer ses taches.

4. Le partage d'informations : est concerné par la divulgation de la personne qui possède l'information ou celle autorisée à le faire, à des individus ou organismes extérieurs. L'information, dans ce cas là, ne devrait être révélée qu'une fois autorisé de manière explicite et bien spécifiée (specifically authorized) et seulement pour l'usage spécifié.

5. Contrôle approprié de l'accès : traite la restriction de l'accès aux données et aux ressources aux seuls individus autorisés qui sont dans le besoin de le faire. Ainsi, l'information ne devrait pas être révélée à ceux qui ne sont pas autorisés à y accéder ou ceux qui ne sont pas dans le besoin de savoir (même si elles sont autorisées à le faire habituellement).

Pour comprendre comment satisfaire la confidentialité d'une information, nous devons comprendre comment celle-ci peut faire l'objet de violation. La confidentialité de l'information est violée quand l'information personnelle est collectée à l'insu des individus mais aussi quand l'information personnelle, qui a pu avoir été donné librement et en toute conscience léguée à autrui, est par la suite utilisée ou de manière générale révélée plus tard en désaccord avec ce qui a été établit au préalable (accord initial).

La prise en conscience du fait que le contrôle de la collecte d'informations personnelles est la première étape dans l'accomplissement de la confidentialité des informations.

En d'autres termes, le contrôle des informations personnelles d'un individu débute au moment de leur collecte. Une fois l'individu choisit de céder la totalité ou une certaine partie de son information personnelle, l'individu se fie à des lois ou des mécanismes pour contrôler l'éventuelle diffusion et l'utilisation ultérieure d'une partie ou de la totalité de cette information.

2.4. Techniques de gestion(Management) de la Confidentialité

Les données représentent une entité de valeur inestimable pour une organisation donnée, particulièrement pour les sociétés d'aujourd'hui ou la plupart des organismes migrent vers des applications en ligne. Ceci nous a amené à voir un nombre croissant d'organismes qui rassemblent des données, des informations personnelles et les utilisent dans différents contextes. Parfois, ces organismes peuvent se voir donner l'accès aux données ou bien avoir le droit de les divulguer à une tierce partie. Ceci pose de sérieux challenges à la protection de la confidentialité des personnes et des organismes. Puisque la confidentialité est un souci important et constamment abordé, d'innombrables efforts de recherches ont été menés en ce sens ; communément appelées techniques de privacy-preserving data management (techniques de gestion de données préservant leurs confidentialité).

Il y en a trois dans l'ensemble ; ce qu'on nomme anonymization, fouille de données (datamining) et adaptation (conception) de la base de données à la politique de confidentialité.

- Mise en Anonymat (Data Anonymization):

Une première technique traite le problème de la conservation de la confidentialité des données quand ces mêmes données doivent être cédées à autrui. Du au fait que les données une fois publiées ne sont plus sous le contrôle de leurs propriétaire (individu ou organisme). Par

conséquent, les organismes ne peuvent pas contrôler la façon avec laquelle leurs données seront manipulées après qu'ils les ont léguées. L'approche la plus évidente à ce propos consiste à modifier les données de telle façon à enlever toute l'information qui pourrait lier directement les données (items) avec les individus ; cette pratique est connue sous le nom de data anonymization.

- Privacy-Preserving Data Mining

Une Deuxième technique qui traite spécifiquement la conservation de la confidentialité dans le cadre de la fouille de données (data mining). L'utilisation des techniques de data mining peut permettre la récupération d'une information supprimée auparavant.

Plusieurs approches ont été proposées, dans la plupart, des outils pour la recherche de règles d'association, ou encore pour les systèmes de classification. Dans l'ensemble, toutes les approches sont basées sur la modification ou perturbation (changement minime) des données dans un sens ou autre ; par exemple, les techniques spécialisées dans la recherche de règles d'associations préservant la confidentialité (privacy-preserving mining of association rules), modifient les données dans le but de réduire la confiance des règles d'association dites sensibles.

- Adaptation de Bases de Données à la politique de confidentialité

En guise de conclusion, la troisième technique vient des efforts préliminaires qui se basent sur les systèmes de bases de données spécifiquement conçus pour supporter la politique de confidentialité, citons comme exemple, les politiques qui peuvent être exprimées en utilisant le P3P. En particulier, Agrawal, et al.[Agrawal 2002] ont introduit le concept de la base de données hippocratique(Hippocratic Database), incorporant la protection de la confidentialité dans les systèmes de bases de données relationnelles. Ils introduisent les principes fondamentaux de cette base de données hippocratique et proposent même une architecture. La caractéristique innovante dans cette architecture définit les métadonnées de la confidentialité comme étant les politiques de confidentialité et les autorisations de cette même confidentialité. La base de données hippocratique effectue la vérification de la confidentialité pendant le traitement de la requête. Dans [Agrawal 2002], Ils discutent également les défis et problèmes techniques au moment de la conception des bases de données hippocratiques, telles que l'efficacité (efficiency), la divulgation (disclosure), la conservation (retention), et la sureté des données.

2.5. Recherche de Savoir (Knowledge Discovery) et Confidentialité

La recherche de la connaissance (Knowledge Discovery) et la fouille de données (Data mining) de puissants outils de l'analyse automatisée des données et qui sont amenées à devenir les outils analytiques les plus utilisés à l'avenir. Cependant, le déploiement rapide de ces technologies nécessite un examen approfondi sur leur incidence sociale.

Le Knowledge Discovery et le Datamining sont employées pour décrire l'extraction non-triviale et implicite d'informations potentiellement utiles et inconnues auparavant. Ces outils emploient le machine learning et les statistiques basés sur la connaissance et les appliquent sur d'énormes bases de données dans le but de révéler des portions d'information intéressantes. Ces outils connaissent une augmentation rapide en matière d'usage.

2.6. Confidentialité et coopération

La confidentialité parfaite n'est pas une partie inhérente d'aucune société [Gavison 1984]. Ceci est du au fait que la participation nécessite la communication, ce qui rend la confidentialité absolue inatteignable. Dans le contexte des pratiques courantes en matière de communication d'une société donnée, les différents membres de cette société développent leurs perceptions indépendantes et uniques de leur propre définition de la confidentialité [Rachels 1975]. Par conséquent, la confidentialité existe dans une société seulement parce qu'elle existe comme perception de ses membres.

Cette perception est cruciale car elle détermine en partie si, et dans quelle mesure, l'intimité d'une personne a été violée. Un individu peut maintenir le contrôle sur son intimité s'il restreint la disponibilité de ses informations personnelles. Si une personne considère le degré d'information connue à son sujet en matière de quantité d'information et type d'information comme étant inapproprié, alors sa confidentialité telle qu'elle la perçoit est en danger. Ainsi, l'intimité peut être violée quand l'information au sujet d'un individu est obtenue, utilisée, diffusée, sans sa connaissance ou son consentement.

En 1980, l'organisation de coopération économique et le développement (OECD :Oragnisation of Economic Cooperation and Development) a mis en place un ensemble de directives pour la protection des données personnelles. Ces directives reconnaissent et supportent le droit des individus à participer au contrôle de leurs informations personnelles. La participation consiste à suivre le maintien de la protection de leur intimité.

Une des directives déclare que les données ne peuvent pas n'être utilisées dans un but autre que celui déclaré quand ces données ont été initialement obtenues à partir de l'individu. Une autre directive déclare que la raison qui pousse à rassembler des données personnelles devrait être clairement définie à l'individu avant de les rassembler. Ces directives permettent à un individu de s'abstenir à fournir des données personnelles, et ce pour n'importe quel but qu'il décide comme étant inadéquat. Le KD (Knowledge Discovery) a le potentiel de contourner et violer ces principes mis en place.

Premièrement, le KD est utilisé généralement avec un très grand nombre de données archivées (historiques) et utilise ainsi des données rassemblées pour un but ou un autre.

Deuxièmement, le but spécifique d'une application de KD est en grande partie inconnu jusqu'à ce qu'elle ait trouvé de l'information précédemment inconnue. C'est-à-dire, son but est intrinsèquement lié à l'information qu'il découvre. En conclusion, l'information révélée pendant le processus du Knowledge Discovery peut se révéler être inadéquate (en terme de type et quantité) par l'individu qu'elle décrit sous une manière ou une autre. Ces contraintes diminuent la capacité de l'individu à participer au contrôle de ses propres données et menacent ainsi de violer son intimité et la confidentialité de ses données.

Il convient noter que le paradigme de Data mining utilisé dans une application de Knowledge Discovery insinue sur l'aptitude à violer la confidentialité. Les trois paradigmes de fouille examinés par [Roddick,Lees 2001] diffèrent dans la façon avec la laquelle les résultats du processus de fouille sont retournés, en l'occurrence :

- Les résultats des routines de DM (Data Mining) sont directement retournés à l'utilisateur ;

- Les résultats du processus de DM sont encapsulés (embedded) dans un processus qui interprète les résultats comme étant pas assez mûres, et effectue davantage de recherches structurées sur l'origine des résultats ;

- Un processus de Knowledge Discovery reçoit une hypothèse et essaye de la raffiner par la modification de l'hypothèse résultante du processus de Data Mining.

Dans les deux dernières techniques, les résultats du processus complet de KD (incluant les recherches sous-jacentes) ont moins de risque de compromettre la confidentialité d'un individu (telle qu'il la perçoit) que l'ancienne technique (La première). Cela est du à la grande signification de plus en plus croissante des résultats de DM. Si la fouille de données (Data Mining) avait été appliquée à des données incorrectes ou sensibles, le risque de se retrouver avec une hypothèse modifiée de manière inappropriée augmente.

Sachant que la confidentialité des données est le produit de perceptions individuelles, une solution infaillible et universelle à ce dilemme est tout simplement impossible.

Une solution acceptable pour une personne peut être insuffisante ou inacceptable pour d'autres. Cependant, il y a des mesures qui peuvent être entreprises pour améliorer la confidentialité.

Les pratiques en vigueur échouent à la mise en place d'un tel idéal. Généralement, un individu doit adopter une attitude proactive et autoritaire dans le but de préserver la confidentialité de ses données, allant jusqu'à entamer des discussions avec les détenteurs de ses données pour que soient appliquées toutes les restrictions qu'il considère appropriées.

D'un autre coté, beaucoup d'individus ne sont pas conscients de l'ampleur des données personnelles enregistrées par des gouvernements (états) et des sociétés privées.

De telles données peuvent contenir des informations sensibles telles que les habitudes d'achats, la race de l'individu, les antécédents médicaux, ... Ce n'est généralement que quand les choses vont mal que les individus exercent leurs droits d'obtenir ces données et qu'ils cherchent à éliminer les brèches ou les corriger.

La sécurité de manière habituelle dans les bases de données consiste à protéger les données par l'intermédiaire des techniques d'autorisation, ne faisant ainsi aucune distinction entre les degrés de sensibilité présents dans la base de données. Un modèle plus sophistiqué, du nom de MLS (Multi Level Security) étend les mesures de sécurité conventionnelles en faisant la classification des données selon leur confidentialité. Les données dans une base de données de type MLS sont typiquement répartis dans quatre niveaux de sécurité, avec des utilisateurs n'ayant accès que dans les niveaux pour les quels ils sont autorisés. Ceci a pour conséquence d'augmenter la protection des données de l'abus par des utilisateurs autorisés ou non.

Le chiffrement (Encryption) est une technique connue dans la sécurité des bases de données. Cette approche a été offerte commercialement dans les mises à jour Oracle8i, qui encryptent les items de données. Le chiffrement est particulièrement applicable aux bases de données accessibles par l'intermédiaire du Web, en raison de l'exposition et de la vulnérabilité des données en dépit des moyens d'accès aux données. Ce n'est pas une technique infaillible de protection des données et une brève recherche dans la littérature nous indiquera de nombreux exemples ou le cryptage des données n'a pas été suffisant.

VI. Conclusion :

Les nouvelles technologies de l'information permettent à la fois un stockage de larges volumes de données et une contribution à leur croissance exponentielle. Ceci a pour conséquence de créer une disproportion entre les volumes de données et les moyens matériels et humains pour les traiter. Face à ces problèmes, les technologies du datamining permettent de traiter ces ensembles de données en vue d'extraire de la connaissance, qui sera déterminante pour la prise de décisions efficaces. Ces techniques font l'actualité des presses spécialisées en informatique, bien sûr, mais aussi dans les rubriques « Informatiques » des presses spécifiques à chaque type d'activité.

Cependant, le problème de la protection de la vie privée devient de plus en plus important ces dernières années à cause des nombreux besoins de partage de données, de la contrainte de la préservation de la vie privée et du besoin, sans cesse croissant, d'extraire de la connaissance à partir de données. Deux problèmes sont étudiés dans ce domaine : le premier est la protection des données privées ; le second est la protection de la connaissance sensible contenue dans les données. C'est pour cela que dans le prochain chapitre nous parlerons de la préservation de la vie privée dans l'extraction de données.

