Université Abou Bekr Belkaid
Tlemcen Algérie
جامعة أبي بكر بلقايد
تلمسان الجزائر

République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatiqueμ

Mémoire de fin d'études

pour l'obtention du diplôme de Master en Informatique

OPTION : MODELE INTELLIGENT ET DECISION (M.I.D)

Thème

PROTECTION DES ITEMSETS FREQUENTS SENSIBLES PAR RECONSTRUCTION DES BASES DE DONNEES TRANSACTIONNELLES

Réalisé par :

- **BELLIFA Mohammed Islam**
- **BENZENGLI Mohammed Riadh**

Présenté le 04 juillet 2022 devant le jury composé de MM.

- *Mr. HADJILA FETHALLAH* *(Président)*
- Mr. BENTALLAH MOHAMMED AMINE *(Examinateur)*
- *Mr. BELABED AMINE* *(Encadreur)*

Année universitaire : 2021-2022

REMERCIMENT :

Nous tenons tout d'abord à nous remercierons ***ALLAH*** *tout puissant qui nous donnés la force, la volonté et la patience afin d'accomplir ce modeste travail.*

Ce mémoire n'aurait pas été possible sans l'intervention, consciente, d'un grand nombre de personnes que nous souhaitons remercier dans cette modeste page.

Nous tenons aussi à remercier du fond du cœur Monsieur BELABED AMINE qui nous a permis de bénéficier de son encadrement. Nous sommes reconnaissants du soutien que vous avez apporté dans le cadre de notre projet durant toute la durée du deuxième semestre.

Sans vous, nous sommes convaincus que nous n'aurions pas pu obtenir un aussi bon résultat, En effet, votre pédagogie et votre patience ont rendu cela possible. Vous avez toujours su rester à notre écoute et votre soutien permanent nous a été réellement précieux. Grâce à vous, nous avons pu progresser dans le domaine. Pour tout ceci, nous vous sommes sincèrement reconnaissants.

Nous ne saurions assez remercier les examinateurs du grand honneur qu'ils nous font en jugeant ce travail.

Nos remerciements vont aussi à tous nos enseignants du département d'informatique.

Enfin nous tenons à remercier tous ceux qui ont contribué de prés ou de loin à la réalisation du projet.

DÉDICACE
BELLIFA MOHAMMED ISLAM

Louange à ALLAH tout puissant, qui m'a permis de voir ce jour tant attendu

Je dédie cette thèse :

A ma très chère mère ***BAHIA.***

Affable, honorable, aimable : Tu représentes pour moi le symbole de la bonté par excellence, la source de tendresse et l'exemple du

dévouement qui n'a pas cessé de m'encourager et de prier pour moi.

Je te dédie cetravail en témoignage de mon profond amour.

Puisse Dieu, le tout puissant, te préserver et t'accorder santé, longue vie et bonheur.

A mon très cher père ***LAHCEN.***

Tu as toujours été pour moi un exemple du père respectueux, honnête, de la personne méticuleuse, je tiens à honorer l'homme que tu es.

Grâce à toi papa j'ai appris le sens du travail et de la responsabilité. Je voudrais te remercier pour

ton amour, ta générosité, ta compréhension... Ton soutien fut une lumière dans tout mon parcours.

Aucune dédicace ne saurait exprimer l'amour l'estime et le respect que j'ai toujours eu pour vous mes parents.

Ce modeste travail est le fruit de tous les sacrifices que vous avez déployés pour mon éducation et ma formation.

Je vous aime et j'implore le tout-puissant pour qu'il vous accordez une bonne santé et une vie longue et heureuse.

A ma sœur ***LAMIA*** *et chère frère* ***HOUSSAM***

A tous les moments de mon enfance passés avec vous mes frères, en témoignage de

ma profonde estime pour l'aide que vous m'avez apportée. Vous m'avez soutenu,

réconforté et encouragé. Puissent nos liens fraternels se consolider et se perpétuer encore plus .

A mon ***grand père*** *et ma* ***grande mère****, A mes oncles et tantes*

A celui, le spécial, qui était là pour moi dans les hauts et les bas, qu'Allah vous bénisse.

A TOUTE MA FAMILLE *Aucun langage ne saurait exprimer mon respect et ma vous avez envoyé*

considération pour votre soutien et encouragements. Je vous dédie ce travail en reconnaissance de l'amour que vous m'offrez quotidiennement et votre bonté

exceptionnelle. Que Dieu le Tout Puissant vous garde et vous procure santé et bonheur.

A tout mes amis, Je ne peux trouver les mots justes et sincères pour vous exprimer

mon affection et mes pensées, vous êtes pour moi des sœurs et des amies sur qui je peux compter. En témoignage de l'amitié qui nous unit et des souvenirs de tous les moments que nous avons passés ensemble, je vous dédie ce travail et je vous souhaite une vie pleine de santé et de bonheur.

ISLAM

DÉDICACE
BENZENGLI MOHAMMED RIADH

Je dédie cet ouvrage à mes parents qui m'ont soutenus et encouragés durant ces années d'études qu'ils trouvent ici le témoignage de ma profonde reconnaissance

à ma sœur , mes frères, ma grand-mère et tous ceux qui ont partagé avec moi tous les moments d'émotions lors de la réalisation de ce travail

à ma famille , mes proches et à tous mes amis qui m'ont toujours encouragé et à qui je souhaite plus de succès.

A tout ceux que j'aime.

RIADH

Résumé :

Ce travail se focalise sur le domaine de protections des itemsets fréquents sensibles. Dans ce cadre nous avons implémenté une approche qui assure la protection des données contre tout accès non autorisé suite à un traitement ou une analyse. L'approche est basée sur la reconstitution d'une nouvelle base transactionnelle à partir d'une liste des itemsets fréquents. Les résultats obtenus ont montré que l'approche implémentée assure un grand taux de protection, mais elle nécessite des améliorations sur le plan de la qualité des données produites.

Mots clés : Itemsets fréquents, base transactionnelle, Itemsets sensibles, protection, reconstitution.

Abstract :

This work focuses on the protection domain of sensitive frequent itemsets. In this context, we have implemented an approach that ensures the protection of data against unauthorized access following processing or analysis. The approach is based on the reconstitution of a new transactional database from a list of frequent itemsets. The results obtained showed that the implemented approach provides a high protection rate, but it requires improvements in terms of the quality of the data produced.

Keywords: Frequent itemsets, transactional database, sensitive itemsets, protection, reconstitution.

ملخص

يركز هذا العمل على مجال حماية العناصر المتكررة الحساسة. في هذا السياق ، قمنا بتنفيذ نهج يضمن حماية البيانات من الوصول غير المصرح به بعد المعالجة أو التحليل. يعتمد النهج على إعادة تكوين قاعدة بيانات معاملات جديدة من قائمة مجموعات العناصر المتكررة. أظهرت النتائج التي تم الحصول عليها أن النهج المنفذ يوفر معدل حماية عاليًا ، لكنه يتطلب تحسينات من حيث جودة البيانات المنتجة.

الكلمات الرئيسية: مجموعات العناصر المتكررة ، قاعدة بيانات المعاملات ، مجموعات العناصر الحساسة ، الحماية ، إعادة التكوين.

Table des matières

LISTE DES TABLEAUX

Abbreviation:

- FIM : fréquent itemset mining
- SFIM : secret fréquent itemset mining
- BDD : base de données • NSFIM : non secret fréquent itemset mining
- IFIM : inverse fréquent itemset mining

LISTE DES FIGURES

1 INTRODUCTION :

A. Contexte de travail :

De nos jours les systèmes d'informations donnent accès à de plus en plus de sources de données hétérogènes et distribuées. A fur et à mesure que les sources se multiplient et que le volume de données disponibles s'accroit, les entreprises et la communauté scientifique ont développé des outils et des approches performantes du Datamining pour le traitement de ces grandes quantités de données. Parmi ces approches, c'est *l'extraction des motifs fréquents*. Le but principal de cette approche est de valoriser l'information et la connaissance contenue dans les bases de données, et de découvrir les corrélations entre les motifs fréquents afin d'en tirer de nouvelles informations. Cependant, il existe des situations dans lesquelles une partie des données extraites doivent être tenue secrètes à cause de son caractère sensibles (données privées, secret médical, secret industriel, etc.). Cela, nécessite l'implémentation des approches qui assurent la protection de ces données sensibles.

B. Problématique :

C. Notre travail se focalise sur la protection des Itemsets fréquents sensibles (***Frequent Itemset*** Mining **SFIM**) extraits à partir des bases de données transactionnelles. Le but est de proposer et d'implémenter une approche qui assure la protection des données contre tout accès non autorisé suite à un traitement ou une analyse. L'approche est basée sur la reconstitution d'une nouvelle base transactionnelle à partir d'une liste des Itemsets fréquents (**FIM**) de tel sort que le minage sur cette nouvelle base ne produit pas les Itemsets sensibles (SFIM). Cette nouvelle base sera publiée à la place de la base originale.

C. Objectif :

À part cette d'introduction générale, notre document est organisé sous forme de trois chapitres et une conclusion générale :

- ***LE CHAPITRE I***, intitulé « L'extraction des Itemsets fréquents », présente une vue générale sur le domaine d'excitation des motifs fréquents. Ainsi, nous introduisons dans ce chapitre : les notions de bases de ce domaine, les types des données utilisées, les domaines d'application ainsi que quelques algorithmes d'extraction.

- ***LE CHAPITRE II***, intitulé « Reconstruction des data-sets à base des Itemsets fréquents », est consacré à la définition de la problématique de reconstruction, ainsi qu'un état de l'art sur les méthodes de reconstitution.

- ***LE CHAPITRE III***, introduit les détails de notre travail qui consiste à implémenter une approche de protection des Itemsets fréquents sensibles à base de reconstitution. Le chapitre présente aussi une analyse et une discussion des résultats obtenus.

- Enfin, une ***CONCLUSION GENERALE*** qui conclut notre travail et introduit quelques perspectives.

CHAPITRE I :

L'EXTRACTION DES ITEMSETS FREQUENTS

I. CHAPITRE 1 : EXTRACTION DES ITEMSETS FREQUENT

1.1 INTRODUCTION :

L'extraction des Itemsets fréquents sont l'une des techniques les plus utilisées dans le processus de fouille de données. Introduite par Agrawal et al [1]. Cette technique est d'un grand intérêt pour la communauté de la fouille de données où plusieurs recherches ont été menées afin de développer de nouveaux algorithmes permettant, à la fois, de découvrir et d'extraire de nouvelles relations entre un grand nombre d'attributs.

Dans ce chapitre nous introduisons une vue générale sur ce domaine. Nous commençons par présenter la problématique de l'extraction des motifs fréquents. Ensuite, nous donnons quelques définitions pour éclairer les notions de base, et enfin, nous présentons quelques exemples des algorithmes d'extraction.

