République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieure et de la Recherche Scientifique
Université Abderrahmane Mira
Faculté de la Technologie

جامعة بجاية
Tasdawit n Bgayet
Université de Béjaïa

Department

Département d'Automatique, Télécommunications et d'Electronique

Projet de Fin d'Etudes

Pour l'obtention du diplôme de Master

Filière : Télécommunications

Spécialité : Réseaux et télécommunications

Thème

Détection des transactions frauduleuses par cartes de crédit à base d'algorithmes d'apprentissage automatique

Préparé par :

- Mokrane Karima

Dirigé par :

M. Diboune Abdelhani

Examiné par :

M. Tounsi Mohamed

M. Boualem Mohamed

Année universitaire : **2022/2023**

Remerciement

Je remercie tout d'abord le bon Dieu, le tout puissant de m'avoir accordée le savoir de m'avoir orientée vers le droit chemin.

Je tiens à exprimer aussi mes reconnaissances à mon encadrant M. DIBOUNE, pour l'honneur qu'il m'a fait en assurant la direction et le suivi scientifique de ce mémoire et pour sa patience et son encouragement.

Je tiens aussi à remercier les membres du jury qui ont accepté d'examiner ce modeste travail.

Dédicace

Au nom de Dieu clément et miséricordieux Je dédié ce modeste travail à :

Celui qui m'a toujours soutenu, à mon défunt père parti

Avant la fin de ce travail.

A ma chère mère pour son soutien et sa patience.

A mes sœurs, Ainsi qu'à tous les membres de ma famille Pour

Leurs soutiens, leur aide et leurs sympathies dans les moments difficiles.

A mes chères amies Pour leurs aides et leur support :

Mayliss, Imane, Massyl et Brahim.

A tous ceux qui ont contribué de près ou de loin

À La réalisation de ce travail.

Sommaire

Liste des Figures

Liste des Abréviations

Abréviation	Libellé en anglais	Libellé en français
AVS	Address verification system	Système de vérification d'adresse
B2B	Bisness to bisness	Entreprise à entreprise
B2C	Bisness to client	Entreprise à client
CVC	Card verification systm	Système de vérification de carte
EDI	Electronic Data Interchange	Données électronique
FTC	Federal Trade Commission	Commission fédérale du commerce
KNN	K-Nearest Neighbords	K-plus proche voisin
LOF	Local outlier factor	Facteur local d'anomalie
ML	Machine learning	Apprentissage automatique
PCA	Analyse en composantes principale	Analyse en composantes principale
SET	Secure Electronic Transaction	Sécurité de la couche de transport
SSL	Secure Socket Layer	Couche de sockets sécurisée
SMOTE	Synthetic Minority Over-sampling Technique	Technique de sur-échantillonnage synthétique des minorités
SVM	Support Vector Machine	Machine à Vecteurs de Support
TLS	Transport Layer Security	Sécurité de la couche de transport
WWW	World Wide Web	/

Liste des tableaux

Introduction générale

Introduction générale :

Nous vivons actuellement dans un monde numérique, les transactions financières par cartes de crédit jouent un rôle essentiel dans notre vie quotidienne : pour effectuer des achats ou pour payer des factures. Cependant, cette commodité n'est pas sans risques, car les fraudes liées aux transactions par cartes de crédit constituent une grande menace pour les consommateurs, les institutions financières et l'économie en générale.

La détection des transactions frauduleuses est vite devenue une priorité majeure pour les institutions financières. Les méthodes traditionnelles de détection préventives ont montré leurs limites et leurs incapacités à suivre l'évolution continue des schémas de fraude. C'est là que l'apprentissage automatique entre en jeu [24].

Les algorithmes d'apprentissage automatique permettent de détecter les transactions frauduleuses de manière efficace et précise. Grâce à leur capacité à analyser d'énormes quantités de données et à identifier des schémas complexes, ces algorithmes permettent de repérer les comportements suspects et les activités frauduleuses difficiles à détecter.

L'objectif principal de ce mémoire se concentre sur l'utilisation d'algorithmes d'apprentissage automatique pour la détection des transactions frauduleuses par cartes de crédit. Avec l'exploration des différentes techniques d'apprentissage telles que (la gaussienne multivariée, la forêt d'isolation et le facteur local d'anomalie) et l'étude de leurs performances pour améliorer l'efficacité des systèmes de détection des fraudes pour renforcer la sécurité des transactions par cartes de crédit et réduire les pertes financières pour les individus et les institutions.

Pour ce faire, nous avons structuré notre mémoire en trois chapitres :

Le premier chapitre se repose sur le principe des transactions par cartes de crédit dans l'e-commerce, l'impact de la fraude sur les finances. Ainsi que les systèmes de détection des transactions frauduleuses par carte de crédit et la sécurité des transactions par cartes de crédit.