CHAPITRE III

Préservation de la vie privée dans le Data Mining

I. Introduction :

Au cours des dernières années, l'exploration de données a été vue comme une menace à la vie privée en raison de la prolifération généralisée des données électroniques administrés par des sociétés. Ceci a conduit à une augmentation des inquiétudes quant à la confidentialité des données sous-jacentes. Au cours des dernières années, un certain nombre de techniques ont été proposées pour modifier ou transformer les données de telle manière afin de préserver la vie privée.

L'exploration de données et la découverte de connaissances dans les bases de données sont deux nouveaux domaines de recherche qui se penchent sur l'extraction automatique des modèles jusque-là inconnues de grandes quantités de données. Les récents progrès dans la collecte des données, la diffusion de données et de technologies connexes ont inauguré une nouvelle ère de la recherche où les algorithmes d'exploration de données existantes devraient être réexaminés d'un point de vue différent pour la préservation de la vie privée. Il est bien connu que cette nouvelle explosion sans limites de nouvelles informations à travers l'Internet et d'autres médias, a atteint un point où les menaces contre la vie privée sont très fréquentes sur une base quotidienne et ils méritent une réflexion sérieuse.

La principale considération dans la préservation de l'intimité dans l'exploration de données est double. Tout d'abord, les données brutes sensibles comme des identifiants, noms, adresses et autres, devraient être modifiés ou coupés à partir de la base de données originale, afin que le destinataire des données à ne sois pas en mesure de compromettre la vie privée d'une autre personne. Deuxièmement, la connaissance sensible qui peut être exploitée à partir d'une base de données en utilisant des algorithmes d'exploration de données, devrait également être exclue, car une telle connaissance peut aussi bien compromettre la confidentialité des données, comme nous l'indiquerons.

L'objectif principal dans la préservation de la vie privée dans le data mining est de développer des algorithmes permettant de modifier les données originales d'une certaine façon, de sorte que les données personnelles et les connaissances privées restent privées, même après le processus d'extraction. Le problème qui se pose lorsque des renseignements confidentiels peuvent être dérivés à partir de données publiées par les utilisateurs non autorisés est aussi communément appelé le problème «d'inférence de base de données".

Dans ce chapitre nous allons donner une description des différentes techniques et méthodes qui ont été développées dans le domaine de la préservation de la vie privée dans le data mining. Nous allons donner un aperçu des principaux algorithmes disponibles pour chaque méthode, et les variations sur les différentes techniques tels que *k-anonymat*, *l-diversité*, *randomisation*, le déclassement ou encore la préservation distribuée.

II. Techniques de préservation de la vie privée dans le data mining (PPDM) :

Récemment, de nombreux travaux se sont intéressés à la définition d'algorithmes de fouille de données préservant la vie privée [48, 59].

Ainsi, des approches adaptées à la classification, à la recherche de règles d'association [34] et au clustering [25] permettent d'extraire de la connaissance tout en respectant la contrainte de la vie privée. Par exemple, [48] proposent une nouvelle approche de classification en utilisant un protocole de transfert sécurisé, inspiré des SMC (Secure Multi-Party Computation), entre les différentes bases de données. Vaidya et Clifton [60] considèrent une architecture sécurisée composée de deux parties effectuant les opérations avec les différentes bases afin de rechercher les règles d'association. Enfin, une nouvelle architecture garante de la sécurité et particulièrement adaptée aux problèmes de fouille de données à été proposée dans [55].

A notre connaissance, seuls les travaux de Zhan [26] ont abordé la problématique de la recherche de motifs séquentiels tout en préservant la vie privée. Dans leur approche, les données de chaque base sont transformées. Un protocole sécurisé est alors exécuté pour rechercher les séquences. Théoriquement cette approche est valide et robuste. Cependant, elle souffre de deux lacunes liées aux contraintes sur les données.

Tout d'abord les items ne peuvent appartenir qu'à une seule base. Ainsi, en considérant notre exemple précédent et en ne prenant pas en compte la possibilité d'avoir des items partagés entre les différentes parties, nous n'obtenons pas le résultat final.

Un item comme (1), qui n'est pas supporté par suffisamment de clients dans une base individuelle n'apparaîtra pas dans les résultats finaux. Cette contrainte est également une sérieuse limitation pour les applications réelles dans lesquelles le partage d'items est un impératif (e.g. entreprises sur plusieurs sites, e-commerce). Le second problème est lié à la représentation des données séquentielles : le même client achetant le même item plus d'une fois dans la même base mais avec un TID différent n'est pas autorisé.

Cette dernière contrainte réduit considérablement le type de séquence recherché.

Le domaine de la préservation de la vie privée dans le data mining n'a pas cessé de grandir durant la dernière décennie. Plusieurs techniques ont été développées pour répondre aux préoccupations de l'intimité dans l'exploration de données, tout en tentant de préserver l'utilité des données.

L'objectif de telles techniques est de modifier les données mises à disposition pour l'extraction externe de données pour que l'intimité soit protégée tout en occasionnant le moins d'impacts possibles sur l'utilité des données.

- Cacher les informations sensibles

❖ Retrait des noms, des adresses, des numéros d'identification (comme les numéros de sécurité sociale) à partir de données extraites [30].
❖ En supprimant simplement les identifiants (nom, adresse, SSN) ne suffit pas à protéger la vie privée des personnes concernées. 87% des Américains peuvent être identifiés aussi par leur anniversaire, le sexe et le code postal. [31]

- Discrétisation des valeurs [32]

❖ Le partitionnement des données dans un ensemble de disjoints, les classes exhaustives.
❖ Laisser la donnée ouverte afin qu'elle soit liée à d'autres ensembles de données.

- Déformation / perturbation des valeurs [32]

❖ Ajout d'une valeur aléatoire aux données dans l'ensemble des données privées, soit pour masquer les valeurs de données sensibles ou bien pour masquer des règles sensibles.
❖ Uniforme (sur toute la plage [-a, a] avec une moyenne de 0) vs Distribution gaussienne (distribution normale centrée sur 0).
❖ Globale (reconstruction pour toutes les classes) vs sous-classe reconstruction de distribution (pas de valeurs initiales).
❖ Distribution gaussienne avec reconstruction de sous-classes s'exécute le mieux.

- Valeur dissociation [30]

❖ Permutation des valeurs pour la même colonne pour différentes lignes.

- Calcul multipartite sécurisé

Exemple de calcul d'une somme en utilisant le calcul distribué :

✓ Exige au moins trois partis
✓ Chaque partie peut connaître sa propre valeur ainsi que le résultat
✓ Première partie génère un nombre aléatoire et l'ajoute à leur valeur secrète
✓ Chaque partie subséquente transmet la somme, en ajoutant leur propre valeur secrète ensuite de la transmettre.
✓ Lorsque le résultat est renvoyé à la partie d'origine, ils peuvent soustraire le nombre aléatoire (qu'ils sont les seuls à connaitre) et partager les résultats de la somme avec l'ensemble du groupe.

1. La méthode de randomisation :

1.1. Définition :

La méthode de randomisation est une technique permettant la préservation de la vie privée, dans laquelle le bruit est ajouté aux données afin de masquer les valeurs d'attributs des documents [33, 34].

Le bruit ajouté est suffisamment grand pour que les valeurs d'enregistrement individuelles ne puissent pas être récupérées. Par conséquent, les techniques sont conçues pour tirer des distributions totales à partir des enregistrements perturbés.

1.2. Description détaillée :

Dans cette partie, nous allons discuter de la méthode de randomisation de la préservation de la vie privée dans l'exploration de données (PPDM). La méthode de randomisation a été traditionnellement utilisée dans le contexte de la distorsion des données par la distribution de probabilité pour des méthodes telles que les enquêtes qui ont un biais de réponse évasive en raison de préoccupations en matière de vie privée [35, 36]. Cette technique a également été étendue au problème de PPDM [33].

La méthode de randomisation peut être décrite comme suit. Considérons un ensemble d'enregistrements de données dénoté par $X = \{x1 \ldots xN\}$.

Pour chaque $xi \in X$, nous ajoutons une composante de bruit qui est tiré de la distribution de probabilité $fY(y)$.

Ces composantes de bruit sont tirés indépendamment, et sont notés Y1. . . yN.

Ainsi, le nouveau jeu d'enregistrements distordus sont désignés par x1 + y1. . . xN + yN.

On note cette nouvelle série d'enregistrements par *Z1. . . Zn*. En général, on suppose que la variance du bruit ajouté est assez grande, de sorte que les valeurs de l'enregistrement d'origine ne peuvent pas être facilement devinées à partir des données déformées. Ainsi, les documents originaux ne peuvent pas être récupérés, mais la distribution des documents originaux peuvent être récupérée.

Ainsi, si *X* la variable aléatoire indiquant la distribution des données pour l'enregistrement d'origine, Y la variable aléatoire décrivant la distribution du bruit, et Z la variable aléatoire désignant l'enregistrement final, nous avons:

$$Z = X + Y$$
$$X = Z - Y$$

Maintenant, nous notons que N instanciations de la distribution de probabilité Z sont connus, tandis que la distribution Y est connue publiquement. Pour un assez grand nombre de valeurs de N, la distribution Z peut être approchée de près par l'aide de diverses méthodes telles que l'estimation de la densité du noyau. En soustrayant Y de la distribution approximative de Z, il est possible d'approcher la distribution de probabilité initiale X.

En pratique, on peut combiner le processus de rapprochement des Z avec soustraction de la distribution de Y de Z en utilisant une variété de méthodes itératives telles que ceux discutés dans [33, 34]. Ces méthodes itératives ont généralement une plus grande précision que la solution séquentielle de première approximation Z, puis en soustrayant Y de lui.

Nous notons qu'à la fin du processus, nous avons seulement une distribution contenant le comportement de X. Les enregistrements individuels ne sont pas disponibles. En outre, les distributions ne sont disponibles que sur les dimensions individuelles. Par conséquent, de nouveaux algorithmes de data mining doivent être conçus pour fonctionner avec les distributions uni à variables plutôt que les enregistrements individuels.

Cela peut parfois être un défi, étant donné que de nombreux algorithmes de data mining sont intrinsèquement dépendante des statistiques qui ne peut être prélevé soit depuis les dossiers individuels ou bien les distributions de probabilité à plusieurs variables associées aux enregistrements.

Alors que l'approche peut certainement être étendue à des distributions multi variées, l'estimation de densité devient intrinsèquement plus difficile [37] avec l'augmentation dimensionnalités. Car même dimensionnalités modestes tels que de 7 à 10, le processus d'estimation de la densité devient de plus en plus imprécis, et tombe en proie à la malédiction de la dimensionnalité.

Un avantage clé de la méthode de randomisation est qu'il est relativement simple, et ne nécessite pas de connaitre la distribution d'autres enregistrements de données. Ce n'est pas le cas d'autres méthodes telles que le k-anonymat qui exigent la connaissance d'autres enregistrements dans les données.

Par conséquent, la méthode de randomisation peut être mise en œuvre au moment de la collecte des données, et ne nécessite pas l'utilisation d'un serveur sécurisé contenant tous les documents originaux afin d'effectuer le processus d'anonymisation.

Bien que ce soit un atout de la méthode de randomisation, il conduit aussi à certaines faiblesses, car il traite tous les enregistrements également, indépendamment de leur densité locale. Par conséquent, les enregistrements aberrants sont plus sensibles aux attaques contradictoires par rapport aux dossiers dans les régions plus denses dans les données [10].

Afin d'éviter cela, on peut avoir besoin d'être inutilement plus agressif dans l'ajout de bruit pour tous les enregistrements de données. Cela réduit l'utilité des données à des fins d'exploitation minière.

2. Anonymisation basée groupe :

La méthode de randomisation est une technique simple qui peut être facilement implémentée au moment de la collecte des données, parce que le bruit ajouté à un enregistrement donné est indépendant du comportement des autres enregistrements de données. C'est aussi une faiblesse car des enregistrements aberrants peuvent souvent être difficiles à masquer.

De toute évidence, dans les cas où l'intimité de conservation n'a pas besoin d'être effectuée au moment de la collecte de données, il est souhaitable de disposer d'une technique dans laquelle le niveau d'imprécision dépend sur le comportement de la localisation de cet enregistrement donné. Une autre faiblesse majeure du cadre de randomisation est qu'il ne tient pas compte de la possibilité que enregistrements accessibles au public peuvent être utilisés pour identifier l'identité des propriétaires de ce dossier. Dans [30], il a été démontré que l'utilisation d'enregistrements accessibles au public peut conduire la vie privée à devenir fortement compromise dans les cas de grande dimension.

Cela est particulièrement vrai des enregistrements aberrants qui peuvent être facilement distingués des autres dossiers dans leur localité. Par conséquent, une approche large de la transformation de l'intimité est de construire des groupes d'enregistrements anonymes qui sont transformerons d'une manière spécifique au groupe.