1.2 DEFINITION DU PROBLEME D'EXTRACTION :

Le problème d'extraction des Itemsets fréquents a attiré beaucoup l'intention des chercheurs de la communauté de Data Mining, et par conséquent un nombre important des algorithmes a été proposédans la littérature. Dans cette partie nous allons expliquer les notions de base de cette problématique à travers un ensemble de définitions.[27]

1.2.1 Définitions

- **Item :** un Item **« xi »** est tout objet, article, attribut, ou littéral, appartenant à un ensemble fini d'éléments distincts B = {x1, x2, ..., xn} appelé aussi « ***base d'items*** » . Dans les applications de type analyse du panier de la ménagère, les articlesen vente dans un magasin sont des items.
- **Itemset** : Un Itemset est un ensemble de n Items. L'ensemble de tous les Itemsets possibles formés par les éléments d'un ensemble d'Items « B » est 2^n, tel que | B |=n.

- **Transaction** : Une transaction sur une base d'items *B* est une paire *t* = (*tid*, *J*), où, *tid* est un identificateur unique de transaction, et J est un itemset. Par exemple les items achetés par un client C à une date précise.
- **Base de données transactionnelle :** Une base de données transactionnelle

$T = \{t1, \ldots, tm\}$ est un ensemble de transactions. Une base de données transactionnel peut être représentée sous forme horizontale, verticale ou binaire. Le tableau 1.1 présente une base transactionnelle sous forme binaire (matrice de transactions) .

Transaction	**Item 1**	**Item 2**	**Item 3**
T1	0	1	1
T2	1	0	1
T3	1	0	0

Tableau 1.1 : Matrice de transactions

- **Couverture** : Une transaction t = (tid, J) couvre un itemset I ssi I ⊆ J.
- **Le support d'un Itemset** : le support d'un Itemset est le nombre de transactions couvrant cet Itemset. On peut utiliser le support absolu S_T, ou relative σ_T :

  $S_T(I) = |\{(t, Xt) \setminus I \subseteq Xt\}|$, le support absolu de l'itemset I selon la base T.

  $\sigma T\ (I) = \frac{|\{(t,Xt)|I \subseteq Xt\}|}{|T|}$, le support relatif de l'itemset I selon la base T.

- **Itemset fréquent** : Un Itemset est fréquent si et seulement si son support est supérieur ou égale à un support minimum (supp_min) défini par l'utilisateur.

Le problème d'extraction des Itemsets fréquents consiste à chercher tous les Itemsets dont le support est supérieur ou égale à un support minimum. La recherche des itemsets fréquents dans une base de données quelconque est un problème non trivial car le nombre d'itemsets potentiellement fréquents (appelés itemsets candidats) est fonction exponentielle du nombre d'items dans cette base ($2n - 1$). De plus, de nombreux balayages coûteux en temps doivent être répétés sur cette base de données afin d'évaluer les supports des itemsets candidats à chaque étape de la recherche. [2]

1.3 DOMAINES D'APPLICATION :

Avec la performance des systèmes informatiques actuels et la maturité des méthodes d'apprentissage automatique, le FIM est devenu très attrayant dans de nombreux domaines d'applications : médecine, génétique, astronomie, processus industriels, agriculture ou encore la gestion de la relation client, et la production industrielle, … etc.

Les entreprises ont mis en œuvre ces outils pour améliorer leur connaissance afin de mieux les servir et augmenter leur satisfaction et leur fidélité, pour augmenter leur rentabilité. Les principaux secteurs économiques utilisant ces techniques sont le secteur financier (banques et assurances), les télécommunications ainsi que les entreprises de grande distribution. Dans ces secteurs, massivement informatisés depuis longtemps, les données sont disponibles au sein d'entrepôts de données. On peut résumer les champs d'applications les plus importantes du FIM dans les domaines suivants : [28]

- **Scoring :** En marketing, le scoring consiste à affecter une note à un client ou un prospect. Le but est de déterminer le profil du client par rapport à l'activité de l'entreprise, et ainsi réduire le coût d'acquisition ou de conservation d'un client, en ciblant les opérations marketing sur les profils considérés les plus "réceptifs". Le scoring est par exemple utilisé chez les assurances, les banques ou encore les opérateurs téléphoniques. (Ex : ne pas accorder un prêt à un client qui présente un profil reconnu par le FIM comme présentant un haut risque de non remboursement.)[28]

  Le FIM peut par exemple être utilisé pour déterminer quels sont les critères à prendre en compte pour considérer un client comme "réceptif".

- **La recherche scientifique et médicale :** Le FIM fournit aux établissements hospitaliers des solutions et des services pour leur permettre de mieux connaître les comportements sanitaires et les pathologies rencontrées à savoir : le diagnostic médical, l'état des lieux des comportements en matière de santé, l'analyse des risques sanitaires, l'étude de traitements de thérapies, ainsi que les différentes études en milieu hospitalier telle que la génomique, le code génétique, …etc. [29]
- **Le domaine financier (Banques et Assurances) :** grâce au FIM, un organisme financier peut déterminer le profil exact de ces clients afin de cibler ceux de même profil (mailing). D'autres applications peuvent y avoir telles que la gestion et calcul du risque client, l'analyse des sinistres, l'assistance au recouvrement en orientant la bonne démarche, la recherche de fraudes, la recherche des corrélations entre les indicateurs financiers, le retour sur investissement de portef euilles d'actions.
- **Détection de fraudes :** Dans les systèmes complexes gérant un nombre d'utilisateurs importants (les administrations par exemple), un problème se pose fréquemment : la fraude. Le FIM, utilise la classification sur les données. Ce mécanisme peut notamment permettre de détecter les données qui vont sortir de l'ordinaire, qui n'auront pas la même empreinte que les comportements "normaux". [3]

Certains comportements "normaux" peuvent également sortir de l'ordinaire et constitueront des faux positifs dans le cas de la détection de la fraude, mais c'est une méthode qui permettra de faire ressortir les cas à surveiller.[29]

- **Laboratoires pharmaceutiques et cosmétiques** : Le FIM fournit aux laboratoires pharmaceutiques et de cosmétologie des solutions et des services pour leur permettre à la fois de mieux connaître leur cœur de cible et d'améliorer les procédés de fabrication, de s'assurer de la qualité de leurs produits et d'évaluer leur potentiel de commercialisation. [29]
- **La gestion de relation client (CRM Customer Relationship Management) :** C'est le domaine principal où le FIM a prouvé son efficacité. En effet, dans ce cas, le FIM permet d'accroître les ventes par une meilleure connaissance de la clientèle. Dans un contexte concurrentiel de plus en plus soutenu, la capacité à conquérir et à retenir les clients repose sur une connaissance fine de leurs besoins et de leur comportement. Les objectifs des analyses en FIM sont multiples, tels que la fidélisation, les ventes additionnelles et croisées, l'efficacité de la force de vente, la personnalisation de l'offre, le contact client, l'enquête de satisfaction des clients, …etc. [29]

1.4 TYPE DE DONNEE POUR L'EXTRACTION DES ITEMSETS :

En principe, le FIM peut s'appliquer à tous les types de données. Toutefois, selon chaque type de données, les algorithmes de la fouille de données diffèrent. Quelques exemples de types de données auxquels peut s'appliquer le FIM :

A. Base de données relationnelle :

Une base de données relationnelle est une base de données consistant dans des tableaux séparés, avec des dont les éléments peuvent être combinés sélectivement comme des résultats à des interrogations. Chaque tableau contient des colonnes (correspondantes à des tuples) et des lignes (correspondantes à des attributs).[30]

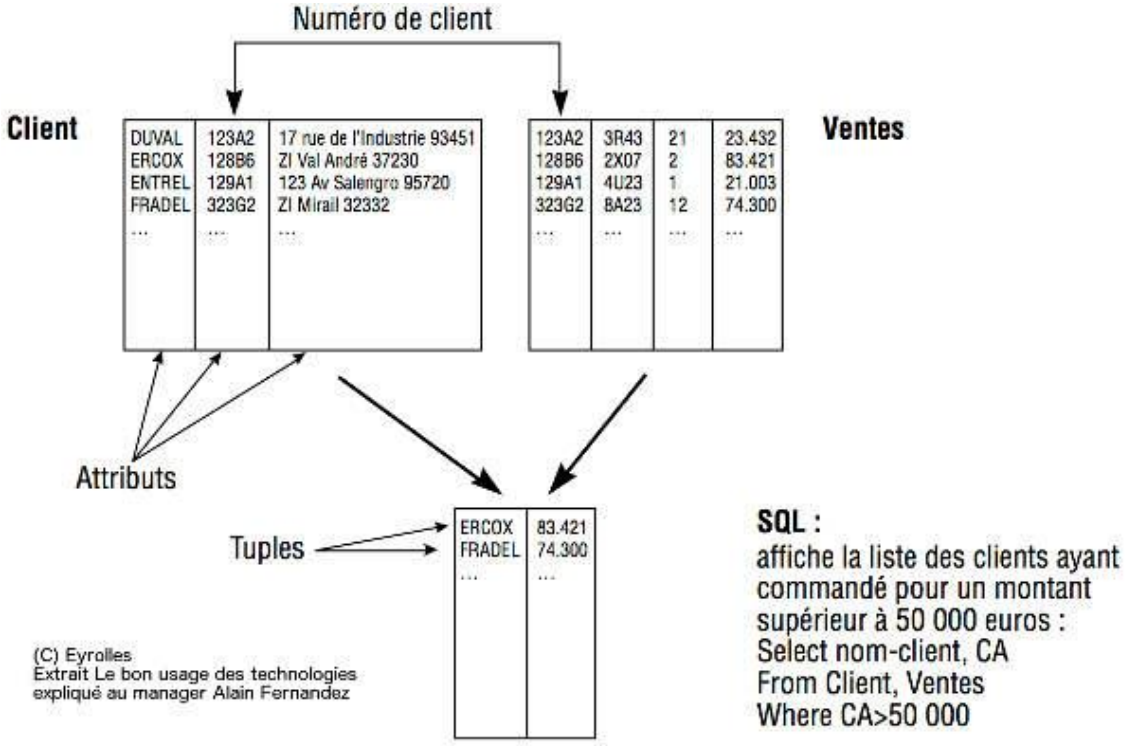

Figure 1.1 : exemple de BDD relationnelle.[30]

B. Entrepôt de données :

Ce terme désigne une base de données utilisée pour collecter et stocker des informations provenant de multiples bases de données (souvent hétérogènes) et qui les traite comme un tout-unitaire (comme une seule base de données).