Le deuxième chapitre est consacré à l'apprentissage supervisé pour la détection d'anomalies, le problème de classes déséquilibrées et les solutions proposées. Ainsi que les méthodes utiliser pour la détection d'anomalies et la définition des métriques de performances Le troisième chapitre, porte sur l'implémentation de la partie expérimentale de notre travail,

avec une évaluation des performances des différents modèles utilisés (gaussienne multivariée, la forêt d'isolation, le facteur local d'anomalie, le KNN, le SVM et l'arbre de décision) avec une discussion des résultats obtenus. On termine par une conclusion générale.

Chapitre I :

Généralités

1 Contexte générale :

1.1. Qu'est-ce que l'e-commerce ?

L'E-Commerce a révolutionné la manière de faire ses achats en la rendant plus facile et plus rapide à partir de n'importe quel endroit du monde et à n'importe quelle heure.

L'e-commerce qui est l'abréviation d'électronique commerce est une méthodologie commerciale moderne qui répond au besoin des organisations, des commerçants et des consommateurs en réduisent les coûts tout en améliorant la qualité des biens et des services en accélérant la livraison.

L'e-commerce est aussi associé au fait d'acheter ou de vendre des informations, des services et des produits sur les réseaux informatiques en combinant plusieurs procédés tel que, l'échange de données électronique (EDI), l'email électronique (e-mail), World Wide Web (WWW) et les applications internet.

Dans une transaction d'e-commerce le client peut parcourir à travers plusieurs produits, passer une commande, payer et recevoir le produit le tout en ligne, ces échanges peuvent être des transactions d'entreprise à entreprise (B2B) (business to business), des transactions entre entreprises et clients final (B2C) (business to client) ou des échanges entre clients [15].

1.2. Importance des transactions par carte de crédit dans l'e-commerce

Parmi l'ensemble des systèmes de paiement, le paiement par carte de crédit est le plus utilisé dans l'e-commerce et cela pour :

Sa facilité d'utilisation car il suffit seulement de rentrer les coordonnées et les informations de la carte de crédit lors du processus de paiement en ligne.

Sa rapidité de transaction qui signifie que les clients peuvent effectuer leur paiement et finaliser leur achat en quelques secondes seulement.

Sa sécurité qui est constituée des protocoles de sécurisation des paiements par carte de crédit qui sont caractérisés par des niveaux de sécurité tels que SSL (Secure Socket Layer), TLS (Transport Layer Security) et SET (Secure Electronic Transaction).

En générale, les transactions par carte de crédit sont un mode de paiement indispensable à

L'e-commerce, car elles permettent aux clients d'effectuer leurs achats en ligne de manière facile, rapide et sécurisée [5].

1.3. L'émergence de la fraude financière par carte de crédit

Avec le développement de l'e-commerce au cours de ces dernières années, la fraude financière par carte de crédit a augmenté avec le développement des nouvelles techniques de fraude. La fraude à la carte de crédit désigne l'utilisation frauduleuse des coordonnées de la carte bancaire d'une personne à son insu alors que celle-ci est pourtant toujours en possession de sa carte. En 2014 le nombre de fraudes a atteint 5000 fraudes par jour [24].

- **Méthodes de fraudes par carte de crédit**

On matière de fraude et de vols de données les hackers ne manquent pas d'idées, parmi les méthodes les plus utilisées nous décrivons.

- **Fraude par opération sans présentation de la carte :** ce type de fraude ne requiert pas la carte matérielle, il suffit d'avoir les identifiants nom, numéro de carte et mot de passe pour l'utilisation frauduleuse de la carte.
- **Fraude par carte de crédit contrefaite (skimming)** : qui consiste à obtenir les renseignements enregistrés sur la bande magnétique de la carte de crédit et le code pin à 4 chiffres, avec ces informations l'hacker peut créer un clone de la carte original et l'utiliser de manière normale.
- **Hameçonnage :** un fraudeur se fait passer pour une entité de confiance, par exemple une banque ou une autre institution financière reconnue en utilisant le courriel message texte ou le téléphone. Il peut s'agir aussi d'un lien vers un faux site Web similaire à celui d'une entreprise légitime, ainsi le fraudeur peut vous convaincre de transmettre vos renseignements de carte de crédit pour résoudre un problème ou éponger une dette, il peut aussi vous faire croire que vous payez une facture sur un site Web légitime. Le tout pour obtenir vos renseignements de carte de crédit.
- **Piratage d'une base de données en ligne :** en utilisant les techniques d'hacking et de l'ingénierie sociale les pirates exploitent les failles de sécurités dans les systèmes d'authentification, les pare-feux et les logiciels pour accéder aux données personnelles stockées dans la base de données telles que les coordonnées bancaires.
- **Fraude par interception de courrier :** lors de l'envoie d'une nouvelle carte ou d'une carte de remplacement cette dernière pourrait être volée dans votre boîte aux lettres à votre insu.
- **Fraude par prise de possession de votre compte :** si un fraudeur possède vos renseignements personnels (nom, date de naissance et numéro d'assurance sociale, etc.)