2.1. Le k-anonymat :

2.1.1 Définition :

Le modèle k-anonymat a été développé en raison de la possibilité d'une identification indirecte des enregistrements de bases de données publiques. C'est parce que les combinaisons d'attributs d'enregistrement peuvent être utilisés pour identifier exactement les dossiers individuels.

Dans la méthode k-anonymat, nous réduisons la granularité de la représentation des données avec l'utilisation de techniques telles que la généralisation et la répression. Cette granularité est suffisamment réduite que les enregistrements tracent des données sur au moins k autres enregistrements de données.

2.1.2 Description détaillée :

Attribut sensible	Un attribut qui ne doit pas être associé publiquement avec un individu. [38]
Quasi-identificateur	Un ensemble d'attributs non sensibles quand ils son pris ensemble peuvent être utilisés pour identifier un individu. L'ensemble des attributs peut être entièrement contenu dans une table ou peut être réparti entre plusieurs tables liées. [38] [39] [40]
Classe d'équivalence	L'ensemble des enregistrements qui ont les mêmes valeurs pour tous les quasi-identificateurs. [38]
k-anonymat	Un tableau est dit avoir k-anonymat si, pour chaque classe d'équivalence, il ya au moins k tuples dans la classe. [38] [39] [40]

Dans de nombreuses applications, les enregistrements de données sont mis à disposition par la simple élimination des clés d'identification telle que le nom et les numéros de sécurité sociale des enregistrements personnels. Toutefois, d'autres types d'attributs (connu sous le nom de pseudo-identification) peuvent être utilisés afin d'identifier avec précision les enregistrements. Par exemple, des attributs tels que l'âge, le code postal et le sexe sont disponibles dans les documents publics tels que les rouleaux de recensement.

Lorsque ces attributs sont également disponibles dans un ensemble de données, ils peuvent être utilisés à déduire l'identité de l'individu correspondant. Une combinaison de ceux-ci attributs peut être très puissante, car ils peuvent être utilisés pour réduire les possibilités à un petit nombre d'individus.

Dans techniques k-anonymat [41], nous réduisons la granularité de la représentation de ces pseudo-identificateurs avec l'utilisation de techniques telles que la généralisation et la suppression. Dans le procédé de généralisation, les valeurs de l'attribut sont généralisées à une plage afin de réduire la granularité de la représentation.

Par exemple, la date de naissance pourrait être généralisée à toute une gamme comme l'année de naissance, de manière à réduire le risque d'identification. Dans le procédé de suppression, la valeur de l'attribut est supprimée complètement. Il est clair que de telles méthodes réduisent le risque de l'identification avec l'utilisation de documents publics, tout en réduisant l'exactitude des applications sur les données transformées.

Afin de réduire le risque d'identification, l'approche k-anonymat nécessite que chaque tuple dans le tableau soit indiscernable et liée à pas moins de k répondants. Cela peut être formalisé comme suit:

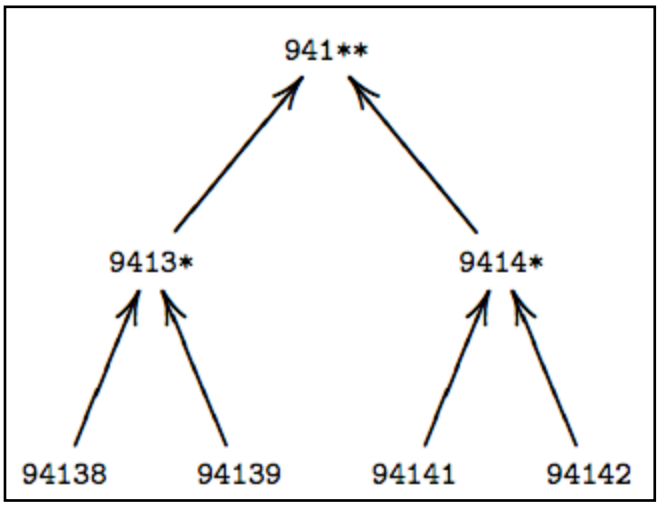
Chaque version des données doit être telle que toutes les combinaisons des valeurs de quasi-identificateurs peuvent être indistinctement correspondantes à au moins k répondants.

2.1.3 Méthode :

- En utilisant la généralisation de domaine et la suppression de tuple, une table peut être faite k-anonyme.
- Mettre en domaine des classes distinctes dans la généralisation des partitions de données, comme cela se fait dans la discrétisation des valeurs.
- Suppression de Tuple supprime les valeurs individuelles ou des enregistrements.
- Algorithmes multiples pour calculer k-anonymat: Samarti'S, Bayardo-Agrawal, incognito, Mondrian: [42]

• Renforcer la représentation des généralisations

Exemple de hiérarchie de généralisation d'un ensemble de codes postaux :



• Trouver des généralisations qui répriment la moindre quantité de données tout en conservant k-anonymat.

2.1.4 Les principaux problèmes :

- Attaque d'homogénéité: le manque de diversité parmi les attributs sensibles pour une classe d'équivalence donnée [39] [40]
- Les attaques des connaissances de base: la connaissance externe sur une personne, séparée de l'ensemble des données, permet à un individu d'être identifié au sein d'une classe d'équivalence [38]
- La k-anonymisation optimale est NP-difficile pour k> 2 [43]

Nous notons que le problème de l'anonymisation optimal est intrinsèquement difficile. Dans [89], il a été montré que le problème de k- anonymisation optimale est NP-difficile. Néanmoins, le problème peut être résolu de façon très efficace par l'utilisation d'un certain

nombre de méthodes heuristiques. Une méthode proposée par Bayardo et Agrawal [] est l'algorithme k-Optimize qui peut souvent obtenir des solutions efficaces.

L'approche suppose un ordre parmi les attributs quasi-identificateurs. Les valeurs des attributs sont discrétisés en intervalles (attributs quantitatifs) ou regroupés en différents ensembles de valeurs (attributs catégoriques). Chacun de ces groupements est un élément. Pour un attribut donné, les éléments correspondants sont également commandés.

Un index est créé en utilisant ces paires attribut-intervalle (ou articles) et un arbre d'énumération définie est construit sur ces paires attribut-intervalles. Cet arbre d'énumération ensemble est une énumération systématique de toutes les généralisations possibles avec l'utilisation de ces groupements. La racine du nœud est le nœud nul, et tous les niveaux successifs de l'arbre sont construits en ajoutant un élément qui est lexicographiquement plus grand que tous les éléments à ce nœud de l'arbre.

Nous constatons que le nombre de nœuds possibles dans l'arbre augmente de façon exponentielle avec la dimensionnalité des données. Par conséquent, il n'est pas possible de construire l'arbre entier même pour des valeurs modestes de n. Cependant, l'algorithme k-Optimize peut utiliser un certain nombre de stratégies d'élagage à bon escient. En particulier, un nœud de l'arbre peut être taillé lorsqu'il est prouvé qu'aucun descendant de celui-ci ne peut être optimal.

Cela peut être fait en calculant un bond sur la qualité de tous les descendants de ce nœud, et en le comparant à la qualité de la meilleure solution courant obtenu pendant le processus de traversée. Une branche et technique liée peuvent être utilisées pour améliorer successivement la qualité de la solution au cours du processus de traversée. Éventuellement, il est possible de mettre fin à l'algorithme à un temps de calcul maximale, et utiliser la solution actuelle à ce point, qui est souvent très bon, mais peut-être pas optimale.

Nous constatons que la généralisation et la suppression ne sont pas les seules techniques de transformation pour la mise en œuvre de k-anonymat. Par exemple, dans [45], il est expliqué comment utiliser la micro-agrégation dans laquelle des groupes d'enregistrements sont construits. Pour chaque groupe, la valeur représentative est la valeur moyenne le long de chaque dimension du groupe. Une méthode similaire pour atteindre l'anonymat via le clustering est proposée dans [39]. Le travail dans [39] fournit également des algorithmes d'approximation de facteur constant de concevoir le clustering.

Algorithmes	Pratiques ?	Garantie
Sweeny-Datafly	Oui	Aucune
Sweeny-MinGen	Non	Optimal
Samarati-AllMin	Non	Optimal
Iyengar-GA	Oui	Aucune
Winkler-Anneal	Possible	Aucune
Meyerson-approx	Possible, mais seulement pour K utilisant la suppression seul	O(k logk) Optimal

Table : Quelques approches connues de k-anonymat

2.2. Le l-diversité :

2.2.1. Définition :

Le modèle L-diversité a été conçu pour traiter certaines faiblesses du modèle k-anonymat puisque la protection des identités au niveau de k-personnes n'est pas la même que la protection des valeurs sensibles correspondantes, surtout quand il ya homogénéité des valeurs sensibles au sein d'un groupe. Pour ce faire, le concept de diversité intra-groupe de valeurs sensibles est promu au sein du système d'anonymisation [46].

2.2.2. Description détaillée :

l-diversité	Une table a l-diversité s'il y a au moins l « bien représentées » valeurs pour les attributs sensibles pour chaque class d'équivalence. [38][40]
La diversité distincte (moins conservatrice)	Au moins *l* valeurs distinctes d'attributs sensibles pour chaque classe d'équivalence.
La diversité entropie(plus conservatrice)	entropie (E) = - Σ p (E, s) log p (E, s) E est une classe d'équivalence p (E, s) est le pourcentage des tuples dans E Une table satisfait la l-diversité entropie si pour chaque E, entropie (e) ≥ log l
La diversité récursive (c, l)	r_i est le nombre de fois où la $i^{ème}$ plus fréquente m valeurs de l'attribut sensible pour E E est récursif (c, l) si : $r_1 < C (r_1 + r_{1+1} + ... + r_m)$ Une table satisfait la diversité récursive(c, l) si pour chaque E, $r_1 < C (r_1 + r_{1+1} + ... + r_m)$ (r-sub-one est inférieure à un nombre constant de fois la somme du nombre de fois que le $l^{ième}$ plus fréquente valeur sensible apparaît ainsi que toutes les valeurs sensibles moins fréquentes) Les classes les moins fréquentes ne devraient pas apparaître trop rarement; classes les plus fréquentes ne devraient pas apparaître trop souvent.

La k-anonymat est une technique intéressante en raison de la simplicité de la définition et les nombreux algorithmes disponibles pour effectuer l'anonymisation.

Néanmoins, la technique est sensible à de nombreux types d'attaques surtout quand connaissances de base est disponible à l'attaquant. Certains types de ces attaques sont les suivantes:

- *Homogénéité Attaque:* Dans cette attaque, toutes les valeurs d'un attribut sensible au sein d'un groupe de k enregistrements sont les mêmes. Par conséquent, même si les données sont k-anonyme, la valeur de l'attribut sensible pour ce groupe de k enregistrements peut être prédit avec précision.
- *Contexte attaque de connaissances:* Dans cette attaque, l'adversaire peut utiliser une association entre un ou plusieurs attributs quasi-identifiant de l'attribut sensible afin d'affiner les valeurs possibles du champ sensible encore.

De toute évidence, tandis que k-anonymat est efficace dans la prévention de l'identification d'un enregistrement, il peut ne pas toujours être efficace dans la prévention de l'inférence des valeurs sensibles des attributs de cet enregistrement. Par conséquent, la technique de l-diversité a été proposée, qui non seulement maintient la taille minimale d'un groupe de k, mais se concentre également sur le maintien de la diversité des attributs sensibles. Par conséquent, le modèle L-diversité [46] de la vie privée est définie comme suit:

Soit un q*bloque est un ensemble de tuples tel que c'est une insensible généralisation de valeurs pour q*. Un bloque q* est l-diverse si il contient l valeurs « bien représentée» pour l'attribut sensible S. Une table est l-diverse, si chaque bloque q* à l'intérieur est l-diverse.

2.2.3. Méthode :

l-diversité s'appuie sur k-anonymat en ajoutant la condition supplémentaire que les tuples de la même classe d'équivalence ne devraient pas avoir les mêmes attributs sensibles.

- Limitations primaires [38]:

Difficulté: Prenons l'exemple du VIH statut :

99,997% de la population américaine est séronégatif. Une valeur négative pour l'attribut sensible du statut VIH n'est pas considérée comme privée, car une valeur négative peut être mise pour la population générale avec une très grande probabilité. Compte tenu de 100.000 enregistrements avec seulement 300 ayant une valeur positive pour le VIH, il pourrait y avoir au plus 300 classes d'équivalence.

Attaque Asymétrique: comme dans l'exemple ci-dessus de la séropositivité, il est supposé qu'un individu de la population totale a une chance de 0,003% d'être séropositif, mais si cette personne appartient à une classe d'équivalence de 50% entre les positifs et des valeurs

négatives, on peut supposer que l'individu a une probabilité plus élevée d'avoir la valeur positive que la population générale.