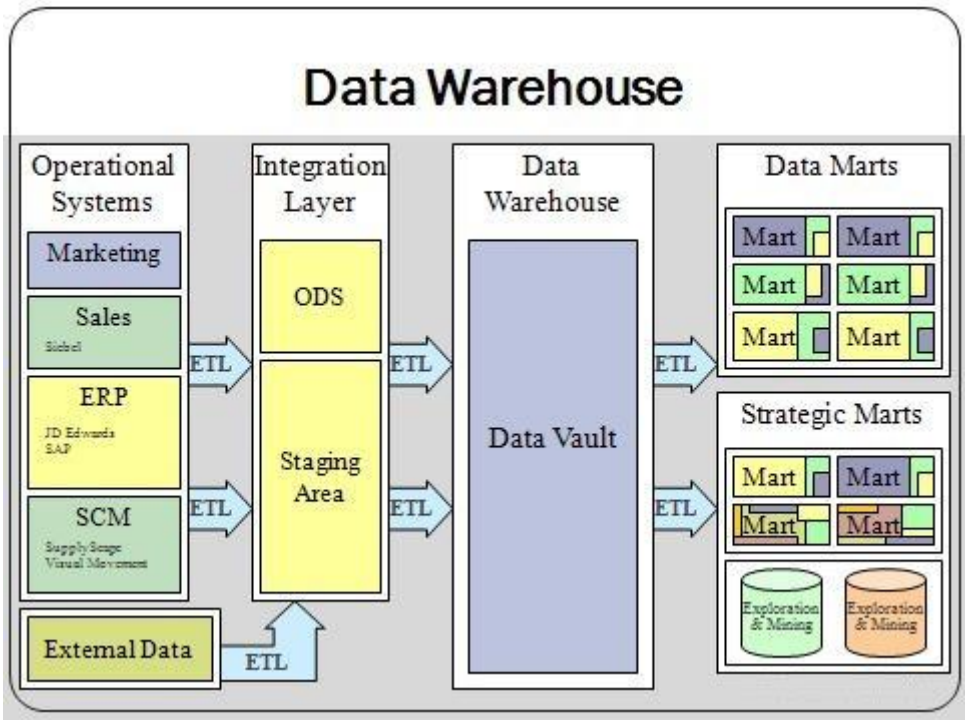

Figure 1.2 : exemple d'entrepôt de données[31]

C. Base de données transactionnelle :

Une base de données transactionnelle est un ensemble d'enregistrements représentant des transactions, chaque enregistrement ayant une marque temporelle, un identificateur et un ensemble d'articles.

D. Base de données multimédia :

Ce type de base de données inclut des vidéos, des images, des audio et des textes média ; base de données temporelles : Elles contiennent des données organisées dans le temps, comme par exemple des activités log.

E. Bases de données orientées objet et relationnelle objet :

Il s'agit d'un type spécial de base de données (ou base de données relationnelle) où les données sont des objets.

F. **Bases de données spatiales :**

Une telle base de données est optimisée afin de garder des données spatiales et de pouvoir en être interroger.

1.5 ALGORITHMES D'EXTRACTIONS DES ITEMSETS FREQUENTS :

L'extraction de connaissances dans les bases de données, également appelé data mining, désigne le processus permettant d'extraire des informations et des connaissances utiles qui sont enfouies dans les bases de données, les entrepôts de données (data warehouse) ou autres sources de donnée. Depuis sa création, le Data Analytique joue un rôle important dans le processus de prise de décision, du coup plusieurs algorithmes FPM ont été développés pour accélérer les performances d'extraction. Durant cette partie du mémoire nous allons les diffèrent algorithme de FIM, affin relever les forces et les faiblesses des algorithmes FIM.,

1.5.1 Quelques algorithmes :

1.5.1.1 L'algorithme Apriori :

1.5.1.1.1 Principe :

L'algorithme Apriori était le premier algorithme proposé pour l'extraction fréquente d'ensembles d'éléments. Il a ensuite été amélioré par R Agarwal et R Srikant[2] et est devenu connu sous le nom d'Apriori. Cet algorithme utilise deux étapes «joindre» et «élaguer» pour réduire l'espace de recherche. Il s'agit d'une approche itérative pour découvrir les ensembles d'éléments les plus fréquents. [32]

La jointure : l'union de tous les itemsets n'ayant qu'un seul élément différent.

L'élagage : Seuls les itemsets dont tous les sous-ensembles sont fréquents sont conservés

1.5.1.1.2 Les étapes de l'algorithme Apriori :

a. Dans la première itération de l'algorithme, chaque élément est considéré comme un candidat d'ensembles à 1 éléments. L'algorithme comptera les occurrences de chaque élément.

b. Soit un support minimum, min_sup (par exemple 2). L'ensemble des 1 - itemsets dont l'occurrence satisfait le min sup est déterminé. Seuls les candidats qui comptent plus ou égal à min_sup, sont pris en avance pour l'itération suivante et les autres sont élagués.

c. Ensuite, les éléments fréquents de 2 éléments avec min_sup sont découverts. Pour cela, dans l'étape de jointure, l'ensemble de 2 éléments est généré en formant un groupe de 2 en combinant des éléments avec lui-même.

d. Les candidats à 2 éléments sont élagués en utilisant la valeur de seuil min-sup. Désormais, la table aura 2 ensembles d'éléments avec min-sup uniquement.

e. L'itération suivante formera 3 ensembles d'éléments en utilisant l'étape de jointure et d'élagage. Cette itération suivra la propriété antimonotone où les sous-ensembles d'ensembles de 3 éléments, c'est-à-dire les sous-ensembles de 2 éléments de chaque groupe, tombent dans min_sup. Si tous les sous-ensembles de 2 éléments sont fréquents, le sur-ensemble sera fréquent sinon il sera élagué.

f. La prochaine étape suivra la création d'un ensemble de 4 éléments en joignant l'ensemble de 3 éléments à lui-même et en l'élaguant si son sous-ensemble ne répond pas aux critères min_sup. L'algorithme est arrêté lorsque l'ensemble d'éléments le plus fréquent est atteint. (voir figure1.3)[32]

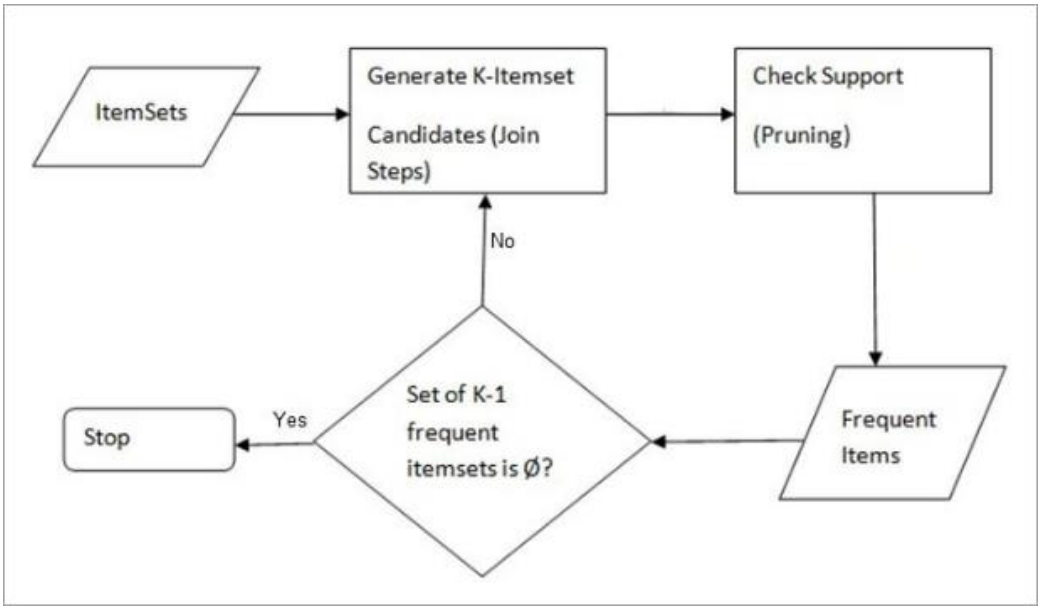

Figure 1.3 principe d'algorithme Aprioiri [32]

1.5.1.1.3 Exemple :

Soit D la base de transactions suivante : Appliquer l'algorithme Apriori pour extraire les itemsets fréquents avec minSupp = 2

TID	Itmest
001	I1 I2 I4
002	I2 I3 I5

003	**I1 I2 I3 I5**
004	**I2 I5**

1. C1 = {I1 : 2, I2 : 3, I3 : 3, I4 : 1, I5 : 3}
F1 = {I1 : 2, I2 : 3, I3 : 3, I5 : 3}
2. C2 = {I1I2 : 1, I1I3 : 2, I1I5 : 1, I2I3 : 2, I2I5 : 3, I3I5 : 2}
F2 = {I1I3 : 2, I2I3 : 2, I2I5 : 3, I3I5 : 2}
3. C3 = {~~I1I2I3~~, ~~I1I3I5~~, I2I3I5 : 2}
F3 = {I2I3I5 : 2}
4. Solution: F = {F1 ∪ F2 ∪ F3}

Tableau1. 2 : Exemple de base transaction

1.5.1.1.4 L'Algorithme Apriori (pseudo-code)

```
function apriori (B,T,smin)
 begin                                  (∗ – Apriori algorithm ∗)
 k := 1 ;                               // Initialiser la taille du Itemset
Ck := S i∈B{{i}};                       // Commencez par un seul élément
Fk := prune(Ck ,T,smin);                    // et déterminer les items fréquents
    while Fk 6= ∅ do begin                  // tant qu'il y a des itemsets fréquents.
Ck+1 := candidates(Fk );                    // créer des candidats avec 1 élément de +
Fk+1 := prune(Ck+1,T,smin);             // et déterminez les itemsets fréquents.
k := k + 1 ;                            // incrémenter le compteur d'items
end ;
return Sk j=1 Fj ;                          // return les itemsets frequents      end
(∗ apriori ∗)
```

Cj : itemsets candidats de taille j, Fj : itemsets frequents de taille j.

Figure1.4 : Algorithme Apriori

1.5.1.1.5 *Les avantage et les inconvénients :*

- **AVANTAGES**
  Algorithme facile à comprendre
  Les étapes de jointure et d'élagage sont faciles à implémenter sur de grands ensembles d'éléments dans de grandes bases de données

- **INCONVENIENTS**
  Cela nécessite un calcul élevé si les ensembles d'éléments sont très volumineux et que la prise en charge minimale est maintenue très faible.

La base de données entière doit être analysée.