Il peut communiquer avec votre banque on se faisant passer pour vous. Il pourra ainsi demander une nouvelle carte en votre nom et se la faire livrer [12].

1.4. Pertes financières par les transactions frauduleuse

Étant donné que les pertes financières causées par les fraudes bancaires ont fortement augmenté ces dernières années avec le développement des nouvelles techniques de fraude.

La FTC (Federal Trade Commission) a déclaré que en 2021 environs 390 000 rapports ont été signalés pour fraude par carte de crédit [11].

D'après le Nilson Report de décembre 2022, les pertes financières par carte de crédit vont atteindre près de 165.1 milliards de dollars durant la prochaine décennie aux Etats-Unis. Et 397.4 milliards dollars à travers le monde.

Durant les dernières années le taux de perte due aux fraudes par cartes de crédit n'a pas cessé d'augmenter comme le montre la figure I.1 [11].

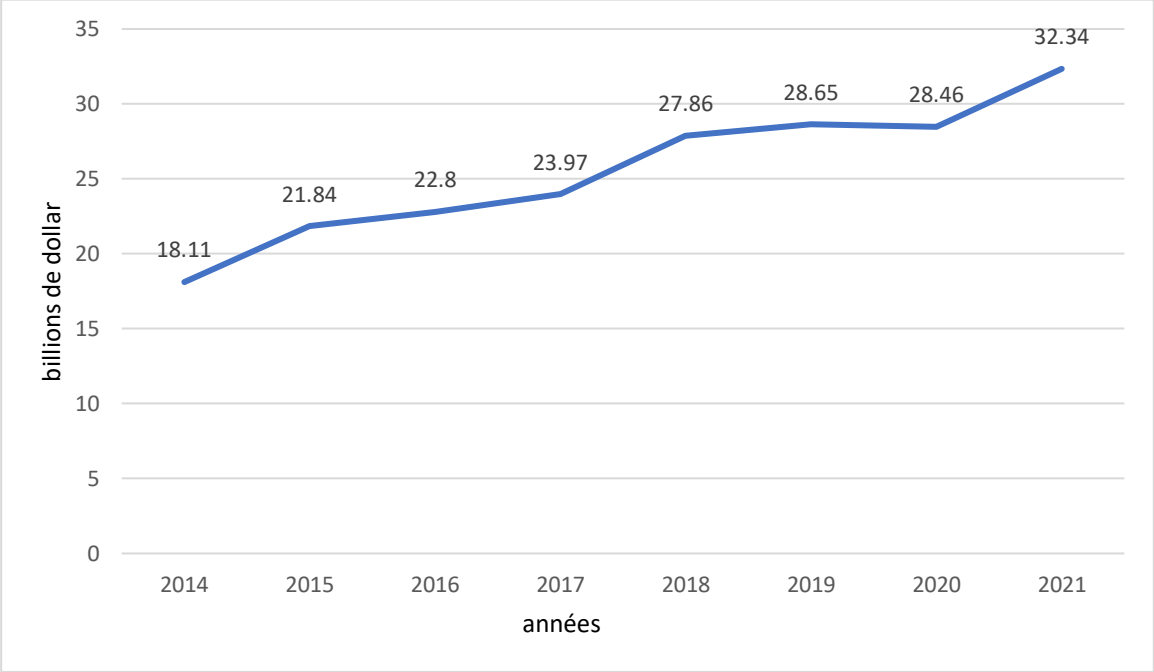

Figure I. 1: Les pertes financières dues aux fraudes par carte de crédit enregistrées durant les dernières années [11].

Selon la Fédéral Trade Commission chaque année plusieurs millions de rapports sont faits pour signaler les fraudes par carte de crédit tel que le montre la figure I.1 suivante pour les cinq dernières années [11].