Similarité: les valeurs des attributs sensibles peuvent être traités différents, mais peut-être sémantiquement similaire.

2.3. Le modèle T-proximité :

2.3.1. Définition :

Le modèle T-proximité est une amélioration supplémentaire sur le concept de l-diversité. Une caractéristique du modèle l-diversité est qu'elle traite toutes les valeurs d'un attribut donné de la même manière quelle que soit sa distribution dans les données. C'est rarement le cas pour les ensembles de données réels, puisque les valeurs d'attributs peuvent être très biaisées. Cela peut rendre plus difficile de créer des représentations l-diverses possibles. Souvent, un adversaire peut utiliser les connaissances de base de la répartition mondiale afin de tirer des conclusions sur les valeurs sensibles dans les données. En outre, toutes les valeurs d'un attribut sont tout aussi sensibles.

2.3.2. Description détaillée :

t-proximité	Une classe d'équivalence a t-proximité si la distance entre la distribution d'un attribut sensible dans la classe et de la distribution de l'attribut à travers l'ensemble du tableau n'est pas supérieure à t. Une table a t-proximité si chaque classe d'équivalence a t-proximité. [38]

Par exemple, un attribut correspondant à une maladie peut être plus sensible lorsque la valeur est positive, plutôt que quand elle est négative. Dans [45], un modèle t-proximité a été proposé, qui utilise la propriété que la distance entre la distribution de l'attribut sensible rendu anonymes au sein d'un groupe ne doit pas être différente de la distribution globale de plus d'un seuil t. En outre, l'approche t-proximité tend à être plus efficace que de nombreux autres méthodes de préservation de la vie privée dans le data mining pour le cas des attributs numériques.

3. Préservation distribuée de la vie privée :

Dans de nombreux cas, les entités concernées peuvent souhaiter obtenir des résultats agrégés à partir d'ensembles de données qui sont cloisonnées entre ces entités. Un tel cloisonnement peut être horizontale (lorsque les dossiers sont répartis entre plusieurs entités) ou verticale (lorsque les attributs sont réparties sur plusieurs entités). Alors que les entités individuelles ne veulent pas partager leurs ensembles de données entiers, ils peuvent consentir à l'échange d'informations limité à l'utilisation d'une variété de protocoles. L'effet global de ces

méthodes est de maintenir la confidentialité de chaque entité individuelle, tout en tirant les résultats globaux sur l'ensemble des données.

Le but essentiel dans la plupart des méthodes distribués pour la PPDM est de permettre le calcul de statistiques globales utiles sur l'intégralité des données sans compromettre la confidentialité des données individuelles définit dans les différents participants. Ainsi, les participants peuvent souhaiter collaborer à l'obtention de résultats globaux, mais ne peuvent pas faire entièrement confiance à l'autre en fonction de la distribution de leurs propres ensembles de données. A cet effet, les ensembles de données peuvent être soit horizontalement partitionnées ou être partitionnées verticalement.

Dans les ensembles de données partitionnées horizontalement, les enregistrements individuels sont répartis à travers de multiples entités, dont chacun possède le même ensemble d'attributs. Dans partitionnement vertical, les entités individuelles peuvent avoir des attributs différents (ou sur) d'un même ensemble d'enregistrements. Les deux types de partitionnement posent des défis différents au problème de la préservation distribuée de la vie privée dans le data mining.

Le problème de la préservation distribuée de la vie privée dans le data mining c'est qu'elle est étroitement chevauchée avec un champ de cryptographie pour déterminer les calculs multipartites sécurisés.

Un large aperçu de l'intersection entre les domaines de la cryptographie et la préservation distribuée de la vie privée peut être trouvé dans [46]. L'approche globale de méthodes cryptographiques a tendance à calculer des fonctions sur les intrants (inputs) fournis par des destinataires multiples sans partager réellement les intrants (inputs) entre elles. Par exemple, dans un cadre 2-partie, Alice et Bob peuvent avoir deux entrées x et y respectivement, et pourraient à la fois calculer la fonction f (x, y) sans révéler X ou Y à l'autre.

Ce problème peut également être généralisé à travers k partis en concevant le k argument de la fonction h (x1 ... xk). De nombreux algorithmes de data mining peuvent être considérés dans le contexte des calculs répétitifs de plusieurs de ces fonctions primitives comme le produit de graduation scalaire, somme sécurisée…etc. Afin de calculer la fonction f (x, y) ou h (x1 ..., xk), un protocole devra être conçu pour l'échange d'informations de telle manière que la fonction est calculée sans compromettre la confidentialité. Nous notons que la robustesse du protocole dépend du niveau de confiance que l'on est prêt à placer sur les deux participants Alice et Bob. C'est parce que le protocole peut être soumis à différents types de comportement contradictoire:

- Adversaires semi-honnêtes: Dans ce cas, les participants Alice et Bob sont curieux et tentent d'apprendre à partir de l'information reçue par eux durant le protocole, mais ne s'écartent pas du protocole lui-même. Dans de nombreuses situations, cela peut être considéré comme un modèle réaliste du comportement contradictoire.

- Les adversaires malveillants: Dans ce cas, Alice et Bob peuvent varier du protocole, et peuvent envoyer des entrées sophistiqués les uns aux autres pour apprendre à partir de l'information reçue de l'autre.

4. L'efficacité de l'application de déclassement :

Dans de nombreux cas, même si les données ne sont pas disponibles, la production des applications telles que la règle d'association minière, la classification ou le traitement des requêtes peut entraîner des violations de la vie privée. Ceci a conduit à des recherches dans la rétrogradation de l'efficacité des applications de données ou des modifications de l'application. Quelques exemples de ces techniques comprennent règle d'association se dissimulant [47], classificateur de déclassement [48], et l'audit requête [49].

Une importante application sensible de respect de la confidentialité est celui de la classification, dans lesquels les résultats d'une application de classification peuvent être des informations sensibles pour le propriétaire d'un ensemble de données. Par conséquent, la question est de modifier les données de telle sorte que la précision du processus de classification est réduite, tout en conservant l'utilité des données pour d'autres types d'applications. Un certain nombre de techniques ont été discutées dans [50, 48] dans la réduction de l'efficacité du classificateur dans le contexte de la règle de classification et les applications d'arbre de décision. La notion de déclassement parcimonieux est proposée [50] dans le contexte de blocage des canaux d'inférence pour des fins de classification pendant l'extraction de l'effet de l'utilité générale. Un système appelé Rational Downgrader [48] à été conçu avec l'utilisation de ces principes.

Les méthodes de dissimulation des règles d'association peuvent également être généralisées à des classificateurs basés sur les règles. C'est parce que les classificateurs à base de règles ont souvent recours à des méthodes d'extraction de règles d'association en tant que sous-programmes, de sorte que les règles avec les étiquettes de classe dans leur conséquente sont utilisées à des fins de classification. Pour une approche de déclassement de classification, ces règles sont des règles sensibles, alors que toutes les autres règles (avec des attributs non-classe dans la conséquente) sont des règles non sensibles. Un exemple de méthode pour des règles de déclassement à base de classificateur est discuté dans [51], dans lequel il a été montré comment cacher efficacement les règles de classification pour un ensemble de données.

5. Inférence à partir de résultats:

Les techniques dont on va parler assurent une mesure de protection pour les données qui servent de support à la dérivation de règles. Les règles, par définition, offrent de la connaissance. Le problème est que cette connaissance peut en soi poser un risque à la confidentialité.

Par exemple, supposons que nous réalisons une étude médicale pour déterminer des facteurs de risque pour une maladie à un stade terminal. Les méthodes décrites s'assurent que des individus ayant la maladie ne sont pas révélés. Cependant, une règle dont la confiance est élevée, combinée avec une connaissance externe concernant ces individus, peut permettre l'inférence du statut de la maladie d'un individu. Spécifiquement, si les antécédents de la règle ne constituent pas des informations privées, une règle avec une confiance élevée risque de contenir en soi une information comme quoi les individus qui répondent au critère de la règle sont susceptibles d'avoir la maladie.

Aborder ce problème représente un challenge en soi, car les résultats que nous désirons (règles d'association) est la source de violation de la confidentialité. Donner une quantification à cette probabilité de la divulgation de données privées est franche ; la confiance donne une distribution de probabilité sur la valeur de la donnée privée des individus qui ont le même antécédent, Le support indique quelle fraction d'individus possède cet antécédent. Ceci permet de faire une évaluation du risque de confidentialité, concernant cet individu. Plus difficile encore, décider si le risque supposé au départ est acceptable ou pas, tenant compte de la valeur de la connaissance portée par la règle.

Des recherches sont requises pour la construction d'un framework global pour évaluer la valeur de cette confidentialité. Si le risque estimé est supérieur aux bénéfices, est-ce que ceci élimine notre capacité à faire de la fouille de données ?

[Atallah M. et al. 1999] pose une alternative intéressante : modifier les données pour diminuer le support et éventuellement la confiance d'une règle spécifique. Le but, cependant, est d'enlever des règles spécifiques de l'ensemble de sortie standard (règles habituellement générées) plutôt que de protéger les données en entrée. Le problème minimiser les changements apportées aux données en même temps que s'assurer que la confiance et le support des règles en question sont abaissés au-dessous d'un certain seuil s'avère être un problème NP-complet. Cependant, ces approches à elles seules ne permettent pas de donner une solution fiable au problème de la confidentialité des données. Ce n'est qu'en les combinant avec d'autres techniques comme la perturbation des données qu'on peut arriver à une solution plus complète à la fois du problème de la confidentialité des données et celui des règles générées à partir de ces données. Cette solution fait partie des efforts pour garantir la confidentialité dans la recherche de règles d'association préservant la confidentialité (Privacy Preserving Association Rule Mining).

C'est un problème ouvert qui nécessite probablement des avancées significatives en matière de définitions de la confidentialité et des métriques qui lui sont associées.

III. Conclusion

Pour résumer, l'objectif des algorithmes de PPDM est d'extraire des modèles valides tout en préservant la vie privée. Naturellement un algorithme sera efficace s'il assure à la fois une bonne protection de la vie privée et une bonne extraction de la connaissance. Du fait que ces algorithmes font l'objet de distribution dans le cas de grilles de calcul, le coût de communication encouru pendant l'échange d'informations parmi un certain nombre de sites collaborant devrait également être pris en compte.

On a vu que le calcul distribué sécuritaire multi- parties est la principale proposition pour résoudre le problème de la protection des données. Certaines approches de calcul distribué sécuritaire multi-parties utilisent une tierce partie. Ces approches réduisent en général le coût de communication mais sont considérées comme non sécurisées.

D'autres approches n'utilisent pas de tiers serveur. Ces dernières sont plus sécurisées mais ont un coût de communication élevé. Un aspect important dans le développement et l'évaluation des algorithmes de PPDM est l'identification des critères d'évaluation appropriés et le développement de repères relatifs. Aucun des algorithmes existants n'est plus performent que tous les autres sur tous les critères possibles. Plutôt, un algorithme peut être plus optimal qu'un autre sur certains critères spécifiques tels que la complexité et/ou la perte d'informations.

Une liste préliminaire de paramètres pouvant être utilisés pour évaluer les algorithmes de PPDM est donnée ci-dessous :

- La complexité des algorithmes proposés en termes de temps requis : c'est le temps requis par chaque algorithme pour le transfert ou la transformation d'un ensemble spécifique d'informations sensibles. Une première approche dans l'évaluation du temps requis par un algorithme préservant la vie privée est d'évaluer le coût de calcul. Dans ce cas-ci, il est évident qu'un algorithme ayant une complexité polynômiale est plus performant qu'un autre avec une complexité exponentielle. Une approche alternative serait d'évaluer le temps nécessaire en termes de nombre moyen d'exécutions requis pour réduire la fréquence d'apparition d'informations sensibles spécifiques en-dessous d'un seuil indiqué. Ces valeurs, ne fournissent pas une mesure absolue mais peuvent être considérées afin de faire une comparaison parmi les différents algorithmes.

- La perte d'informations à la fin du processus de préservation de vie privée est une question importante du fait que, pour que l'information sensible soit cachée, la base de données doit dans certains cas être modifiée par l'insertion de fausses informations ou à travers le blocage de certaines valeurs des données. Il est évident que, plus des changements sont faits sur la base originale, moins la base de données reflètera un domaine d'intérêt. Par conséquent, un paramètre

d'évaluation pour mesurer la perte d'informations devrait être la quantité d'informations qui est détruite après l'application du processus de préservation de vie privée.

- Le niveau d'incertitude avec lequel les informations sensibles qui ont été cachées peuvent encore être prédites. En effet les stratégies de préservation de la vie privée fonctionnent en diminuant l'information que nous voulons protéger en dessous d'un certain seuil. Certes, l'information cachée peut encore être déduite bien qu'avec un certain niveau d'incertitude.