1.5.1.2 Frequent Pattern Growth (FP-GROWTH):

1.5.1.2.1 Principe *:*

FP-Growth (Frequent-Pattern Growth), a été considéré comme l'algorithme le plus performant par rapport aux autres algorithmes pour extraire des itemsets fréquents.
L'algorithme consiste d'abord à compresser la base de données en une structure compacte appelée FP-tree (Frequent Pattern tree) et qui apporte une solution au problème de la fouille de motifs fréquents dans une grande base de données transactionnelle.
Contrairement aux techniques mentionnées précédemment, l'algorithme FP-Growth ne repose sur aucune approche de génération de motifs candidats. [33]

1.5.1.2.2 La structure FP-GROWTH :

L'algorithme FP-Growth effectue deux passes (scans) à la base de transactions :

— **Passe 1** le premier passage de FP-Growth sur la base de données D est consacré à déterminer la valeur du support de chaque item dans D. L'algorithme ne retient que les éléments fréquents dans une liste F-List. Ensuite, FP-Growth trie F-List dans un ordre décroissant en fonction du valeur de support et qui est comparé avec le seuil de support préfixé (MinSup).
— **Passe 2** un FP-Tree est construit par la création d'une racine vide et un second parcours de la base de données où chaque transaction est décrite dans l'ordre des items donné par la liste F-List. Chaque nœud de l'arbre FP-Tree représente un élément dans L et chaque nœud est associé à un compteur (c'est-à-dire, compte de support initialisé à 1). Si une transaction partage un préfixe commun avec une autre transaction, le compte de support de chaque nœud visité est incrémenté de 1. Pour faciliter la traversée de FP-Tree, une table d'entête est construite pour que chaque élément pointe vers ses occurrences dans l'arbre via une chaîne de liens-nœuds.[33]

En dernier lieu, le FP-Tree est fouillé par la création des (sub-)fragments conditionnels de base. En fait, pour trouver ces fragments, on extrait pour chaque fragment de longueur 1 (suffix pattern) l'ensemble des préfixes existant dans le chemin du FP-Tree (conditional pattern base). L'itemset fréquent est obtenu par la concaténation du suffixe avec les fragments fréquents extraits des FP-Tree conditionnels.[8]

Tableau1. 3 Exemple de base de transactions D

Tid	Items
1	1 2 5 6
2	2 4 7 1
3	2 1 3 4
4	6 2 4 7

Items	Support
2	**4**
1	**3**
4	**2**
6	**2**
7	**2**
3	**1**
5	**1**

Tableau1. 4 Items associés à leur support

1.5.1.2.3 Construction de l'arbre FP-Tree :

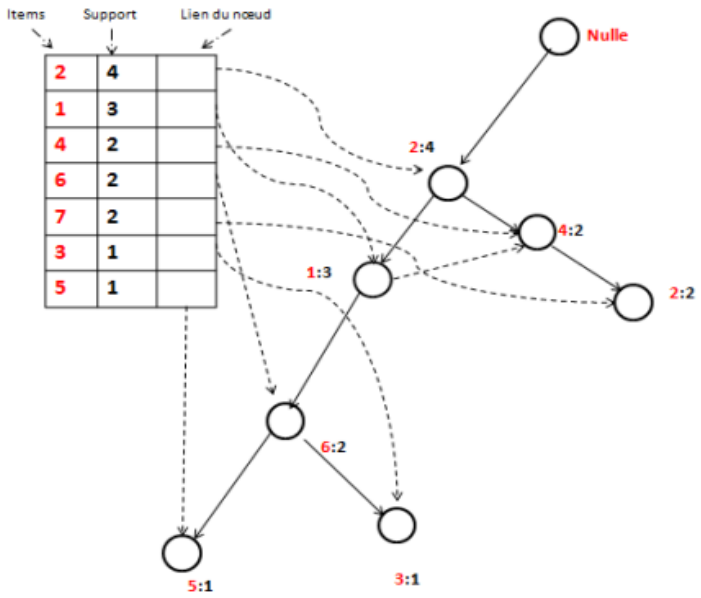

Figure 1.5 Construction du FP-Tree

En dernier lieu, le FP-Tree est fouillé par la création des (sub-)fragments conditionnels de base. En fait, pour trouver ces fragments, on extrait pour chaque fragment de longueur 1 (suffix pattern) l'ensemble des préfixes existant dans le chemin du FP-Tree (conditional pattern base). L'itemset fréquent est obtenu par la concaténation du suffixe avec les fragments fréquents extraits des FP-Tree conditionnels (voir tableau 5)

Itmes	Motifs conditionnels	Fr-tree conditionnels	Motifs fréquents
5	1,2,6,2,1,3	(2 :2 , 1 :1)	2,5 :2 ,1 ,5 :2,2 ,1,5 :2
4	2 :1 ,2,1 :1	(2 :2)	2,4 :2

Tableau 1.5 fouille de fp tree

1.5.1.2.4 *Appel de la méthode : FP_growth(FP_tree, null) :*

```
procedure      FP_growth(Tree, α)
 if Tree contient un seul chemin de préfixe P then
        foreach combinaison β des nœuds du chemin P do
         générer l'itemset fréquent (βUα) avec supp = min_supp des nœuds de
β ;
         end
        end
else
        foreach ai dans la table d'en-têtes do
        générer l'itemset fréquent (β = aiUα) avec supp = ai .supp ; construire
        la "Conditional Pattern Base" de β, et l' FP-tree conditionnelle de β :
        Treeβ ;
                if Treeβ 6= ∅ then
         Appel FP_growth(Treeβ,β )
                end
            end
 end
```

Figure 1.6 : psedo-code FP-GROWTH

1.5.1.2.5 Avantages de l'algorithme et les inconvénients :

Avantages de l'algorithme

1. Cet algorithme n'a besoin d'analyser la base de données que deux fois par rapport à Apriori qui analyse les transactions pour chaque itération.
2. L'élagage des éléments n'est pas effectué dans cet algorithme et cela le rend plus rapide.
3. La base de données est stockée dans une version compacte en mémoire.
4. Il est efficace et évolutif pour l'extraction de modèles fréquents longs et courts.

Inconvénients de l'algorithme

1. FP Tree est plus encombrant et difficile à construire qu'Apriori.
2. Cela peut coûter cher.
3. Lorsque la base de données est volumineuse, l'algorithme peut ne pas tenir dans la mémoire partagée.

1.5.1.3 L'algorithme ECLAT (Equivalence Class Clustring and Bottom Up Lattice Transversal) :

1.5.1.3.1 Principe:

ECLAT (Equivalence Class Transformation) a été introduit par Zaki, Parthasarathy, Ogihara et Li[3] Eclat a été conçu pour surmonter les inconvénients de algorithme Apriori. Il utilise la mémoire agrégée du système en partitionnant les candidats en ensembles disjoints à l'aide du partitionnement par classe d'équivalence. Il dissocie la dépendance entre les transformateurs en droit en commençant de sorte que le coût de redistribution puisse être amorti par les itérations ultérieures. Eclat utilise la structure de base de données verticale qui regroupe toutes les informations pertinentes dans la liste des objets.

Il utilise algorithme de recherche en profondeur d'abord (Depth-First Search) et la base de données n'a pas besoin d'être scannée plusieurs fois pour que identifier les éléments (k + 1).La base de données est analysée une seule fois pour transformer les données du format horizontal dans le format vertical.

1.5.1.3.2 Procédure *:*

Eclat est composé de trois phases principales

- La phase d'initialisation : construction globale des 2-itemsets.
- La phase de transformation : partitionnement de l'ensemble des 2-itemsets fréquents etde ces partitions aux autres processeurs. Transformation verticale de la base.

La phase asynchrone : construction des k-itemsets fréquents.

1.5.1.3.3 L'Algorithme *:*

input : R: a set of itemsets with their tidsets, $minsup$: a user-specified threshold
output: the set of frequent itemsets

```
1  foreach itemset X ∈ R such that |tid(X)| ≥ minsup do
2      Output X;                                          // X is a frequent itemset
3      E = ∅;                          // frequent itemsets that are extensions of X
4      foreach itemset Y ∈ R sharing all but the last item with X do
5          tid(X ∪ Y) = tid(X) ∩ tid(Y);          // calculate the tidset of X ∪ Y
6          if |tid(X ∪ Y)| ≥ minsup then                  // if X ∪ Y is frequent
7              E = E ∪ {X ∪ Y};          // add X ∪ Y to frequent extensions of X
8          end
9          Eclat (E, minsup);                             // recursive call using E
10     end
11 end
```

Figure 1.7 ; Equivalence class transformation (eclat)[36]

1.5.1.3.4 Avantages et inconvénients :

Analyser la base de données pour trouver le nombre de supports de (k + 1) éléments n'est pas requis, mais il a besoin plus d'espace mémoire et de temps de traitement sont nécessaires pour l'intersection de longs ensembles de TID.[34]

1.5.2 Comparison des algorithms d'extraction des motifs frequents:

Heaton[4], a réalisé une étude de performance pour comparer les trois algorithmes(Apriori, Eclat, FPgrowth), en utilisant l'effet de la densité de données et l'augmentation de lataille de transaction.

1.5.2.1 Effets de la densité des données :

L'algorithme Apriori, Eclat et FP-Growth fonctionne de manière similaire, jusqu'à ce que la densité dépasse 70%, Apriori a des besoins en mémoire considérablement plus importants que les autres algorithmes. À 70%, Eclat et FP-Growth affichent tous deux une croissance très similaire mais Apriori avait alloué la totalité de RAM de la machine de test. Cela a rendu nécessaire l'échange de stockage physique et a eu un impact considérable sur

le temps d'exécution de l'algorithme. Il est également intéressant de noter qu'Eclat est légèrement en avance sur FP-Growth à faibles densités. Comme le montre la figure 1.8

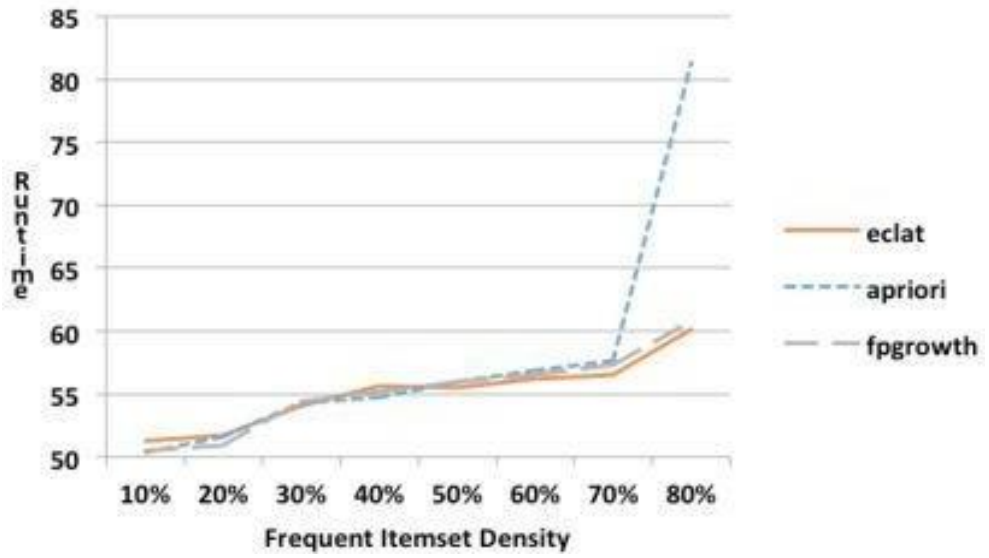

Figure 1.8 : Effet de la densité d'esmeble d'article sur le temps d'exécution[34]

1.5.2.2 Effets l'augmentation de la taille des transactions :

Les trois algorithmes montrent presque exactement les mêmes performances pour des tailles allant jusqu'à 60. Une fois supérieur à 60, Apriori semble croître beaucoup plus vite queles deux autres. Ceci est probablement dû à la mémoire accrue utilisée par Apriori. Fait intéressant, Apriori a réalisé le meilleur résultat entre 60 et 70 tailles de transaction maximales.Comme le montre la figure 1.9.