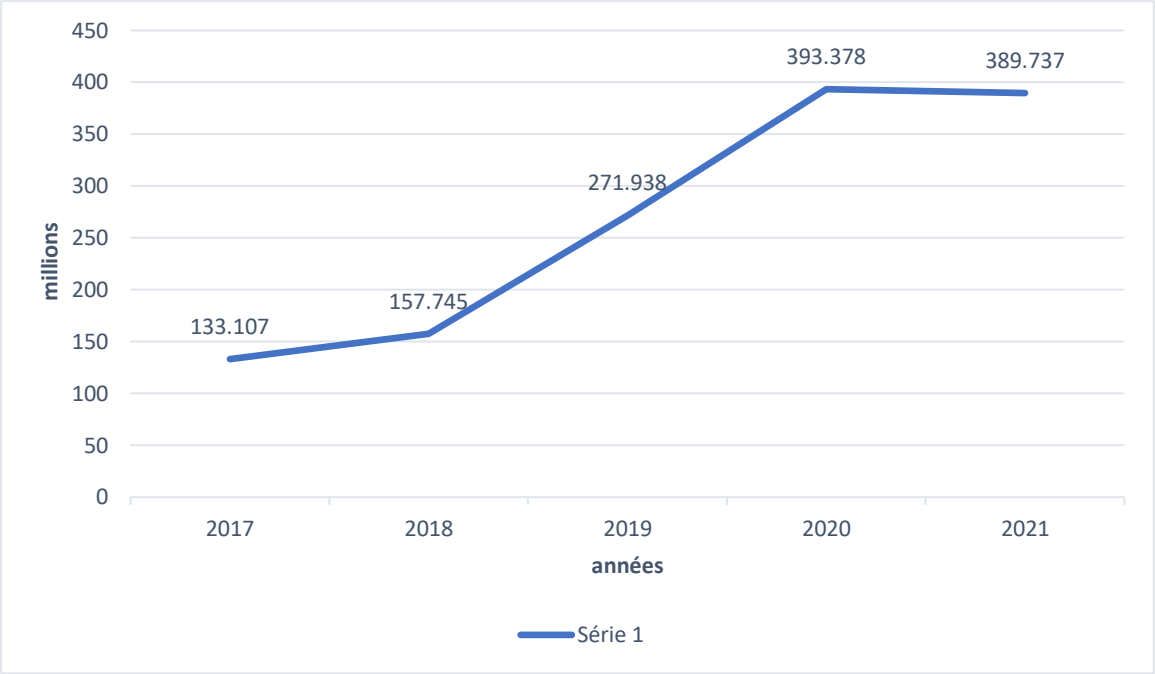

Figure I. 2: Nombre de rapports pour fraude à la carte de crédit [11].

L'implémentation d'un système de détection de fraude par carte de crédit devient indéniable afin de sécuriser pour le mieux les données bancaires et les transactions par carte de crédit [11].