- La résistance présentée par les algorithmes de préservation de vie privée face aux différentes techniques de datamining. L'objectif étant de protéger les informations sensibles contre la révélation non autorisée. Dans ce cas-ci, il est important de ne pas oublier que des intrus essayeront de compromettre l'information en utilisant divers algorithmes de datamining.

Ainsi un algorithme de PPDM développé contre une technique particulière de datamining et assurant la préservation de vie privée peut ne pas parvenir à une protection similaire contre tous les algorithmes de datamining possibles.

La Figure suivante illustre ces critères :

	Distorsion (ou perturbation)	**Blocage**	**Reconstruction**
Complexité algorithmique	Beaucoup d'opérations E/S surtout s'il y a plusieurs transactions dans la base de donnée originale		Moins d'E/S que les algorithmes de modification
Perte d'informations	Beaucoup de règles et d'item sets perdus (effet de bords indésirables) à cause des fausses valeurs ou incertitudes introduites dans la BDD		Pas de règles et item sets perdus (contrôle directement les effets de bord)
Protection de la vie privée	Un adversaire infère difficilement les valeurs cachées (mais la BDD reflète moins le domaine d'intérêt)	Moins il y a d'incertitude, plus un adversaire peut inférer les valeurs cachées	Bonne protection de la connaissance

CHAPITRE IV

Etude expérimentale

I. Introduction :

Récemment le problème de la protection de la vie privée est devenu très important dans le domaine du datamining, particulièrement quand les données sont partitionnées verticalement sur plusieurs sites. En datamining, les données sont souvent représentées par des vecteurs d'attributs et, pour plusieurs applications, le produit scalaire est une des opérations qui est utilisée de façon répétée. Son calcul de manière efficace et sécurisée est donc important. Bien que plusieurs solutions existent, le besoin d'avoir une solution plus efficace reste toujours.

Dans le but d'implémenter une solution cryptographique au problème de confidentialité des données dans un environnement de grilles de calcul, on doit au préalable se positionner au niveau des modèles mathématiques et des outils offerts par ce genre de plates-formes.

Dans notre cas, l'algorithme que nous allons présenter, proposé dans [73], en comparaison avec l'algorithme DNSP1 et l'algorithme de VAIDYA est un algorithme qui utilise le crypto système de Paillier qui utilise en l'occurrence la propriété qu'il soit homomorphe additif et que nous allons détailler par la suite. Tout cela en utilisant le Produit scalaire comme outil mathématique puissant pour le calcul d'item sets fréquents et l'implémentation d'algorithmes préservant la confidentialité des données.

II. Implémentation:

L'étude du comportement d'un système peut se faire selon plusieurs approches. La première est la méthode analytique qui se base sur la modélisation mathématique et l'utilisation des démonstrations pour prouver qu'une configuration est plus optimale qu'une autre. Mais avec des systèmes complexes, les modélisations et les démonstrations mathématiques deviennent très difficiles voir impossibles.

Par conséquent, la deuxième méthode qui est l'expérimentation est utilisée. Cette approche consiste à expérimenter directement différentes solutions afin de trouver la meilleure d'entre elles. L'expérimentation peut, à son tour, devenir contraignante si nous voulons expérimenter des systèmes qui nécessitent des moyens importants dont le coût peut être très lourd ou des systèmes qui font intervenir plusieurs paramètres qui font que les conditions expérimentales ne soient pas les mêmes.

Dans ce cas, la troisième méthode qui est la simulation représente un bon outil. La simulation ne nécessite pas de grands moyens, et permet d'itérer autant que nécessaire les expériences tout en ayant les mêmes conditions expérimentales [33].

Il existe de nombreux simulateurs de grille parmi lesquels nous pouvons citer :

- Bricks [42] (Takefusa et al., 2001), pour la simulation de systèmes du type client/serveur,

- OptorSim [23] (Bell et al., 2003), conçu pour l'étude d'algorithmes d'ordonnancement traitant spécifiquement de la réplication ou de la migration de données,

- GridSim [26] (Buyya et al., 2002) et Simgrid [27] (Legrand et al., 2003), des outils de modélisation de ressources de calcul et réseau, et de simulation d'algorithmes d'ordonnancement de grille.

2.1. Gridsim :

Gridsim est une plate-forme de développement de simulateurs, développée à l'université de Monash en Australie depuis 2001. Gridsim est une bibliothèque de fonctions implémentée en java, l'utilisateur utilise ces fonctions pour développer son propre simulateur.

Gridsim permet de définir tous les éléments d'un environnement de calcul : les clients, les éléments de calculs, les liens et les tâches. Il gère aussi les messages qui circulent dans cet environnement et il permet d'accéder aux informations essentielles de la simulation, à savoir, l'état des ressources et la durée d'exécution d'une tâche [33].

Architecture [25]:

1. Utilisateur : Chaque utilisateur diffère du reste des utilisateurs par :

- Types des travaux créés, par exemple le temps d'exécution du travail, nombre de réplication, etc...

- Stratégie d'optimisation, par exemple minimisation du coût, du temps, ou des deux,

- Fuseau horaire.

2. Broker : Chaque utilisateur est relié à un broker. Chaque travail d'un utilisateur est d'abord soumis à son broker. Avant d'ordonnancer les tâches, le broker obtient dynamiquement une liste des ressources disponibles, cette liste est fournie par le service d'information de la grille. Chaque broker essaye d'optimiser la politique de son utilisateur et donc, on s'attend à ce que les brokers fassent concurrencent en accédant aux ressources. Les algorithmes d'ordonnancement employés par les brokers doivent être fortement adaptables à la situation de l'offre et de la demande du marché.

3. Resource : Chaque ressource peut différer du reste des ressources par :

- Coût de traitement,

- Nombre de processeurs,

- Vitesse de traitement,

- Fuseau horaire.

La vitesse des ressources et le temps d'exécution du travail sont exprimés en MIPS (Million d'instructions par seconde).

Les données associées à chaque tâche sont :

- Taille exprimée en MI (Million d'instructions),

- Temps d'exécution cumulé,

- Temps d'attente cumulé,

- Date d'arrivée,

- Date début d'exécution.

4. Service d'information de grille : Fournit des services d'enregistrement des ressources et maintient une liste des ressources disponibles dans la grille. Ce service est utilisé par les brokers pour obtenir la configuration et le statut des ressources.

5. Entrée et sortie : Le flux d'information entre les entités de GridSim circule via les entités d'Entrée et de Sortie.

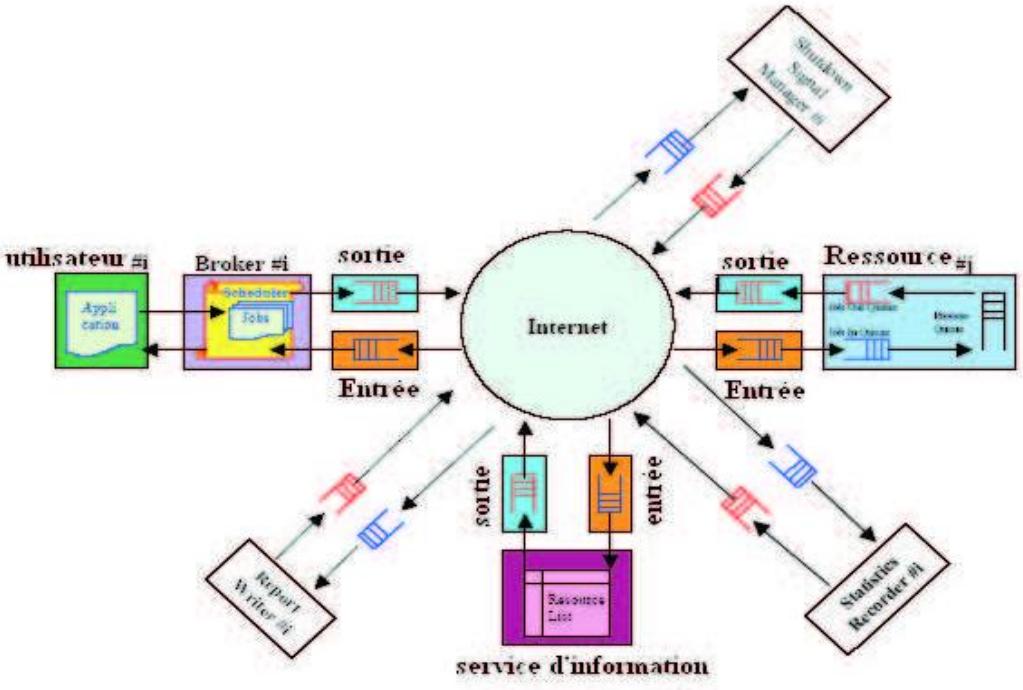


Figure 1 : Structure du système GridSim

2.2. Environnement d'expérimentation :

L'environnement dans lequel les différentes expérimentations ont été réalisées est défini par les éléments suivants :

1. Environnement matériel et logiciel : Nous avons développé notre stratégie sur une machine HP Pavillon g6 fonctionnant sous le système Windows 7, avec les caractéristiques suivantes : Intel inside i5, 2.4 GHZ avec 6Go de RAM,

2. Environnement de développement : Le langage utilisé est le Java.

2.3. Notions de la cryptographie :

2.3.1. Types de chiffrement :

Définition 1: (Cryptosystème symétrique). Un système cryptographique est dit symétrique (à clé secrète) si les correspondants utilisent la même clé pour le chiffrement (resp. déchiffrement) de leurs messages.

Définition 2: (Cryptosystème asymétrique). Un système cryptographique est dit asymétrique (à clé publique) si la clé de chiffrement du message est connue par tous mais seul le destinataire du message connaît la clé de déchiffrement.

2.3.2. Sécurité :

Définition 3 : (Sécurité sémantique): On dit d'un Cryptosystème qu'il est sécurisé sémantiquement si à partir d'un chiffré (une donnée élémentaire), un attaquant ne peut pas, rapidement ou de façon aisée, obtenir des informations sur le message intercepté.

Définition 4 (indistingabilité - Computationally indistinguishable): Soit m deux messages. Soit M1; M2 leurs chiffrés respectifs. Si l'attaquant choisit M, il ne peut pas reconnaître rapidement si c'est un chiffré de m1 ou de m2.

2.4. Interprétation :

Dans notre implémentation, nous allons nous baser sur ce que permet un outil mathématique trés puissant qu'est le produit scalaire. Donc combiner les techniques du data mining et l'échange de données dans une grille de calcul en permettant de garantir la confidentialité des données qui y transitent.

2.5. DHD (Data Hiding in Database) :

2.5.1. Définition :

Ces dernières années, l'utilisation croissante des systèmes multi-bases a entraînée le développement d'un grand nombre de bases de données transactionnelles distribuées.

Dans le but de pouvoir exploiter au maximum ces sources de données, les grandes firmes possédant ces données espèrent extraire de la connaissance à partir de l'ensemble de ces bases de données dans un but bien précis qui est souvent la prise de décision. Par exemple, si on considère une chaîne de magasins spécialisés dans la vente de pièces pour automobiles, cela veut dire que chacune des bases transactionnelles contient des informations sur l'historique des achats d'un même ensemble de clients (automobilistes).

Faire du data mining (Fouille de données) sur toutes les bases transactionnelles offre au décideur une panoplie de choix possibles. Mais le problème de confidentialité de ces données tient encore sa place; prenant le cas d'une base de données médicale et une base de données d'une compagnie aérienne.

Sachant que les données médicales sont privées et que la connaissance par la compagnie qu'un patient (en l'occurrence client de la compagnie) est par exemple cardiaque, est une violation de sa vie privée. La problématique est de résoudre ces points de conflit entre le désir de combiner les informations des deux bases de données et le respect de la vie privée de l'individu concerné (patient).

La principale solution qui fut proposée est le calcul distribué sécuritaire multi-parties ou SMC (Secure Multi-Party Computation) dont le principe est de distribuer le calcul d'une fonction sur un ensemble de données réparties entre plusieurs entités. Chaque entité ne possède qu'une partie des données. Le calcul de cette fonction doit être de telle façon à ce qu'aucune des parties impliquées ne puisse déduire les données des autres entités à partir des résultats du calcul et de ses propres données.

Yao [[74], [75]] a initié les travaux dans ce domaine en proposant une approche bipartie, qui fut généralisée par [76]. Cette approche a été introduite dans la communauté datamining par Lindell et Pinkas dans [77] avec une méthode permettant à deux parties de construire un arbre de décision sans qu'aucune des parties ne puisse déduire quoi que ce soit sur l'autre. D'autres techniques ont depuis vu le jour pour la construction d'arbre de décision dans des bases de données distribuées verticalement [60], l'extraction des règles d'association dans le cas des bases de données distribuées verticalement [[78], [60]] et horizontalement [[56], [59]], la recherche de motifs séquentiels dans un contexte collaboratif [[79],[62]], . . .

III. Présentation des 3 Algorithmes (Vaidya, DNSP, PSPP) :

Les 3 protocoles dont nous allons comparer les résultats sont : DNSP, Vaidya, et Private Scalar Product.