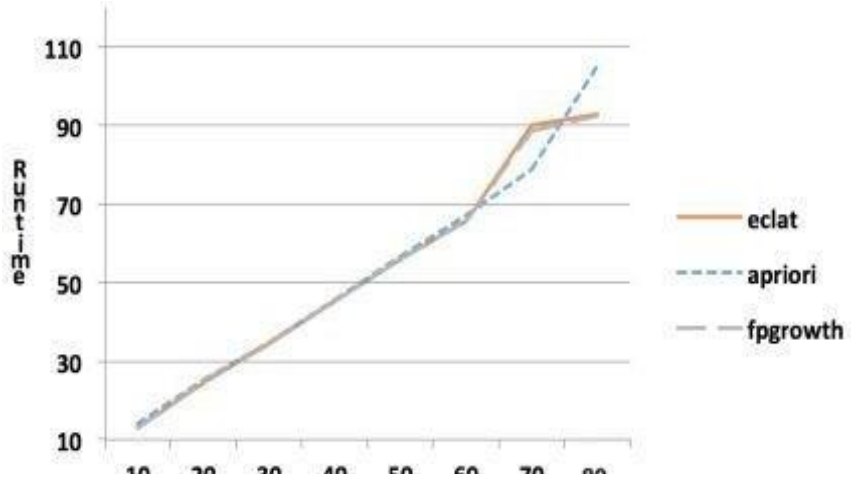

Figure 1.9.. : Effet de taille de transaction maximale sur le temps d'exécution (secondes)

1.6 CONCLUSION:

Dans ce chapitre, nous vous avons donné un bref aperçu des principaux algorithmes d'extraction des motifs fréquents tel qu'Apriori, FP-Grouth et Eclat L'objectif de ce chapitre est de passer en revue les forces et les faiblesses des algorithmes fondamentaux dans la FIM.

CHAPITRE II :

RECONSTITUTION DES DATA-SETS A BASE DE ITEMSET FREQUENTS

II. CHAPITRE II : RECONSTITUTION DES DATA-SETS A BASE DE ITEMSET FREQUENTS :

2.1 INTRODUCTION :

Le processus d'extraction d'items fréquents (FIM) dans des bases de données à grande échelle joue un rôle important dans de nombreuses tâches de découverte de connaissances, où, cependant, des violations potentielles de la vie privée sont possibles. L'extraction d'items fréquents préservant la confidentialité des données (PPFIM) a donc attiré une attention croissante récemment, où le but ultime est de cacher les items fréquents sensibles (SFIM) afin de ne laisser aucune connaissance confidentielle dans la base de données résultante.

2.2 DEFINITION DE LA PROBLEMATIQUE DE RECONSTRUCTION :

La reconstruction de base de données a pour but de créer une base de données cible avec des éléments fréquents donnés. Plus précisément, une méthode IFIM est utilisé pour reconstruire une base de données, où tous les items fréquents d'entrée peuvent être distingués des items peu fréquents. Ensuite, un traitement est fait pour ajuster les fréquences des items existants dans la base de données reconstruite afin de générer une base de données résultante, où les items fréquents identiques peuvent être extraits sous le seuil de fréquence donné.

2.3 RECONSTITUTION AVEC PROTECTION DE LA VIE *PRIVEE* :

Le problème de dissimulation des SFIM peut être formulé comme suit :Étant donné une base de données de transactions D, un seuil de support minimum "MST", un seuil de confiance minimum "MCT", un ensemble de règles d'association significatives R extraites de D et un ensemble de règles sensibles à masquer, trouver une nouvelle base de données D', telle que si les à partir d'une base de données réelle D, trouver une base de données D' satisfaisant aux contraintes suivantes :

- D’est sur le même ensemble d'éléments I.

- À partir de D', nous pouvons découvrir exactement le même ensemble d'objets fréquents F avec le même ensemble de support S sous le même seuil de support minimum MST.

2.4 LES APPROCHE DE LA RECONSTRUCTION :

Les taches de data mining ont pour but de dévoiler des données implicites provenant de larges bases de données, mais des itemsets fréquents qui contiennent des données sensibles risquent d’être divulguées[5][6], des informations publiques ou confidentielles doivent être caché avant que les bases de données soient partagées avec des collaborateurs ou soient publiées publiquement[7], il est donc requis de trouver une solution qui puisse miner autant d’information d’intérêt tout en ne compromettant aucun SFIM.

2.4.1 Approches de la modification des données :

Les méthodes de modification des données masquent les règles d'association sensibles en modifiant directement les données originales. La plupart des premières méthodes appartiennent à cette catégorie. Selon les différents moyens de modification, on peut encore les classer en deux sous-catégories : Les techniques de distorsion des données et les techniques de blocage des données.

2.4.2 La distorsion des données :

Est basée sur la perturbation ou la transformation des données, et en particulier, la procédure consiste à changer un ensemble sélectionné de valeurs 1 en valeurs 0 (supprimer des éléments) ou de valeurs 0 en valeurs 1 (ajouter des éléments) si l'on considère la base de données des transactions Comme une matrice bidimensionnelle. Son objectif est de réduire le support ou la confiance des règles sensibles en dessous du seuil de sécurité prédéfini par l'utilisateur.

2.4.3 Le blocage des données :

Le blocage des données est une autre approche de modification des données pour le masquage des règles d'association. Au lieu de déformer les données (une partie des données est modifiée pour devenir fausse), l'approche de blocage est mise en œuvre en remplaçant certains éléments de données par un point d'interrogation " ?". L'introduction de cette valeur inconnue spéciale apporte de l'incertitude aux données, ce qui fait que le support et la confiance d'une règle d'association deviennent respectivement deux intervalles incertains. Au début, les limites inférieures des intervalles sont égales aux limites supérieures. Au fur et à mesure que le nombre de " ?" dans les données augmente, les limites inférieures et

supérieures commencent à se séparer progressivement et l'incertitude des règles augmente en conséquence. Lorsque l'une ou l'autre des limites inférieures de l'intervalle de soutien et de l'intervalle de confiance d'une règle passe en dessous du seuil de sécurité, la règle est considérée comme cachée.

2.4.4 Les approche base sur la perturbation de donnée :

La méthode la plus couramment utilisée aujourd'hui, et qui a été héritée de la majorité des précédentes approches proposées est d'exécuter des opérations de perturbation sur un ou plusieurs items dans la base de données originale. [8] ont été les premiers à proposer une solution complète à ce problème en modifiant les items de sorte à réduire la fréquence de SFIM, et leur solution a été améliorée par l 'intégration d'un ensemble complet d'algorithmes, d'expériences, d'analyses et d'évaluation de résultats par [9]. Les travaux réalisés par [5][6]. Ont été les premiers à présenter une solution heuristique complète au problème en modifiant les articles pour réduire la fréquence des SFIM [8], et ce travail a été complété plus tard par un ensemble complet d'algorithmes, d'expériences, d'analyses et d'évaluations des résultats par [9]. Les travaux développés par [6].

Ont introduit un cadre algorithmique rapide pour cacher des connaissances sensibles en s'appuyant sur un moteur de recherche de transactions, et ont proposé plusieurs algorithmes tels que Naïve,MFIA , MaxFIA, , IGA[10] , et d'autre part,[11].

Ont cherché à éviter les effets secondaires au lieu de cacher toutes les SFIM en classant les modifications valides par trois attributs, qui indiquent respectivement le schéma de modification, l'ensemble des éléments et la transaction à modifier. Le concept de frontière a été proposé pour la première fois par [12], sur la base duquel [13]ont conçu l'algorithme Max-Min qui sélectionne de manière préférentielle les éléments dits Max-Min pour la perturbation de sorte que l'impact possible minimal sur la frontière puisse être garanti [13]. Contrairement aux approches heuristiques, [14] ont présenté une approche innovante pour le PPFIM, dans laquelle la tâche de dissimulation qui tente d'atteindre l'optimisation globale est formulée comme un CSP [14]. Sur la base de cette idée,[15] ont ensuite utilisé la théorie de la révision des frontières pour déterminer un ensemble minimal d'items pour garantir une dissimulation efficace. Sans effets secondaires [15].

Bien que les approches susmentionnées basées sur la perturbation aient bien réussi à cacher les SFI, l'effet de conservation de l'utilité des données de la base de données résultante n'était cependant pas satisfaisant dans certaines circonstances. Comme l'opération de perturbation ne peut s'exécuter que sur la base de données originale, certaines contradictions entre la protection des connaissances privées et la préservation de l'utilité des données sont inévitables. D'extraction [16] Dans leur stratégie, une procédure d'assainissement a d'abord été effectuée pour éliminer les connaissances sensibles des résultats d'extraction originaux en supprimant directement tous les SFIM. Ensuite, ils ont utilisé l'algorithme IFIM basé sur l'arbre FP pour générer rapidement une certaine quantité de bases de données différentes et

compatibles grâce à l'arbre FP. Qui a été créé après plusieurs procédures. Néanmoins, leur algorithme souffrait d'un grave défaut le nombre de transactions dans la base de données générée était de 1 000.

2.4.5 Approches de reconstruction des données :

La reconstruction de la base de données fournit une nouvelle pensée pour préserver la vie privée dans la FIM, puisqu'une base de données peut être partiellement ou entièrement synthétique selon les exigences de la protection des SFIM et du maintien de l'utilité des données. Bien que cette orientation de recherche n'ait pas été suffisamment étudiée, comme le point de vue des auteurs dans [17], il s'agit d'une méthode particulièrement intéressante qui mérite d'être explorée dans la recherche ultérieure. [18] ont proposé un algorithme basé sur l'extension de la base de données pour le PPFIM, où la base de données partielle a été construite artificiellement pour réduire la fréquence des SFI tout en assurant l'intégrité des NSFI autant que possible [18]. Dans leur algorithme, la taille minimale de l'extension a d'abord été calculée en fonction du support le plus élevé parmi les SFIMs. En utilisant le concept de frontière, seuls les itemsets de la frontière ont été inclus dans l'établissement de la CSP de sorte que l'échelle du problème de dissimulation puisse être minimisée.

Ensuite, l'ensemble de la CSP a été mis en correspondance avec un problème équivalent de programmation binaire en nombres entiers, que l'on a essayé de résoudre pour élaborer la partie de la base de données étendue. Dans le cas où une solution n'était pas réalisable, certaines contraintes de la CSP ont été sélectivement abandonnées jusqu'à ce que le problème soit résoluble, ce qui, cependant, a également entraîné des sacrifices inévitables sur l'utilité des données. [19] ont présenté un schéma de reconstruction de base de données complet pour PPFIM, dans lequel un algorithme de fouille de fréquents inversés (IFIM) basé sur les FP-trees a été mis en t œuvre [19]. En bref, le but de l'IFIM est de créer une nouvelle base de données avec des items fréquents et un seuil d'extraction donnés, où les mêmes items fréquents peuvent toujours être extraits avec le même seuil incontrôlable. Par conséquent, les fréquences des ensembles, qui ont été utilisées pour déterminer si les ensembles ont de la valeur, étaient également imprévisibles dans la base de données résultante. Dans ce cas, les connaissances confidentielles peuvent encore être complètement exposées à l'adversaire, et certaines informations utiles seront perdues.

Les méthodes de reconstruction des données mettent de côté les données originales et commencent par nettoyer ce qu'on appelle la "base de connaissances". Les nouvelles données publiées sont ensuite reconstruites à partir de la base de connaissances nettoyée. En outre, l'extraction inverse d'ensembles fréquents, qui consiste à déduire les données d'origine à partir d'ensembles fréquents donnés, est un sujet émergent dans le partage de données préservant la confidentialité.[20]a été le premier à proposer ce problème. Il a montré que trouver un ensemble de données compatible avec une collection donnée d'ensembles fréquents est NP- complet. Après cela, plusieurs méthodes. Ont été proposé. Le règlement

du problème de l'extraction d'ensembles fréquents inversés apportera un soutien important au masquage de FIM basé sur la reconstruction [20].