II. Système de détection des transactions frauduleuses par carte de crédit :

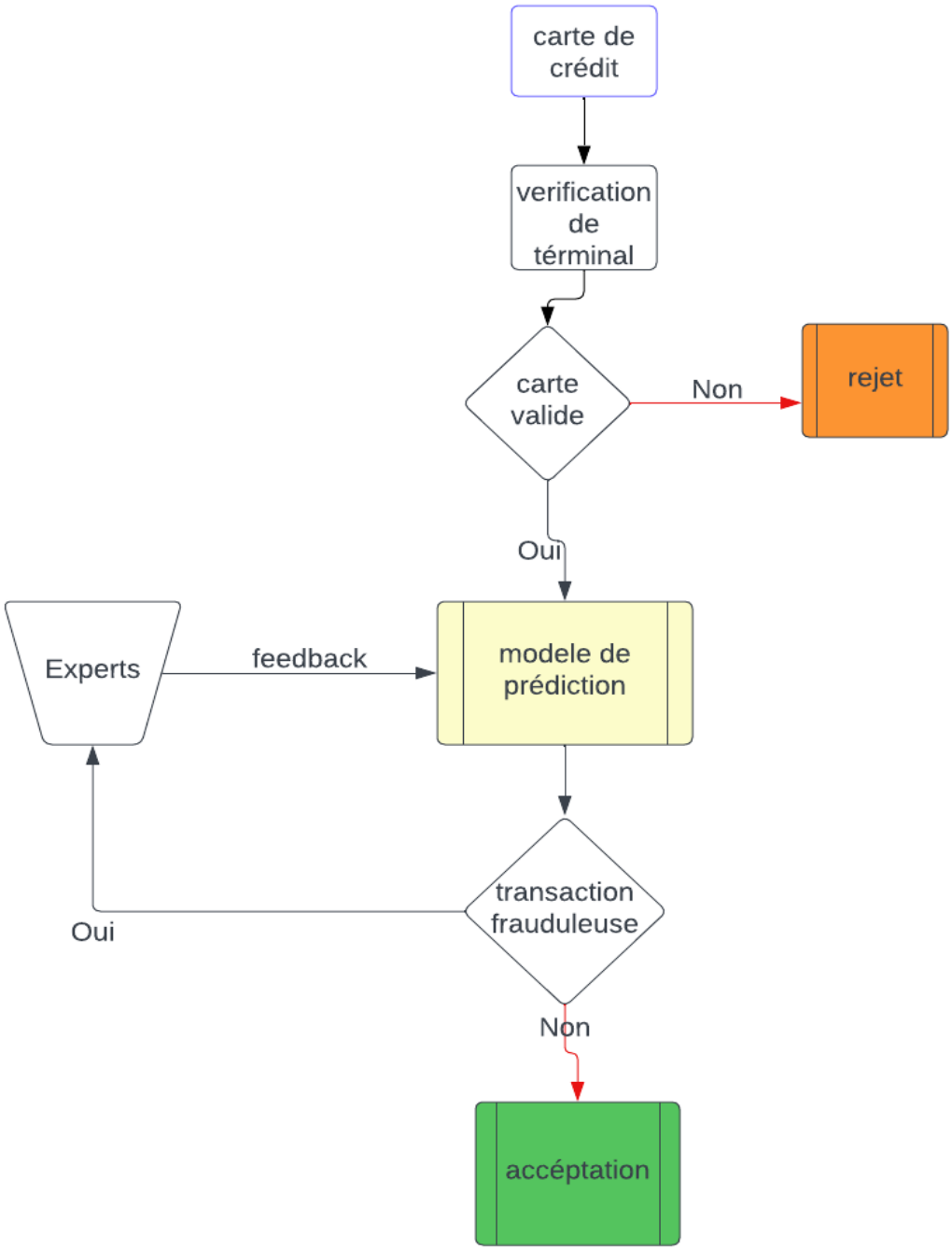

Figure I. 3: Fonctionnement d'un système de détection des transactions frauduleuses

Lorsqu'une transaction par carte de crédit est effectuée, plusieurs étapes de vérification peuvent être mises en place pour réduire les risques de fraude comme on peut le voir sur la figure I.3. Lorsqu'une transaction est effectuée, on vérifie le terminal de paiement et on vérifie si la carte de crédit utilisée est valide et n'est pas associée à une fraude connue si elle n'est pas valide elle

est directement rejetée si non un modèle de prédiction basé sur l'apprentissage automatique peut être utilisé pour évaluer la probabilité qu'une transaction soit frauduleuse si la transaction n'est pas frauduleuse, alors la transaction est accepter si non, une expertise est faites, lorsqu'une transaction est identifiée comme frauduleuse, un feedback est utilisé pour mettre à jour le modèle de prédiction.

III. Sécurité des transactions par cartes de crédit

Avec la diversification des méthodes de fraude par carte de crédit (vol de coordonnées bancaires, skimming, hameçonnage, etc), la prévention contre toutes ses méthodes deviennent obligatoires pour la sécurité des transactions et des payements en ligne afin de diminuer les risques de piratage avec un délai de micro seconde nous disposons de plusieurs méthodes préventives [15].

III.1. Méthodes préventives

- **Système de vérification d'adresse (AVS) :** il désigne la vérification du code postal et de l'adresse de facturation qui permettent de confirmer si les informations correspondent à l'adresse de facturation enregistrée par le détenteur de la carte bancaire, si les informations ne correspondent pas à une règle qui permet de signaler le problème aux banques afin de bloquer les paiements qui ne vérifient pas le code postal.
- **Système de vérification de carte (CVC) :** il comprend un numéro à 3 ou 4 chiffres imprimés sur la carte bancaire ou au verso, ce système permet de vérifier tous les paiements incluant le système CVC. On peut vérifier ce dernier en fournissant le code CVC lors de la création d'un paiement par carte bancaire, pour revérifier le CVC d'une carte préalablement enregistrée on se refaire aux guides d'intégration avec récupération du CVC [17].

III.2. Méthodes de détection automatiques

Ces méthodes sont utilisées afin de détecter les nouvelles techniques de fraude :

III.2.1. Méthodes basées sur les règles : ce sont des méthodes définies manuellement qui mettent en œuvre des règles de détections figées et définies par l'utilisateur nécessitant une grande connaissance. La méthode des règles est basée sur les métriques pour détecter les anomalies (les transactions frauduleuses) et faire une analyse du domaine, ensuite avec un mécanisme de filtrage elle réduit la taille des données et elle détecte tous les défauts et les anomalies afin de les classées en transaction frauduleuse ou transaction normale.

Malgré tout, la méthode basée sur les règles reste une méthode peut pratique car les règles ne peuvent pas évoluer lors de l'exécution du programme de détection (pas de modification automatique) [18].

III.2.2. Méthodes basées sur l'apprentissage automatique : également appelées apprentissage artificiel ou machine Learning. L'apprentissage automatique s'appuie sur l'étude d'algorithmes et de modèles statistiques qui permettent aux ordinateurs d'effectuer des tâches et de prendre des décisions à partir des données sans être programmées, le type d'algorithme à utiliser dépend du type de données, le nombre de variables et le modèle qui convient le mieux.