-Le vecteur d'Alice comme le vecteur de Bob est déclaré comme un tableau de BigInteger. Nous avons utilisé la classe BigInteger car elle permet de faire des opérations arithmétiques, la génération automatique de nombre premier, etc.

- Conformément à la programmation socket, une partie(Alice) est désignée serveur et l'autre(Bob) est client. Dans le code du client(Bob), au niveau de l'instruction suivante : requestSocket = new Socket ("localhost",2004);, localhost doit être remplacé par l'adresse IP de la machine sur laquelle tourne le serveur(Alice).

- Au niveau de chaque fonction main, la fonction Init (int n) est appelée pour initialiser un vecteur aléatoire de dimension n.

- L'environnement de l'expérimentation doit être constitué de deux machines de même caractéristiques reliées par Ethernet.

-En ce qui concerne l'ordre d'exécution des codes, le code du serveur(Alice) doit être exécuté en premier. Le message "waiting for connexion" est affiché pour montrer que le serveur(Alice) attend la connexion du client(Bob).

Le code du client est exécuté en dernier lieu.

3.1. Protocol DNSP (Distributed Non-private Scalar Product):

Algorithm 1 DNSP

Require: N=2 (number of sites; Alice and Bob), Alice vector: $\vec{X} = (x_1, ..., x_n)$, Bob vector: $\vec{Y} = (y_1, ..., y_n)$

1: **for** Alice **do**
2: Sends $(x_1, ..., x_n)$ to Bob;
3: **end for**
4: **for** Bob **do**
5: Computing $\vec{X} \bullet \vec{Y} = \sum_1^n x_i \bullet y_i$;
6: Sends $\vec{X} \bullet \vec{Y}$ to Alice;
7: **end for**

L'algorithme DNSP est défini comme le calcul de deux vecteurs sans contrainte de vie privée (i.e. une des parties (ici, Alice) envoie la totalité de ces données à l'autre partie (Bob) ; Ce dernier calcule le produit scalaire et envoie le résultat au premier). Le coût de communication dans DNSP est de $n + 1$ messages. Ce coût est considéré comme un coût de référence pour les Algorithmes de préservation de la vie privée. DNSP nécessite aussi n opérations de multiplication et $n - 1$ opérations d'addition. De part sa nature simpliste, ce protocole ne présent aucune surcharge en matière de communication, vu qu'il ne crypte pas les données ; ce qu'aucun algorithme de préservation de la vie privée n'arrivera à égaler.

3.2. Protocole de Vaidya (solution algébrique) :

Algorithm 2 Protocole de Vaidya

Require: N=2 is number of sites, site A and site B
Require: Each site has a vector of cardinality n. Thus,
Require: A has $\overrightarrow{X} = (x_1, ..., x_n)$ and
Require: B has $\overrightarrow{Y} = (y_1, ..., y_n)$
Both A and B together decide on a random $n \times n/2$ matrix C
1: **for** Site A **do**
2: A generates a random vector R of cardinality $n/2$ ($\overrightarrow{R} = (R_1, R_2, ..., R_{n/2})$)
3: A generates the $n \times 1$ addition matrix X' by multiplying C with R (i.e $X' = C \times R$)
4: A generates the $X'' = X + X'$
5: A sends X'' to B
6: **end for**{First message from A to B}
7: **for** Site B **do**
8: B generates the scalar product S' of X" and Y (i.e. $S' = \sum_{i=1}^{n} x_i'' \times y_i$)
9: B also generates the $n \times 1$ matrix $Y' = C^T \times Y$
10: B sends both S' and Y' to A
11: **end for**{First message from B to A}
12: **for** Site A **do**
13: A generates the subtraction factor $S'' = \sum_{i=1}^{n} Y_i' \times R_i$
14: A generates the required scalar product $S = S' - S''$
15: A reports the scalar product S to B
16: **end for**{Second message from A to B}

Ce Protocole présenté dans [J.Vaidya, C.Clifton 2002], est une solution algébrique qui utilise une matrice de décision de dimension $n \times n/2$ (n : la dimension des vecteurs).

Chaque partie crypte son vecteur en utilisant la matrice de décision. Dans la première étape, le site A (primaire) génère $n/2$ valeurs aléatoires et exécute n^2 opérations de multiplications et d'additions. Le site B répond, en exécutant pour sa part $n^2 + 2$ opérations de multiplications et d'additions. En dernier lieu, le site A exécute $n/2$ opérations de multiplications et d'additions.

Le protocole de [J.Vaidya,C.Clifton 2002] a donc un coût de calcul de $O(n^2)$. Le coût de communication est de $O(n)$, $3n/2 + 2$ messages en supposant que chaque entrée de vecteur nécessite un message. La surcharge en coût de communication est : $n/2 + 1$

3.3. Private Scalar Product Protocol :

Algorithm 7 Private Scalar Product Protocol

Require: N=2 (number of sites ; Alice and Bob), Alice vector : $\overrightarrow{X} = (x_1, ..., x_n)$, Bob vector : $\overrightarrow{Y} = (y_1, ..., y_n)$

1: **for** Alice **do**
2: Generates a pair of key (sk, pk) ;
3: Generates $(Enc_{pk}(x_1), ..., Enc_{pk}(x_n))$ using the public key pk ;
4: Sends $(Enc_{pk}(x_1), ..., Enc_{pk}(x_n))$ to Bob ;
5: **end for**
6: **for** Bob **do**
7: Computing $\prod_{i=1}^{n} p_i$, where $p_i = Enc_{pk}(x_i)$ if $y_i = 1$ and $p_i = 1$ if $y_i = 0$ then uses the additive property of homomorphic encryption for computing $\prod_{i=1}^{n} p_i = Enc_{pk}(\overrightarrow{X} \bullet \overrightarrow{Y})$;
8: Sends $Enc_{pk}(\overrightarrow{X} \bullet \overrightarrow{Y})$ to Alice ;
9: **end for**
10: **for** Alice **do**
11: Computes $Dec_{sk}(Enc_{pk}(\overrightarrow{X} \bullet \overrightarrow{Y}))$;
12: Sends the final result to Bob ;
13: **end for**

3.3.1. L'algorithme :

Dans l'approche que nous proposons, Alice génère d'abord une paire de clés et calcule $\mathrm{Enc_{pk}}(\mathrm{x_i}, \mathrm{r})$ qu'elle envoie à Bob. La sécurité sémantique, équivalente à l'indisguabilité [13], garantit qu'aucune information n'est révélée à travers ce message. Alice envoie aussi la clé publique à Bob. Bob calcule $\prod_{i=1}^{n} p_i$ avec $p_i = \mathrm{Enc_{pk}}(\mathrm{x_i})$ si $y_i = 1$, $p_i = 1$ dans le cas contraire. Il utilise ensuite la propriété additive de l'homomorphisme pour obtenir $\mathrm{Enc_{pk}}(\vec{\mathrm{X}} \bullet \vec{\mathrm{Y}})$ à Alice. Possédant la clé secrète, Alice calcule $Dec_{sk} = \mathrm{Enc_{pk}}(\vec{\mathrm{X}} \bullet \vec{\mathrm{Y}})$ pour retrouver le résultat final qu'elle envoie à Bob.

3.3.2. Analyse du coût de communication et de calcul :

Du point de vue communication, notre protocole a besoin des messages suivants :

(i) pour chaque entrée de vecteur, notre protocole nécessite un message ;
(ii) un message est requis pour envoyer la clé publique ;
(iii) Bob à besoin d'envoyer $\mathrm{Enc_{pk}}(\mathrm{x_i}, \mathrm{r})$ à Alice ;

Enfin, Alice doit envoyer le résultat du produit scalaire à Bob. Ainsi, le nombre de messages est $x + 3$ ou n est la dimension du vecteur. Dans ce cas, nous obtenons une complexité, en terme de

coût de communication, de l'ordre de $O(n)$ Du point de vue calcul, Alice exécute, dans notre protocole, n opérations de cryptage et 1 opération de décryptage. Bob exécute moins de $n-1$ additions. Ceci donne une complexité linéaire, de l'ordre de$O(n)$.

3.3.3. Analyse de la sécurité :

La sécurité de notre protocole est basée sur les propriétés du cryptosystème utilisé :

(i) le système de cryptage de Paillier [13] est probabiliste ;

(ii) le système de cryptage de Paillier [13] est un chiffrement homomorphe.

Pour analyser la sécurité, examinons les informations propagées par chaque site participant au protocole :

– Vue d'Alice : Dans les étapes 2, 3 et 4, pour chaque x_i , $i \in \{1..n\}$ Alice choisit un nombre aléatoire r et calcule $\mathrm{Enc_{pk}}(\mathrm{x_i}, \mathrm{r})$, puis l'envoie à Bob.

– Vue de Bob : Bob reçoit $\mathrm{Enc_{pk}}(\mathrm{x_i}, \mathrm{r})$ Ce message est indistinguable au calcul [13], i.e. qu'il n'est pas identifiable par étude statistique, puisque le cryptage est sémantiquement sécurisé. Donc pour chaque entrée x_i , $i \in \{1..n\}$ un adversaire ne peut reconnaître le chiffré correspondant. La probabilité de retrouver le vecteur d'Alice en entier est alors très faible, sachant que la dimension de celui-ci est généralement grande (notons que la technologie du datamining s'applique sur de larges volumes de données). Dans l'étape 7, Bob calcule $\mathrm{Enc_{pk}}(\vec{\mathrm{X}} \bullet \vec{\mathrm{Y}}) = \prod \mathrm{Enc_{pk}}(\mathrm{x_i})$ si $y_i = 1$ La propriété homomorphe du cryptosystème garantit que ce calcul ne révèlera pas les valeurs de x_i , $i \in \{1..n\}$. Aussi, dans l'étape 8, Bob envoie $\mathrm{Enc_{pk}}(\vec{\mathrm{X}} \bullet \vec{\mathrm{Y}})$ à Alice. La sécurité de cette étape, de même que les étapes suivantes, n'est pas importante car le résultat du produit scalaire n'est pas privé.

3.3.4. Travaux connexes :

La préservation de la vie privée ne peut se faire et être efficace qu'à un certain coût. Ce coût est souvent comparé au coût de celui de DNSP (Distributed Non-private Scalar Product) qui est défini comme étant le calcul du produit scalaire de deux vecteurs sans contrainte de vie privée (i.e. une des parties envoie la totalité de ses données à l'autre partie qui calcule le produit scalaire et envoie le résultat au premier). Le coût de communication dans DNSP est de l'ordre de $n+1$ messages. DNSP nécessite n opérations de multiplication et $n-1$ opérations d'addition. Dans le tableau 1, nous comparons notre protocole avec DNSP et deux autres Vaidya et Camara [60, 73] qui sont très connus dans le domaine de la préservation de la vie privée.

	DNSP(Distributed Non-private Scalar Protocol)	Protocole de Vaidya	Private Scalar Product Protocol
Coût de communication	$n+1$	$3n/2+2$	$n+3$
Surcoût de communication	0	$n/2+1$	2
Complexité de calcul	$O(n)$	$O(n^2)$	$O(n)$

Tableau 1 : Tableau comparatif entre protocoles de sécurisation.

L'analyse du tableau de la Tableau 1 met en évidence la différence entre notre protocole et les autres, suivant trois métriques : la complexité de calcul, le coût de communication et le surcoût de communication. Si la complexité de calcul n'est pas très critique en datamining c'est parce qu'il existe des algorithmes et des architectures parallèles permettant d'obtenir des complexités de calcul acceptables. Par contre, le nombre de messages échangés entre les différents sites peut créer un goulot d'étranglement, qui pourrait considérablement dégrader les performances d'un processus de datamining. C'est donc un critère non négligeable dans la proposition d'un protocole de cryptage.