2.5 CONCLUSION :

Dans Ce chapitre, nous proposons une approche pour cacher les SFI dans les bases de données transactionnelles, où le schéma de reconstruction de la base de données est employé. L'approche proposé supprime d'abord tous les ensembles d'éléments fréquents qui peuvent révéler des informations sensibles des résultats de l'extraction dans le processus de pré-nettoyage. Un algorithme IFIM basé sur les arbres FP et le concept d'extension sont ensuite appliqués et incorporés pour faciliter la reconstruction d'une base de données.

CHAPITRE III : IMPLEMENTATION ET EXPIRIMENTATION

III. CHAPITRE II:IMPLIMENTATION ET EXPIRIMENTATION

3.1 INTRODUCTION :

La reconstruction de base de données à base d'itemsets fréquent et un domaine relativement jeune et qui est en cours de développement, il attire beaucoup d'attention de la communauté des chercheurs en datamining.

Dans ce qui suit nous allons traiter ce problème en proposant un simple algorithme de reconstitution. Cet algorithme est utilisé dans une approche plus générale pour implémenter une méthode de protection des itemsets fréquents sensibles

3.2 DEFINITION DU PROBLEME :

Notre problématique est défini comme suit :

Etant donné une base transactionnelle B, l'ensemble des itemsets fréquents FIM extrait de B. L'ensemble FIM est divisé en 2 ensembles disjoints :

FIM ={NSFIM} + {SFIM}, tel que, l'ensemble NSFIM représente les itemsets fréquents non sensibles, et l'ensemble SFIM et l'ensemble des itemsets fréquents sensibles.

Le problème de reconstruction avec protection des itemset fréquents sensibles consiste à définir une base transactionnelle B' à partir de l'ensemble « FIM – SFIM » de telle sorte que l'ensemble des itemsets fréquents FIM' extrait de la nouvelle base soit : FIM'= NSFIM.

3.3 L'APPROCHE IMPLIMENTE :

L'approche implémentée dans notre travail est divisée en trois parties principales (voir figure 3.1) :

- **Partie 1** : elle consiste a appliquer un algorithme de FIM sur la base de donnée initiale, dans notre cas nous avons utilisé l'algorithme FP-GROWTH.
- **Partie 2** : à partir des FIM trouvés précédemment, nous allons supprimer une partie que l'on nomme SFIM. (Les itemsets fréquent sensibles), les itmesets restants représentent l'ensemble NSFIM (les itemsets fréquents non sensibles).
- **Partie 3** : l'algorithme de reconstitution va reconstruire une nouvelle base de données à partie des NSFIM engendrés dans la partie 2.

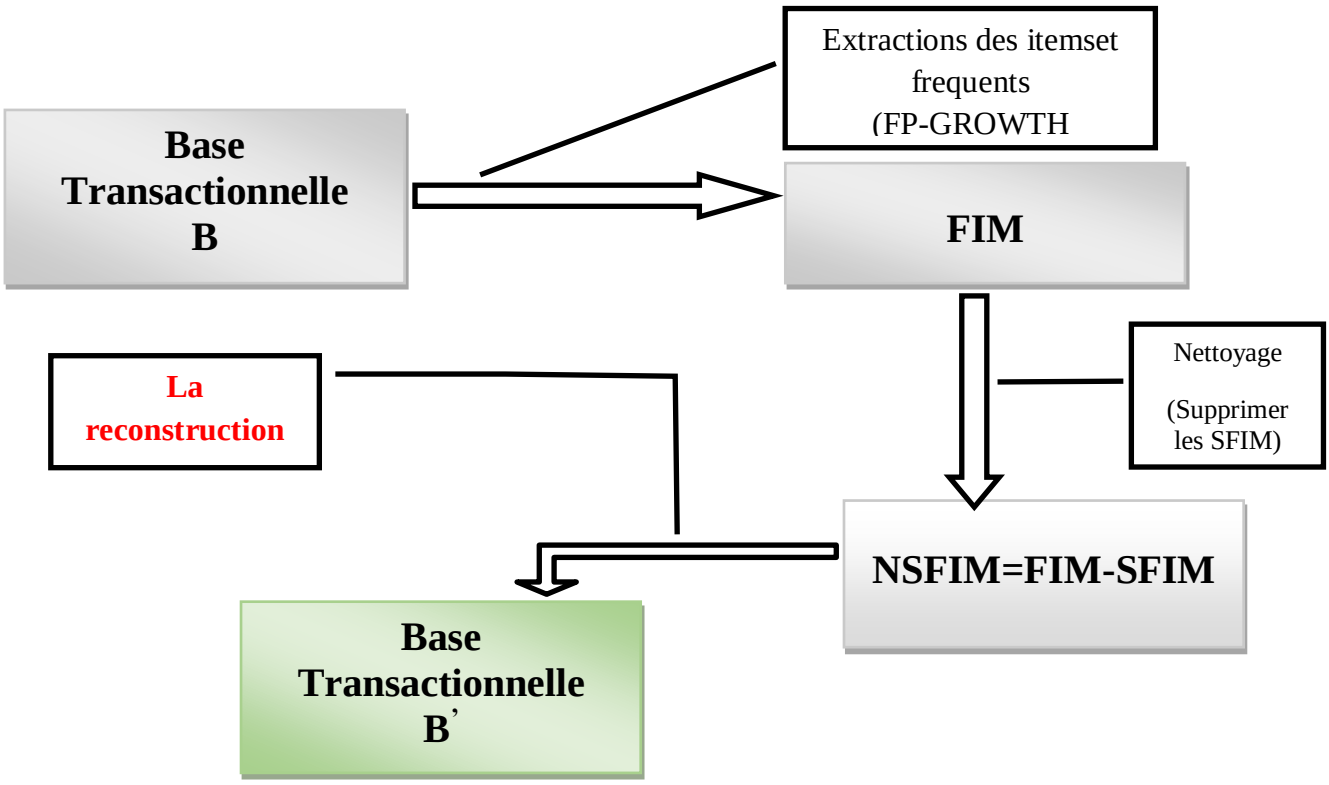

Figure 3.1: principe de fonctionnement de l'algorithme

- Le listing suivant présente les étapes détaillé sur notre implémentation :

Entré:
Dataset B: la base de données originale qui comprend les NFIM et SFIM
min_support : le support minimum
Sortie:
Dataset B': la base reconstituée (new_dataset).
Debut
1-Réorganiser Dataset **B** souforme de représentation binaire ;
2-Application de l'algorithme FP-Growth sur B pour trouver les FIM ;
3-Stockage des FIM avec leurs super-sets et leurs support ;
4-Recherche les FIM Closed fréquents C et leurs supports ;
5- Définition de la liste « ***secret*** » des SFIM ;
6-Supression des SFIM contenus dans « secret » et de leurs super-set ;
7- Reconstruction de la base B'(new_dataset) ;
8-Assainissement de new_dataset ;
9-Affichage new_dataset ;
Fin

- La partie la plus importante de notre implémentation est l'algorithme de reconstituons. Ce algorithme est représenté par le pseudo code suivant :

```
DEBUT : fonction Reconstruction (dataset, min_sup, secret)
    frequent = fpgrowth(dataset, min_sup) appel de la fonction fpgrowth
    for i jusqu'a taille(frequent) parcour de la liste frequent qui contient les FIM de dataset
        for j jusqu'a taille (secret) parcour de la liste secret qui contient les SFIM
            if secret(j) dans frequent(i)
                supprimer frequent(i)  supprimer SFIM qui sont present  dans frequent
        end
    end
    for i jusqu'a taille (frequent)
        new_dataset = frequent(i) ajou des NSFIM ainsi que leur superset dans new_dataset
    end
    return new_dataset
FIN
```

Closed représente une compression sans perte da la liste des itemsets fréquents. Ce qui nous a permis d'améliorer le temps d'exécution de notre algorithme.

3.4 DEVELOPPEMENT ET IMPLEMENTATION

Dans cette partie nous présentons premièrement l'environnement matérielle puis coté logiciel utilisé durant notre implémentation , et a la fin les différents Bibliothèques les plus utilisé .

3.4.1 Environnement Matérielle :

Un ordinateur portable dell g3 3590 avec la configuration suivante :

- 16 gb ram DDR4
- CPU: Intel core™ i7-9750h 2.5ghz,4.5 ghz ,6 core 12 thread
- GPU: NVidia GTX 1660 TI 6G GDDR5
- 512gb SSD et 1tb HDD
- Système d'exploitation : Windows 11 64bit

Configuration de la machine virtuelle colab :

- 12 de ram
- CPU : Intel core™ Xeon 2 core 4 thread
- GPU : NVidia quadro K80/T4 12G GDDR
- 80gb SSD
- Système d'exploitation : linux

3.4.2 Environnement logiciel :

Au courant dans notre implémentations nous avons utiliser

3.4.2.1 Python3:

C'est un langage de programmation dynamique de haut niveau, interprété et polyvalent, qui met l'accent sur la lisibilité du code.

La syntaxe dans Python aide les programmeurs à coder en moins d'étapes que Java ou C++ Le langage fondé en 1991 par le développeur Guido Van Rossum présente une programmation facile et amusante, Le Python est largement utilisé dans les grandes organisations en raison de ses multiples paradigmes de programmation. Ils impliquent généralement une programmation fonctionnelle impérative et orientée objet. Il possède une vaste bibliothèque standard complète dotée d'une gestion automatique de la mémoire et de fonctionnalités dynamiques. [21]

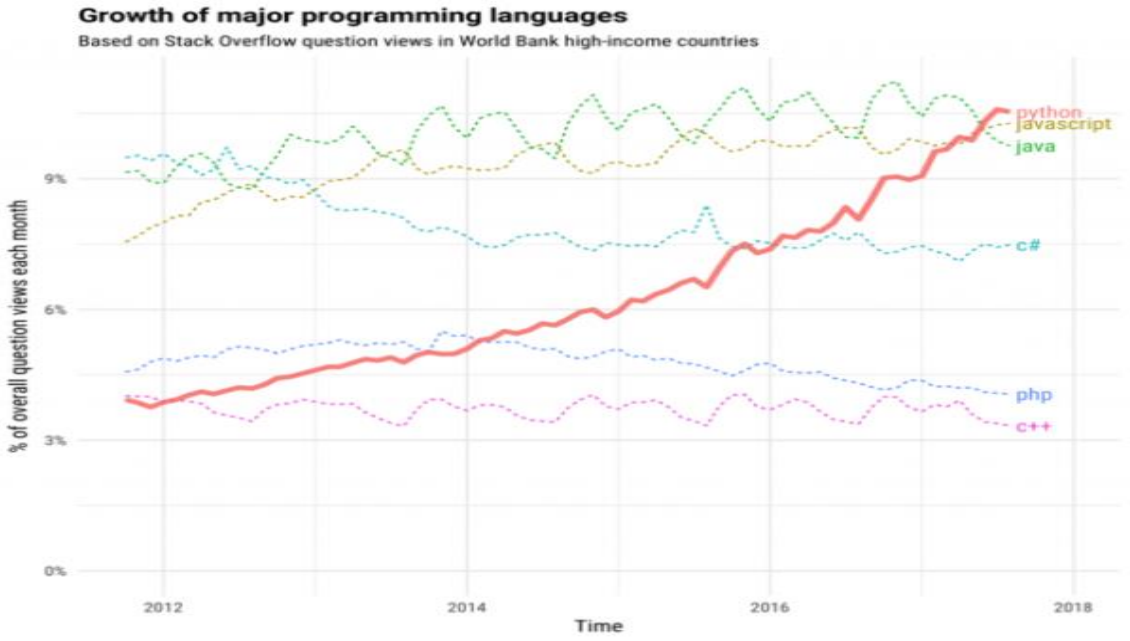

Figure 3.2 :la croissance des langages de programmation [35]