Avec le progrès des nouvelles technologies, l'apprentissage automatique est présent dans la vie quotidienne et dans plusieurs champs d'études, comme la détection de fraude, les systèmes de recommandation utilisé par les sites web et les réseaux sociaux, la prédiction des comportement des clients (modèles d'achats et habitudes de dépenses) ainsi que dans la médecine, les systèmes d'apprentissage automatique médicaux en particulier. Les arbres de décisions donnent de bons résultats dans le domaine de la cardiologie pour la prédiction de risque de mortalité [8].

On trouve différentes catégories d'apprentissage automatique on peut les diviser en 3 catégories principales.

- **Apprentissage supervisé** : dans ce type d'apprentissage l'ensemble des données d'apprentissage sont bien définis et classifiées, la variable de sortie est connue et l'algorithme d'apprentissage fournit un mappage entre les caractéristiques d'entrées et la variable de sortie, l'apprentissage supervisée a deux clases la régression et la classification, il s'agit d'une régression quand la variable prédite est une valeur réelle tandis qu'il s'agit d'une classification lorsque la variable prédite est discrète [10].

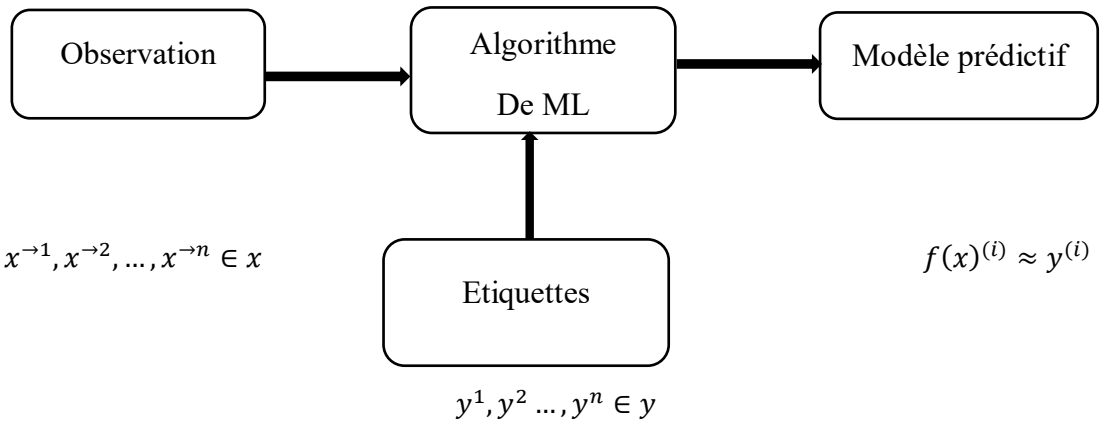

Figure I. 4: Apprentissage supervisé [10]

- **Apprentissage non supervisé** : dans cet apprentissage les variables de sortie ne sont pas explicites et les relations sont générées en fonction des données d'entrées fournies à l'algorithme, le but est de trouvées les différences dans les données qui peuvent être utilisées pour la prise de décision. Dans l'apprentissage non supervisé on trouve deux exemples classiques : la réduction de dimension et le regroupement. La réduction de dimension permet d'extraire un plus petit nombre de caractéristiques qui décrivent les données suffisamment, le regroupement quant à lui consiste à partitionner les données en groupes (le clustering) de sorte que les objets de chaque groupe partagent certaines caractéristiques communes entre eux [2].
- **Apprentissage semi-supervisé** : c'est une approche de l'apprentissage automatique ou la quantité de données étiquetées est plus inferieure que celle des données non étiquetées. Les algorithmes d'apprentissage semi-supervisé visent à utiliser des données non étiquetées pour apprendre un niveau supérieur de représentations, puis utiliser les exemples étiquetés pour guider la tâche d'apprentissage en aval. On peut trouver les algorithmes d'apprentissages dans plusieurs technologies comme le cloud et l'assistant virtuel d'Amazon Alexa [2].