IV. Résultats et Perspectives :

4.1. Résultats :

4.1.1. Résultats de l'application du chiffrement de Paillier :

1ère exécution :

Java Paillier $m1 = 0$; $m2 = 1$ (Lancement) ;

$E(m1) =$
$232077517264917208772424314006496236076327448850961388680763134$
$657086803948327407708714334301727473208163833201626913955359124399423640$
$192167555951623196697225698543777697825225494147668491736304805388260709$
$909652868993049462490140982074695404284862803689510993118298913925146303$
$202733161505093273252818896 22$

$E(m2) =$
$320346932988128038688938081215472517439335662363773552205435003$
$469303751394790262018940039814905213279093070064771557148958534908564579$
$269305499705863061920607743531231347007523968522897207136966263744402075$
$436202568112987640252091060102025741922222579500577020221576672212877653$
$181095787148567381738967510 37$

$D\big(E(m1)\big) = 0$;
$D\big(E(m2)\big) = 1$;
$E(m1) * E(m2) mod\ n^2 = E(m1 + m2 mod\ n)$
$i.e\ D(E(m1) * E(m2) mod\ n^2) = (m1 + m2) mod\ n$;
$D(E(m1) * E(m2) mod\ n^2) = 1 (m1 + m2) mod\ n = 1$

2ème exécution :

Java Paillier $m1 = 0$; $m2 = 1$

$E(m1) =$
$122838445382525285555824785297689710755230478388819163039585477830$
$772397905011799956890892273592039107735877993892820852112924290960301188$
$889600403951646501523723096782687927877055910139380979070956380110451969$
$365877625229375011904181185980831313276646313615393766188128828748841217$
$992838564380742667331787 0654$

$E(m2) =$
19139991506437636935785718109659071007489687861871951500273804527251622000603541683260709083038681043154687762673798698543073741836038612938178592467062911445632284203833399052637803634610953597352880753032733995662905421419538310786582245774110415953437964415523821739897519272650167671715070662198742622 49

$D\big(E(m1)\big) = 0$;
$D\big(E(m2)\big) = 1$;
$E(m1) * E(m2) mod\ n^2 = E(m1 + m2 mod\ n)$
$i.e\ D(E(m1) * E(m2) mod\ n^2) = (m1 + m2) mod\ n$;
$D(E(m1) * E(m2) mod\ n^2) = 1 (m1 + m2) mod\ n = 1$

- Le chiffré de $m0$ ou $m1$ change, donc le chiffrement est probabiliste.

$D(E(m1) * E(m2) mod\ n^2) = (m1 + m2) mod\ n$; i.e
$E(m1) * E(m2) mod\ n^2 = E(m1 + m2 mod\ n)$ donc le chiffrement est homomorphe additif.

4.2. Résultats Expérimentaux :

Le tableau 2 montre la variation du temps d'exécution moyen (en unité de10^7 secondes)

On peut remarquer que : Le protocole PSPP fournit un gain en matière de temps moyen d'exécution par rapport µa celui de Vaidya et aussi qu'en ayant un nombre de nœuds entre 200 et 800, même les différences qui ne sont pas significatives restent du domaine du raisonnable.

Les meilleurs résultats ont été obtenus avec une taille de vecteurs entre 100000 et 150000. Ceci est du µa une meilleure utilisation des ressources après un équilibrage de la charge entre les nœuds(Le système n'est ni trop chargé, ni sous-chargé).

Un exemple de la figure suivante montre que pour un nombre de nœuds de 600 nœuds, le résultat est significatif.

On peut aussi par le biais de la table 3 jeter un coup d'œil sur le temps moyen de communication, qui peut servir de métrique pour le contrôle de la bonne fluidité du trafic dans la grille et anticiper d'éventuels goulots d'étranglement.

Nœuds / Taille Vecteurs		50000	100000	150000	200000
200	Algo Vaidya	5.561	3.786	1.365	1.183
	P.S.P.P	3.285	1.365	0.418	0.583
400	Algo Vaidya	9.222	8.798	7.369	2.066
	P.S.P.P	4.166	3.446	2.641	1.381
600	Algo Vaidya	9.394	9.812	9.154	3.936
	P.S.P.P	6.545	3.547	4.589	2.774
800	Algo Vaidya	19.515	18.725	18.135	16.578
	P.S.P.P	12.878	9.669	9.971	13.236
1000	Algo Vaidya	28.478	30.225	25.615	21.825
	P.S.P.P	20.795	23.615	19.315	15.735

Table 2 : Temps Moyen d'Exécution (en 10^7 secondes)

Le temps moyen de communication diminue de manière constante jusqu'µa atteindre son niveau le plus bas et donc confirme que notre protocole ne surcharge pas le réseau comme le montre la 3 en comparaison avec l'algorithme de Vaidya.

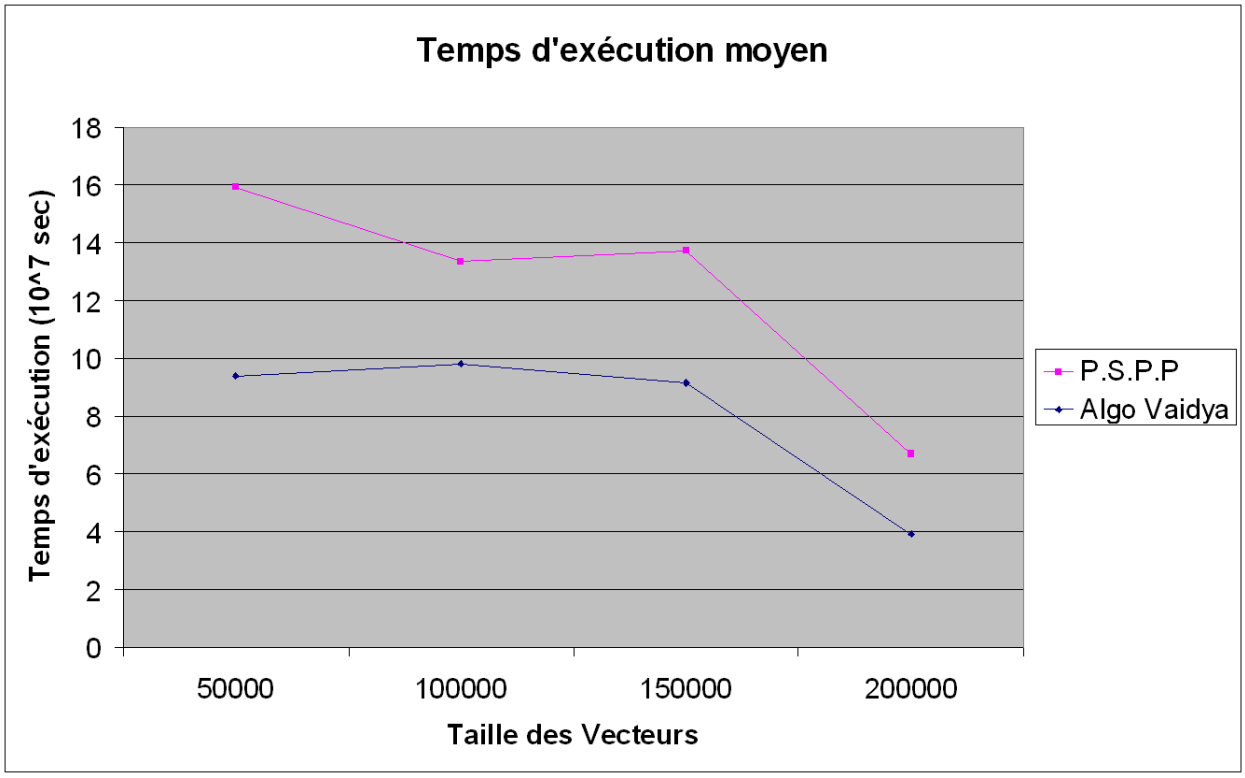


Figure 5.2 : Temps d'exécution moyen

Nœuds / Taille Vecteurs		50000	100000	150000	200000
200	Algo Vaidya	0.097	0.064	0.07	0.061
	P.S.P.P	0.068	0.049	0.055	0.05
400	Algo Vaidya	0.136	0.107	0.096	0.072
	P.S.P.P	0.102	0.087	0.073	0.056
600	Algo Vaidya	0.185	0.191	0.189	0.177
	P.S.P.P	0.144	0.155	0.166	0.157
800	Algo Vaidya	0.287	0.286	0.275	0.251
	P.S.P.P	0.248	0.237	0.217	0.132
1000	Algo Vaidya	0.323	0.330	0.329	0.294
	P.S.P.P	0.279	0.264	0.287	0.234

Table 3 : Temps Moyen de Communication (en 10^7 secondes)

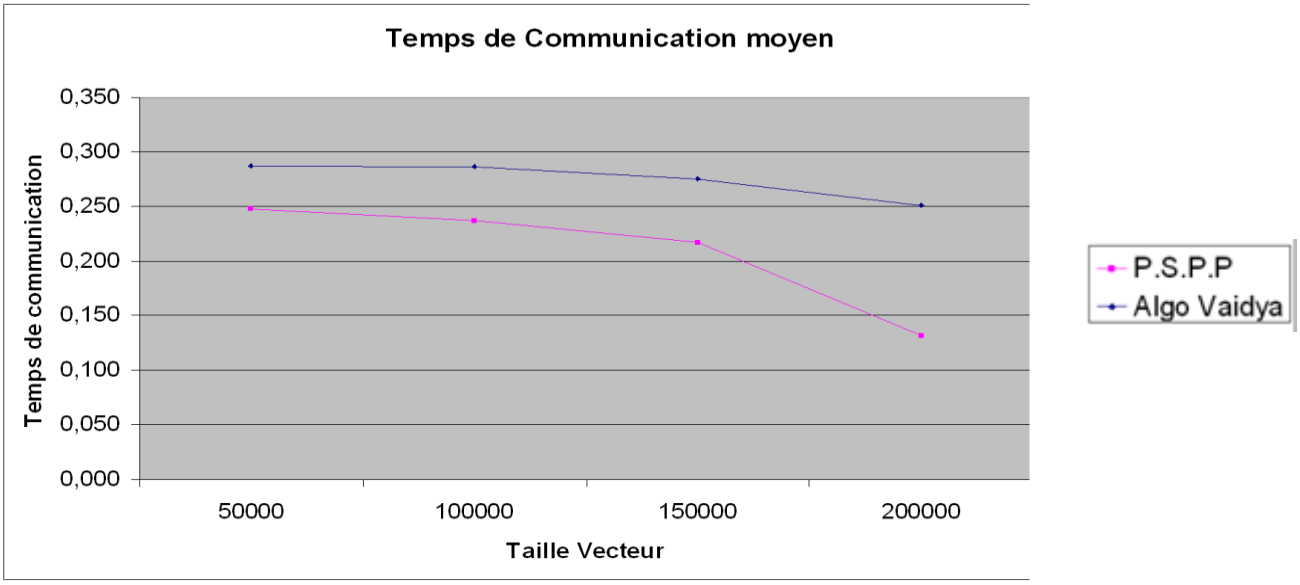


Figure 5.3 : Temps de communication moyen

Conclusion

Le Domaine de Confidentialité des données dans les grilles de calcul, et de façon plus explicites dans les systèmes parallèles et distribués est très vaste et difficile à cerner, vu qu'il se trouve à l'intersection de plusieurs disciplines ; cependant notre travail nous a permis de prendre conscience des défis à relever, surtout pour la communauté du data mining. On a été confronté durant ce projet à des difficultés de l'ordre technique, vu que la plate-forme grille n'était pas évidente à réaliser ou à simuler. Ceci dit, ce n'est qu'en forgeant qu'on devient forgeron ; on espère donc continuer dans notre lancée pour pouvoir apporter notre contribution dans tout ce qui touche à la confidentialité des données dans les grilles de calcul.

Dans ce mémoire nous avons abordé la problématique de la préservation de la vie privée dans le datamining. Nous avons présenté une large description et le groupement d'une panoplie d'algorithmes de datamining et vie privée.

Le travail présenté dans ce document démontre que datamining et vie privée ne sont pas naturellement en conflit. Pour les différentes techniques de datamining (extraction de règles d'association, recherche de motifs séquentiels, classification, etc.), différentes solutions ont été proposées pour la protection de la vie privée. Nous avons vu que pour un partitionnement vertical des données, plusieurs applications de datamining se réduisent au calcul du produit scalaire.

Plusieurs protocoles ont été proposés pour le calcul sécurisé de cette primitive.

L'évaluation de ces différents protocoles montre qu'ils provoquent une surcharge sur la communication. Le protocole Private Scalar Product Protocol possède l'avantage de ne pas provoquer une telle surcharge ; cependant son application reste toujours du domaine binaire.

Perspectives :

- Bien que les principes de conception de notre approche nous semblent pertinent et consistent à la fois, il est possible d'en apporter une modification. La plus probable consiste à régler la limitation de l'application du protocole dans un contexte binaire en l'étendant à un contexte plus général de bases de données transactionnelles. De plus, on passera à une échelle d'application plus générale, et qui reflète beaucoup plus la réalité, qui est celle d'une vraie grille de Calcul, en utilisant un framework, en l'occurrence : Alchemy, qui est un framework offrant les fonctionnalités d'analyse de données, de génération de rapport et d'intégration sur une même plateforme, dans un réseau de quinze PCs.

- Comparaison des résultats fournis par le simulateur avec ceux comparables à une Grille de Calcul.

- Etablir un seuil de Confiance des résultats fournies par les simulateurs, en mettant en évidence le rôle de l'équilibrage de charges dans l'amélioration des performances de la grille de calcul.

Bibliographie

Bibliographie

[1] I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the grid : Enabling scalable virtual organizations. International J. Supercomputing Applications, 15(3) :200{222, August 2001.

[2] I. Foster and C. Kesselmann. The Grid 2 : Blueprint for a New Computing Infrastructure. Elsevier Series in Grid Computing, 2004.