3.4.2.2 Jupyter Notebook :

Jupyter Notebook est une application client-serveur créée par l'organisation à but non lucratif Project Jupyter. Elle a été publiée en 2015. Elle permet la création et le partage de documents Webauformat JSON constitués d'une liste ordonnée de cellules d'entrées et de sorties et organisés en fonction des versions successives du document Les cellules peuvent contenir, entre autres, du code, du texte au format Markdown, des formules mathématiques ou des contenus médias (Rich Media). Le traitement se fait avec une application client fonctionnant par Internet, à laquelle on accède par les navigateurs habituels.[22]

Un Jupyter Notebook, c'est deux choses à la fois [23]

Une application Web interactive dans laquelle on peut développer, documenter, exécuter et partager du code. C'est un excellent outil notamment dans le domaine scientifique vous importez des données, vous les affichez, vous les étudiez, vous les exploitez avec des algorithmes traduits en programmes Python, il est utilisé aussi pour l'apprentissage automatique.

Un document qui permet d'intégrer en plus du code, des équations écrites en LATEX, du texte formaté en Markdown (convertit du format txt en html), différents médias (audio, vidéo) et qui s'exporte dans un certain nombre de formats (HTML, PDF, LATEX...).

3.4.2.3 Google colab :

Google Colab ou Colaboratory est un service cloud, offert par Google (gratuit), basé sur Jupyter Notebook et destiné à la formation et à la recherche dans l'apprentissage automatique[24].

3.4.2.4 Bibliothèques utiliser :

Apart les bibliothèques Pandas et Numpy et Matplotlib, etc. (voir figure 3.3),la bibliothèque qui a une relation directe avec notre travail est **MLxtend** (extensions d'apprentissage automatique). Cette bibliothèque possède de nombreuses fonctions intéressantes tel que :

- Créer un contrefactuel (pour l'interprétabilité du modèle)
- Cercle de corrélation PCA
- Décomposition biais-variance
- Régions de décision des modèles de classification
- Matrice de nuages de points
- Etc

```
[ ]  1 #Import all basic libray
     2 import pandas as pd
     3 import numpy as np
     4 from mlxtend.preprocessing import TransactionEncoder
     5 import time
     6 import csv
     7 from mlxtend.frequent_patterns import fpgrowth
     8 import matplotlib.pyplot as plt
     9
```

Figure3.3 : les bibliothèques utilisées

Nous avons utilisé 2 fonctions principales de cette bibliothèque :

- **TransactionEncoder :** C'est une fonction de Preprocessing qui permet de convertir les listes d'éléments en données transactionnelle.

```
[ ]  1
     2 # initializing the transactionEncoder
     3 te = TransactionEncoder()
     4 te_ary = te.fit(dataset).transform(dataset)
     5 df = pd.DataFrame(te_ary, columns=te.columns_)
```

Figure 3.4 : initialisation du transactionencoder

- ***Fpgrowth*** : c'est une fonction de « frequent patterns mining » elle génère les FIM a partir d'une BDD transactionnelle en se basant sur l'algorithme FP-growth.

```
]  1 # Applying one hot encoding
   2
   3 frequent = fpgrowth(df.applymap(hot_encode_inv) , min_support=0.4, use_colnames=True)
```

Figure 3.5 : appelle de la fonction fp-growth

3.4.3 Quelques Aspects de l'implémentation

Dans cette section, nous allons présenter les aspects les plus important de notre implémentation.

3.4.3.1 Importation et transformation des données

La première étape de notre implémentation est l'importons notre Dataset, la transformation de ce Dataset sous forme d'une base transactionnelle binaire.

Après l'importation, les données sont stockées dans des structures de données adéquates pour être traitées. Les figures 3.4 et 3.5 représentent respectivement la procédure de l'importation du Dataset, et la forme des données après la transformation.

```
# dataset
dataset = np.loadtxt('data.txt')
te = TransactionEncoder()
te_ary = te.fit(dataset).transform(dataset)
df = pd.DataFrame(te_ary, columns=te.columns_)

```

Figure 3.6 importation des donnés

	1040	1280	2372	2495	2872	3159	3186	4750	4995
0	True	False	True	False	True	True	False	True	False
1	True	True	False	True	True	False	True	True	True
2	True	False	False	False	True	False	False	True	False
3	False	True	False	False	True	False	True	False	True
4	True	True	False	False	True	False	True	False	True
...	...	...	...	...	...	...	...	...	...
269	True	True	True	False	True	True	True	True	True
270	True	False	False	True	True	False	False	True	True
271	False	True	False	False	True	False	True	False	True
272	True	False	False	False	True	False	False	True	False
273	True	True	False	False	False	False	False	False	False

274 rows × 9 columns

Figure3.7: Partie du data sets après transformation

3.4.3.2 Définition de la liste des SFIM

Pour faire les tests de validation, la liste des SFIM est générée aléatoirement. Cette liste peut prendre une taille qui varie entre 0 et le nombre total des FIM.

```
## LES SFIM

import random
sfim=[]
dataset
z=len(dataset)

x = random.randint(0,z)
for i in range(1,7):
 sfim.append(''.join(random.sample(dataset[x], k=i)))
sfim
# Faster and more efficient
```

Figure 3.8 : la définition de la liste SFIM.

3.5 Expérimentation :

3.5.1 Définition de la base de données utilisé :

Pour réaliser nos expérimentations, nous avons utilisé la base **data2** [25], c'est une base de données transactionnelle qui représente une fraction de la base ***kaggle*** [26], elle représente les transactions d'un panier de la ménagère. Cette base contient 274 transactions et une base d'items de 9 éléments.

3.5.2 Expérimentation 1 :

Dans cette expérimentation nous mesurons la qualité de la base produite par notre algorithme. Cette qualité est quantifiée en utilisant une mesure de similarité entre les itemsets fréquents non sensibles NSFIM de la base originale D et les itemsets fréquents de la base générée D'. Cette similarité représente le rapport du nombre des itemsets fréquents similaires entre les NSFIM de la base originale D et les FIM' de la base produite D'sur le nombre total des NFIM. Cela signifier que plus la similarité est grande plus la qualité de la base produite est meilleure. La similarité est calculée comme suit :

$$\boldsymbol{Sim(D, D') = \frac{|\{Xi | Xi \in NFIM \text{ and } Xi \in FIM'\}|}{|NFIM|}}$$

Telle que Xi est un itemset, NSFIM est l'ensembles des itemsets fréquents non sensibles de la base D, et FIM' est l'ensemble des itemsets fréquents de la base D'.

Pour réaliser cette première expérimentation, nous avons varié le nombre des itemsets sensibles, et on a calculé à chaque fois la similarité entre les deux bases. La figure 3.7 représente les résultats de cette expérimentation.

Cela défini le nombre de similarité entre les FIM's sans les SFIM et les FIM's

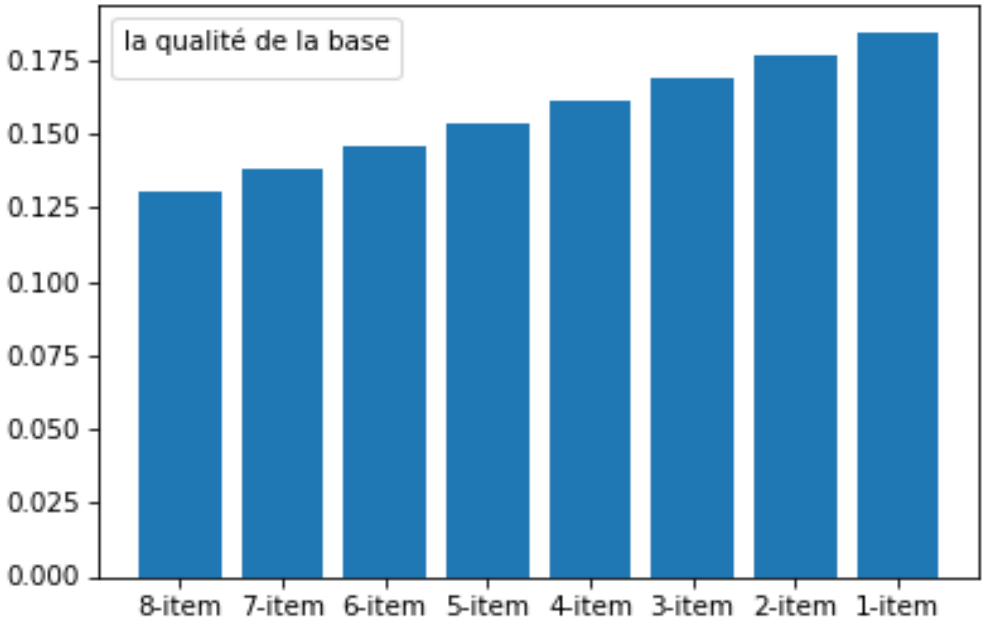

Figure 3.9 : mesure de la qualité de la base produite.

Les résultats de cette expérimentation montrent clairement que plus le nombre des itemsets sensibles est grand plus la qualité de la base produite se dégrade. Dans un autre sens, plus le nombre des itemsets à cacher est grand, plus qu'il y a des itemsets qui sont censés apparaître dans la nouvelle base et n'apparaissent pas. Cela est un effet de bord très connu dans ce type de problème. Notons que les résultats présentent une qualité des données très faible, qui ne dépasse pas les 18 % dans les meilleurs des cas.

3.5.3 Expérimentation 2 :

Dans cette expérimentation nous mesurons le taux de protection des itemsets sensibles suite à la génération de la nouvelle base. Ce taux est défini comme étant le rapport des itemsets sensibles qui apparaissent dans la nouvelle base, sur le nombre total des itemsets sensibles (SFIM). Le taux de protection est calculé selon la formule suivante :

$$\boldsymbol{Tprotect(D) = 1 - \frac{|\{Xi | Xi \in SFIM \text{ and } Xi \in FIM'\}|}{|SFIM|}}$$

Telle que Xi est un itemset, SFIM est l'ensembles des itemsets fréquents sensibles de la base D, et FIM' est l'ensemble des itemsets fréquents de la base D'.

Pour réaliser cette expérimentation, nous avons adopté la même stratégie que la première expérimentation, c'est-à-dire, nous avons varié le nombre des itemsets sensibles, et on a

calculé à chaque fois le taux de protection. La figure 3.8 représente les résultats de cette expérimentation.