IV : Défis à relever lors de l'implémentation d'un système de détection automatique des transactions frauduleuses

Lors de l'implémentation d'un système de détection de fraude, il y a plusieurs défis a relevés tels que ;

- Le choix du bon modèle d'apprentissage automatique à utiliser et cela dépend de plusieurs facteurs, tels que la nature des données si elles sont structurées ou numériques, leurs formats et leurs taille disponible ainsi que leurs distributions, le type de problème à résoudre (classification ou régression) et les performances souhaitées.
- L'extraction et la création des bonnes caractéristiques est une étape cruciale du processus de features engineering qui permet d'améliorer la performance des modèles d'apprentissage automatique. Il transforme les données brutes qui peuvent être utilisées en entrées d'un algorithme de l'apprentissage automatique en nouvelles variables [21].
- La distribution asymétrique de classes peut perturber les modèles d'apprentissage automatique et les rendre moins performants lors de la détection de fraude, dans de nombreux cas, la majorité des transactions sont légitimes et seulement une petite partie est frauduleuse. Cette distribution peut perturber la détection de fraude en biaisé en faveur de la classe majoritaire. Pour surmonter ce défi, la conception des modèles de

détection doit tenir compte des distributions de classe asymétriques pour garantir une évaluation équitable en tenant compte des taux de faux positifs.

- Le déséquilibre de données (imbalanced data), lorsque la taille de la classe positive est très inférieure à la taille de la classe négative, dans notre cas le nombre de transaction frauduleuse est très inferieure par rapport au nombre de transaction légitime.

  Pour remédier à ce problème dans l'apprentissage automatique les trois méthodes les plus utilisées sont, le sur-échantillonnage de la classe minoritaire, le sous-échantillonnage de la classe majoritaire et l'utilisation d'algorithmes d'apprentissage spécifiques (arbre de décision et les forêts aléatoires) [2].

  Il est important de choisir la technique la plus appropriée en fonction des caractéristiques de données spécifiques. On peut également combiner plusieurs techniques pour obtenir de meilleurs résultats [2].

- Le principal défi de l'utilisation des données dynamiques dans la détection de fraude est de gérer un grand flux de données générées et mis à jour en temps réel à mesure de nouvelles transactions. Les systèmes doivent collecter, stocker et analyser de grandes quantités de données provenant de diverses sources à une grande vitesse en temps réel et avec une grande précision. Il est aussi nécessaire de garantir la confidentialité et la sécurité des données des transactions par carte de crédit qui contiennent des informations personnelles [26].

V. Objectifs du mémoire

La détection automatique efficace des transactions frauduleuses par carte de crédit et le maintien de la confidentialité des données est un domaine en extension continue dans la sécurité des payements.

Ce travail permet de présenter les différentes méthodes de détection utilisées par les institutions financières pour protéger leurs clients, on abordera la détection basée sur l'apprentissage automatique et tous les différents algorithmes utilisés en prenant en considération les bases de données avec des distributions asymétriques de classes et de déséquilibre de données, qui rendent la détection des transactions frauduleuses plus difficile, nous allons voir plusieurs méthodes utilisées pour remédier à ce problème ainsi faire une étude comparative entre elles afin de trouver les plus efficaces pour la détection des transactions frauduleuses par carte de crédit.

Chapitre II :

Apprentissage supervisé pour la détection d'anomalies

1.Introduction

La détection d'anomalies lors des transactions par carte de crédit est très compliquée, surtout avec toutes les méthodes de fraudes développées. Néanmoins, l'utilisation de modèles d'apprentissage automatique supervisé rends la détection d'anomalies plus simple.

Dans ce chapitre, nous allons aborder l'apprentissage supervisé dans la détection d'anomalies, les problèmes de classes déséquilibrées et comment les résoudre, et pour finir on parlera des métriques de performance.

2.Apprentissage supervisé

2.1. Définition

L'apprentissage automatique est une méthode qui utilise des données étiquetées : des données d'entrainement qu'on fournit à l'algorithme comportant les classifications désirées ; pour but de minimiser l'erreur en comparant sa sortie réelle avec les sorties prédite [13].

Dans l'apprentissage automatique on utilise les fonctions de mappage pour transformer les données brutes en une représentation plus adapter aux problèmes de l'apprentissage supervisé, avec l'extraction des caractéristiques à partir des données brutes, telles que le montant de la transaction, la date, l'heure et le lieu de la transaction, le numéro de carte et le code secret. On peut réduire la dimension, en éliminant les variables non pertinentes afin d'avoir une base de données plus facile à interpréter.

Les fonctions de mappage dans l'apprentissage supervisé sont utilisées pour décrire la relation entre les entrées (X) et les sorties (Y) dans un modèle de prédiction.

Un ensemble de données D, est décrit par un ensemble de caractéristiques X, avec un algorithme d'apprentissage supervisé on va trouver une fonction de mappage entre les variables prédictives en entrée X et la variable à prédire Y. La fonction de mapping décrivant la relation entre X et Y s'appelle un **modèle de prédiction**. $f(x) = y$

$$f: X \to Y$$

$$x \to f(x) = y$$

Les caractéristiques (features en anglais) X peuvent être des valeurs numériques, alphanumériques, des images, etc [10]. Quant à la variable prédite Y, elle peut être de deux catégories :

- **Variable discrète** : la variable à prédire peut prendre une valeur d'un ensemble fini de valeurs (qu'on appelle des classes). Par exemple, pour prédire si une transaction est frauduleuse ou normale, la variable Y peut prendre deux valeurs possibles : positive ou négative qu'on appelle problème de **classification.**
- **Variable continue** : la variable Y peut prendre des valeurs réelles. Pour illustrer cette notion, on peut citer un algorithme qui prend en entrée les caractéristiques d'une transaction, et tentera de prédire son montant qu'on appelle problème **régression.**

  Dans le cadre de ce mémoire on s'intéresse au problème de classification.