[3] J-M. Pierson. Une Grille Pervasive Vue du Coté des Données. PhD thesis, Université Claude Bernard - Lyon I, France, Novembre 2005.

[4] F. Berman, A. J. C. Hey, and G. C. Fox. Grid Computing : Making the Global Infrastructure a Reality. Wiley Sons, 2003.

[5] I. Foster and C. Kesselmann. The Grid : Blueprint for Future Computing Infrastructure. Morgan Kaufmann, San Francisco, 1999.

[6] R. Buyya and S. Venugopal. A gentle introduction to grid computing and technologies. CSI Communications : Computer Science of India, 29(1) :9{19, July 2005.

[7] I. Foster. What is the grid ? a three point checklist. GRIDtoday, 1(6), 22 July 2002. http ://www.gridtoday.com/02/0722/100136.html.

[8] P. Plaszczak and R. Jr. Wellner. Grid Computing : The Savvy Manager's Guide. Morgan Kaufmann, San Francisco, 2005.

[9] H. Stockinger. Database Replication in World-Wide Distributed Data Grids. PhD thesis, CERN (European Organization for Nuclear Research), Genf, Schweiz, November 2001.

[10] Z. Juh_asz, P. Kacsuk, and D. Kranzlmuller. Distributed and Parallel Systems : Cluster and Grid Computing. Springer, Boston, USA, 2005.

[11] A-G. Bosser. Réplications Distribuées pour la Définition des Interactions de Jeux Massivement Multi-Joueurs. PhD thesis, Université de Paris 7, France, Novembre 2005.

[12] R. Buyya. Economic-based Distributed Resource Management and Scheduling for Grid Computing. PhD thesis, Monash University, Melbourne, Australia, April 2002.

[13] L. Simoncini. Grid computing evolution and challenges for resilience, performance and scalability. Workshop on Grid Computing and Dependability, 48th Meeting of the IFIP Working Group 10.4, Hakone, Japan, July 1-5 2005.

[14] C. Haddad and Y. Slimani. Economic model for replicated database placement in grid. In 7th IEEE International Symposium on Cluster Computing and the Grid (CCGrid'07), pages 283{289, Rio de Janeiro, Brazil, 14-17 May 2007. IEEE Computer Society.

[15] S. Monnet. Gestion des Données dans les Grilles de Calcul : Support pour la Tolérance aux Fautes et la Cohérence des Données. PhD thesis, Université de Rennes, France, Novembre 2006.

[16] A. Imine. Conception Formelle d'Algorithmes de Réplication Optimiste Vers l'Edition Collaborative dans les réseaux Pair-a-Pair. PhD thesis, Doctorat de l'université Henri Poincaré Nancy 1, France, Novembre 2006.

[17] http://www.cs.utk.edu/netSolve

[18] http://www.edonkey2000-france.com

[19] http://www.emule-france.com

[20] Aggarwal C. C.: On Randomization, Public Information and the Curse of Dimensionality. *ICDE Conference*, 2007.

[21] Aggarwal C. C., Yu P. S.: On Privacy-Preservation of Text and Sparse Binary Data with Sketches. *SIAM Conference on Data Mining*, 2007.

[22] Aggarwal C. C., Yu P. S. On Anonymization of String Data. *SIAM Conference on Data Mining*, 2007.

[23] Aggarwal G., Feder T., Kenthapadi K., Motwani R., Panigrahy R., Thomas D., Zhu A.: Anonymizing Tables. *ICDT Conference*, 2005.

[24] Aggarwal G., Feder T., Kenthapadi K., Motwani R., Panigrahy R., Thomas D., Zhu A.: Approximation Algorithms for *k*-anonymity. *Journal of Privacy Technology*, paper 20051120001, 2005.

[25] Aggarwal G., Feder T., Kenthapadi K., Khuller S., Motwani R., Panigrahy R., Thomas D., Zhu A.: Achieving Anonymity via Clustering. *ACM PODS Conference*, 2006.

[26] Aggarwal C., Pei J., Zhang B. A Framework for Privacy Preservation against Adversarial Data Mining. *ACM KDD Conference*, 2006.

[27] Bertino E., Fovino I., Provenza L.: A Framework for Evaluating Privacy-Preserving Data Mining Algorithms. *Data Mining and Knowledge Discovery Journal*, 11(2), 2005.

[28] Aggarwal C. C., Yu P. S.: A Condensation approach to privacy preserving data mining. *EDBT Confe rence*, 2004.

[29] *Aggarwal C. C., Yu P. S.: On Variable Constraints in Privacy-Preserving Data Mining. SIAM Conference, 2005.*

[30] Aggarwal C. C.: On Randomization, Public Information and the Curse of Dimensionality. *ICDE Conference*, 2007.

[31] Aggarwal C. C., Yu P. S.: A Condensation approach to privacy preserving data mining. *EDBT Conference*, 2004.

[32] Aggarwal C. C., Yu P. S.: On Variable Constraints in Privacy-Preserving Data Mining. *SIAM Conference*, 2005.

[33] Agrawal R., Srikant R. Privacy-Preserving Data Mining. *Proceedings of the ACM SIGMOD Conference*, 2000.

[34] Agrawal D. Aggarwal C. C. On the Design and Quantification of Privacy-Preserving Data Mining Algorithms. *ACM PODS Conference*, 2002.

[35] Liew C. K., Choi U. J., Liew C. J. A data distortion by probability distribution. *ACM TODS*, 10(3):395-411, 1985.

[36] Warner S. L. Randomized Response: A survey technique for eliminating evasive answer bias. *Journal of American Statistical Association*, 60(309):63–69, March 1965.

[37] Silverman B. W.: Density Estimation for Statistics and Data Analysis. *Chapman and Hall*, 1986.

[38] Aggarwal G., Feder T., Kenthapadi K., Motwani R., Panigrahy R., Thomas D., Zhu A.: Anonymizing Tables. *ICDT Conference*, 2005.

[39] Aggarwal G., Feder T., Kenthapadi K., Khuller S., Motwani R., Panigrahy R., Thomas D., Zhu A.: Achieving Anonymity via Clustering. *ACM PODS Conference*, 2006.

[40] Atallah, M., Elmagarmid, A., Ibrahim, M., Bertino, E., Verykios, V.: Disclosure limitation of sensitive rules, *Workshop on Knowledge and Data Engineering Exchange*, 1999.

[41] Samarati P.: Protecting Respondents' Identities in Microdata Release. IEEE Trans. Knowl. Data Eng. 13(6): 1010-1027 (2001).

[42] Bettini C., Wang X. S., Jajodia S.: Protecting Privacy against Location Based Personal Identification. *Proc. of Secure Data Management Workshop*, Trondheim, Norway, 2005.

[43] Blum A., Dwork C., McSherry F., Nissim K.: Practical Privacy: The SuLQ Framework. *ACM PODS Conference*, 2005.

[44] Bayardo R. J., Agrawal R.: Data Privacy through Optimal k- Anonymization. *Proceedings of the ICDE Conference*, pp. 217–228, 2005.

[45] Domingo-Ferrer J,, Mateo-Sanz J.: Practical data-oriented microaggregation for statistical disclosure control. *IEEE TKDE*, 14(1), 2002.

[46] Machanavajjhala A., Gehrke J., Kifer D., and Venkitasubramaniam M.: l-Diversity: Privacy Beyond k-Anonymity. *ICDE*, 2006.

[47] Li N., Li T., Venkatasubramanian S: *t*-Closeness: Orivacy beyond *k*-anonymity and *l*-diversity. *ICDE Conference*, 2007.

[48] Pinkas B.: Cryptographic Techniques for Privacy-Preserving Data Mining. *ACM SIGKDD Explorations*, 4(2), 2002.

[49] Verykios V. S., Elmagarmid A., Bertino E., Saygin Y.,, Dasseni E.: AssociationRule Hiding. *IEEE Transactions on Knowledge and Data Engineering*, 16(4), 2004.

[50] Moskowitz I., Chang L.: A decision theoretic system for information downgrading. *Joint Conference on Information Sciences*, 2000.

[51] Adam N., Wortmann J. C.: Security-Control Methods for Statistical Databases: A Comparison Study. *ACMComputing Surveys*, 21(4), 1989.

[52] Chang L., Moskowitz I.: Parsimonious downgrading and decision trees applied to the inference problem. *New Security Paradigms Workshop*, 1998.

[53] Natwichai J., Li X., Orlowska M.: A Reconstruction-based Algorithm for Classification Rules Hiding. *Australasian Database Conference*, 2006.

[54] Huang Z., DuW., Chen B.: Deriving Private Information from Randomized Data. pp. 37–48, *ACM SIGMOD Conference*, 2005.

[55] Hore B., Mehrotra S., Tsudik B.: A Privacy-Preserving Index for Range Queries. *VLDB Conference*, 2004.

[56] Hughes D, Shmatikov V.: Information Hiding, Anonymity, and Privacy: A modular Approach. *Journal of Computer Security*, 12(1), 3–36, 2004.

[57] Inan A., Saygin Y., Savas E., Hintoglu A., Levi A.: Privacy-Preserving Clustering on Horizontally Partitioned Data. *Data Engineering Workshops*, 2006.

[58] Ioannidis I., Grama A., Atallah M.: A secure protocol for computing dot products in clustered and distributed environments, *International Conference on Parallel Processing*, 2002.

[59] M.Kantarcioglu,C.Clifton,J.Vaidya,X.Lin,M.Y.Zhu 2003, Tools for Privacy Preserving Data Mining. SIGKDD Explrations,4(2), 28-34,2003.

[60] J. S. Vaidya and C. Clifton. Privacy preserving association rule mining in vertically partitioned data. In The Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 639-644, July 23-26 2002.

[61] David Hand, Heikki Mannila, and Padhraic Smyth. Principles of Data Mining. MIT Press, August 2001.

[62] Jiawei Han and Micheline Kamber. Data Mining :Concepts and Techniques. Morgan Kaufmann, April 2000.

[63] Richard O. Duda and Peter E. Hart. Pattern Classification and Scene Analysis. John Wiley & Sons, June 1973.

[64] Philip Chan. An Extensible Meta-Learning Approach for Scalable and Accurate Inductive Learning. PhD thesis, Department of Computer Science, Columbia University, New York, NY, 1996.

[65] Philip Chan. On the accuracy of meta-learning for scalable data mining. Journal of Intelligent Information Systems, 8 :528, 1997

[66] Andreas Prodromidis, Philip Chan, and Salvatore Stolfo. Advances in Distributed and Parallel Knowledge Discovery, chapter 3 :Metalearning in distributed data mining systems : Issues and approaches. AAAI/MIT Press,September 2000.

[67] Ashoka Savasere, Edward Omiecinski, and Shamkant B.Navathe. An efficient algorithm for mining association rules in large databases. In Proceedings of Twenty-1rst International Conference on Very Large Data Bases, pages 432- 444, Zurich, Switzerland, September 11-15 1995. VLDB.

[68] Kargupta Chan 2000[69] Zaki M., " SPADE : An e±cient algorithm for mining frequent sequences", Machine Learning Journal, vol. 42, nº1, p. 31-60, February,1999

[70] S. L. Warner. Randomized response : a survey technique for eliminating evasive answer bias. Journal of the American Statistical Association, 60(309) :63-69, Mar. 1965

[71] W. Du et Z. Zhan. 2003

[72] Y. Zhu and L. Liu. Optimal randomization for privacy preserving data mining. In KDD '04 : Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 761-766, New York, NY, USA, 2004. ACM Press. Falcon fraud managerEA. http
://www.fairisaac.com/Fairisaac/Solutions/Solutions+by+Function/Falcon+Fraud+Mcinager.htm, 2005.

[73] FODE CAMARA,thesis draft, Dept of Mathematics and Computer Science, Senegal.2009

[74] Yao A. C., " Protocols for secure computations ", Proc. of the 23rd annual IEEE Symposium on Foundations of Computer Science, p. 160-164, 1982.

[75] Yao A. C., " How to generate and exchange secrets ", Proc. of the 27th Sympo-sium on Foundations of Computer Science (FOCS), p. 162-167, 1986

[76] Du W., Atallah M. J., " Secure multi-party computation problems and their applications : a review and open problems ", New Security Paradigms Workshop, Cloudcroft,USA, p. 11-20, 2001.

[77] Y. Lindell and B. Pinkas. Privacy preserving data mining. In Advances in Cryptology - CRYPTO 2000, pages 36-54. Springer-Verlag, Aug. 20-24 2000.

[78] Cynthia Dwork and Kobbi Nissim. Privacy-preserving Data mining on Vertically Partitioned Databases. In Proceedings of The 24rd Annual International Cryptology Conference (CRYPTO 2004), Santa Barbara, CA, August 2004.

[79] M. Atzori, F. Bonchi, F. Giannotti, and D. Pedreschi. Kanonymous patterns.In Proceedings of the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 05), Porto,Portugal, 2005.