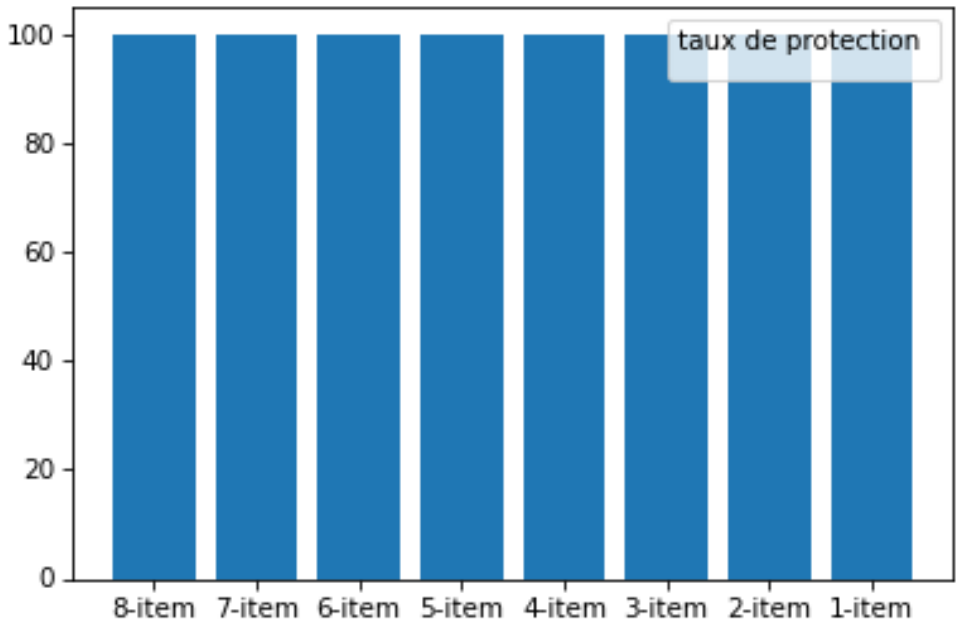

Figure 3.8 : *taux de protection*

Les résultats de cette expérimentation montrent que notre algorithme a généré une base avec un taux de protection de 100%, quelque soit le nombre des itemsets sensibles. En d'autres termes, il n'y a aucun itemset sensible qui apparait dans la base générée.

À la fin, on peut dire que le plus grand avantage de notre approche c'est qu'elle a pu atteindre un taux de protection de 100% sur le jeu de test. L'inconvénient majeur est la qualité faible de la base produite, ce qui nécessite d'introduire des améliorations sur ce plan. Un autre inconvénient de cette approche est sa gourmandise en termes de ressources matérielles (mémoire et temps d'exécution), cela, nous a empêché de faire des tests de validation plus poussés.

3.6 CONCLUSION :

Dans ce chapitre nous avons introduit les détails de conception et d'implémentation d'une approche de protection des itemsets fréquents. Nous avons introduit en premier lieu le principe général de notre approche. Ensuite, nous avons introduit l'environnement et les outils de développent. À la fin, nous avons présenté quelques tests de validation, ainsi que la discussion des résultats obtenus.

CONCLUSION GENERALE ET PERSPECTIVE DE DEVELOPPEMENT

L'extraction des motifs fréquents reste toujours l'un des sujets d'actualités pour les chercheurs car les domaines et les applications qui lui font appels ne cessent de s'accroitre.

A travers ce document, on a abordé le domaine de protection des itemsets fréquents. Pour ce faire, nous avons étudié les principaux algorithmes d'extraction des itemsets fréquents. Par la suite, nous avons entamé la problématique de la reconstruction ainsi que ces approches dans le cadre de la protection de la vie privée. À la suite de cette étude théorique, nous avons présenté notre travail, qui consiste à implémenter une approche de protection des itemsets fréquents. Cette approche se base sur la reconstruction des bases de données transactionnelles. Les expérimentations faites pour valider notre implémentation ont montré un taux de protection considérable, mais malheureusement, une utilité des données produites très faibles. Cela, nous ramène vers notre première perspective qui est l'amélioration de notre implémentation en ajoutant des méthodes qui nous permettent de faire une extension de la base de données comme stipulé par [18]. Cette extension permet de rajouter une partie des itemsest supprimés sans faire apparaitre les SFIM, ce qui aura pour effet de rendre les données plus utiles.

Parmi les autres perspectives qui sont relatives à notre implémentation, on cite l'amélioration des performances de notre implémentation pour optimiser surtout sa consommation en ressources matérielles. En améliorant cette dernière nous pourrons entamer d'autres expérimentations telles que l'étude de l'influence de la taille des SFIM sur la qualité et le taux de protection des bases produites. D'autres perspectives comme le test de notre implémentation sur d'autres bases et la comparaison de notre travail avec d'autres travaux sont aussi envisageables.

REFERENCE :

[1] R.chalal "<<une approche pour la capitalisation coopérative des connaissances sur les risques produit en phase initiale d"un projet industriel>> thése de doctorat,INI, decembre 2007

[2]Agrawal, R. et Srikant, R. (1994) Algorithmes rapides pour les règles d'association minière dans les grandes bases de données. Actes de la 20e Conférence internationale sur les très grandes bases de données, VLDB, Santiago du Chili, 12-15 septembre 1994, 487-499.

[3] Javeed,MZ. Scalable algorithms for association mining. IEEE Trans Knowl Data Eng 12(3):372-390, 2000.

[4] Aggarwal CC, Prasad VVV .A tree projection algorithm for generation offrequent itemsets. J Parallel Distrib Comput 61(3):350–371, 2001

[5] Chen MS, Han J, Yu PS. Data mining: an overview from a database perspective. IEEE Trans Knowl Data Eng 1996;8(6):866-83. doi:10.1109/69.553155.

[6]Oliveira SR, Zaiane OR. A unified framework for protecting sensitive association rules in business collaboration. Int J Bus Intell Data Min 2006;1(3):247-87.

.

[7]Lin CW, Hong TP, Yang KT, Wang SL. The ga-based algorithms for optimizing hiding sensitive itemsets through transaction deletion. Appl Intell 2015;42(2):210-30.

.

[8]Dasseni E, Verykios VS, Elmagarmid AK, Bertino E. Hiding association rules by using confidence and support. In: Proceedings of the international workshop on information hiding; 2001. p. 369-83.

[9]Verykios VS, Elmagarmid AK, Bertino E, Saygin Y, Dasseni E. Association rule hiding. IEEE Trans Knowl Data Eng 2004;16(4):434-47.

doi:10.1109/TKDE.2004.1269668.

[10] h Wu Y, m Chiang C, p Chen AL. Hiding sensitive association rules with limited side effects. IEEE Trans Knowl Data Eng 2007;19(1):29-42. doi:10.1109/TKDE.2007.250583.

[12]Sun X, Yu PS. A border-based approach for hiding sensitive frequent itemsets. In: Proceedings of the fifth IEEE international conference on data mining (ICDM'05); 2005. p. 8. doi:10.1109/ICDM.2005.2.

[13] Moustakides GV, Verykios VS. A max-min approach for hiding frequent itemsets. In: Proceedings of the sixth IEEE international conference on data mining - workshops (ICDMW'06); 2006. p. 502-6.

doi:10.1109/ICDMW.2006.8.

[14] Menon S, Sarkar S. Minimizing information loss and preserving privacy. Manag Sci 2007;53(1),101-16. Menon S, Sarkar S, Mukherjee Maximizing accuracy of shared databases when concealing sensitive patterns.. Inf Syst Res 2005;16(3):256-70.

[15]Gkoulalas-Divanis A, Verykios VS. An integer programming approach for frequent itemset hiding, 10; 2006. p. 748-57.

[16]Gkoulalas-Divanis A, Verykios VS. An integer programming approach for frequent itemset hiding, 10; 2006. p. 748-57.

[17]Xu L, Jiang C, Wang J, Yuan J, Ren Y. Information security in big data: privacy and data mining. IEEE Access 2014;2:1149-76. doi:10.1109/ACCESS.2014.2362522.

[18]Gkoulalas-Divanis A, Verykios VS. Exact knowledge hiding through database extension. IEEE Trans Knowl Data Eng 2009;21(5):699–713.

doi:10.1109/TKDE.2008.199.

[19]Guo Y. Reconstruction-based association rule hiding. In: Proceedings of the SIGMOD Ph. D. workshop innovative database research; 2007. p. 51-6.

[20]Jingu Kim, Taneli Mielikäinen:

Conditional Log-linear Models for Mobile Application Usage Prediction. ECML/PKDD (1) 2014: 672-6872011

[21]Solutions M. Advantages and Disadvantages of Python Programming Language [Internet]. Medium. 2017 [cité 27 janv 2020]. Disponible sur : https://medium.com/@mindfiresolutions.usa/advantages-and-disadvantages-of-python programming-language-fd0b394f2121

[22]Jupyter Notebook [Internet]. IONOS Digitalguide. [cité 27 janv 2020]. Disponible sur https://www.ionos.fr/digitalguide/sites-internet/developpement-web/jupyter-notebook/

[23]Kraeber J-M. Tutoriel Utiliser un Jupyter Notebook [Internet]. Lycée Antoine de Saint Exupery. 2019 [cité 27 janv 2020]. Disponible sur: http://lycee-saint-exupery.fr/tutoriel-utiliser-un jupyter-notebook/

[24]https://ledatascientist.com/google-colab-le-guide-ultime/

[25] https://github.com/chonyy/fpgrowth_py/blob/master/dataset/data2.csv

[26] https://github.com/chonyy/fpgrowth_py/tree/master/dataset/kaggle.csv

[27]file:///C:/Users/HP/Desktop/MEMOIRE/DAAOU%20Zineddine%20(Extractions%20des%20motifs%20fréquents%20orientés%20besoins%20du%20décideur..pdf

[28]http://igm.univ-mlv.fr/~dr/XPOSE2012/datamining/datamining-domaines-application.html

[29]https://fr.wikiversity.org/wiki/Datamining/Applications#:~:text=Domaines%20d%27application,nombre%20de%20domaine%20d%27activités.&text=Identification%20de%20clients%20potentiels%20de,découverte%20de%20clients%20à%20risque.

[30] https://www.piloter.org/techno/support/base-de-donnees-relationnelle-definition.htm

[31] https://fr.wikipedia.org/wiki/Entrepôt_de_données

[32] https://fre.myservername.com/apriori-algorithm-data-mining

[33] https://fre.myservername.com/frequent-pattern-growth-algorithm-data-mining

[34] Heaton, J. Comparing dataset characteristics that favor the Apriori, Eclat or FPGrowth frequent itemset mining algorithms. SoutheastCon, 2016.

[35] . The Incredible Growth of Python | Stack Overflow [Internet]. Stack Overflow Blog. 2017 [cité 27 janv 2020]. Disponible sur: https://stackoverflow.blog/2017/09/06/incredible-growthpython/

[36] Extraction des Motifs Fréquents Données Transactionnelles -Belabed Amine-maitre assis tant Université De Tlemcen – Faculté des sciences, Département d'Informatique