2.2. Les algorithmes de classification

On parle de classification quand la variable à prédire prend une valeur discrète, parmi les algorithmes de classification on peut citer :

- **Machine à Vecteurs de Support (SVM) :** c'est un algorithme d'apprentissage automatique qui permet de résoudre les problèmes de classification ou de régression (prédiction d'une valeur numérique). Le SVM sépare un ensemble de données en différentes classes à l'aide d'un hyperplan en s'appuyant sur la notion de marge maximale. Les points les plus proches de l'hyperplan dans les différentes classes sont appelés vecteurs de support et ces derniers sont utilisés pour prédire les classes des nouveaux points.

  Lorsqu'un nouveau point est placé sur l'équation de l'hyperplan, il est classé dans une classe en fonction du côté de l'hyperplan où il est tombé sur l'espace vectoriel [10].

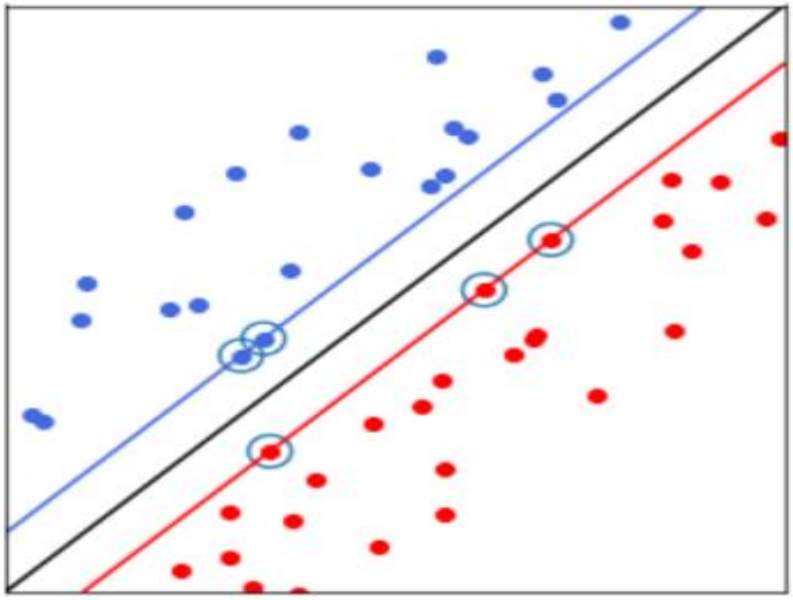

Figure II. 1: Exemple illustratif du fonctionnement du SVM [2].

La Figure II.1 montre le principe général du SVM. Dans cette figure, l'hyperplan est la droite noire, les « vecteurs de support » sont les points entourés (les plus proche de l'hyperplan) et la « marge » est la distance entre l'hyperplan et les droites bleue et rouge.

- **KNN :** l'algorithme des k plus proches voisins, est un classificateur d'apprentissage supervisé non paramétré et non linéaire, qui utilise la proximité pour effectuer des classifications. Il permet d'affecter une nouvelle donnée d'entrée x à la cible la plus présente dans les k données les plus proches selon une distance prédéfinie [2].

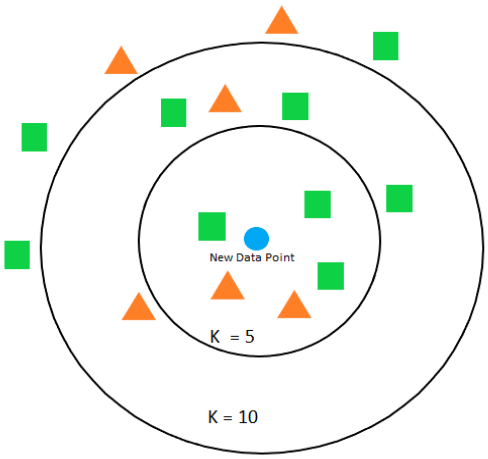

Figure II. 2: Exemple de KNN

La figure II.2 représente un exemple de K-plus proche voisins avec différents paramètres K=5 et K=10. Comme on peut le voir à chaque fois que l'on change le nombre de voisins la classe majoritaire reste la même (les carrés).

- **Arbres de décision :** un arbre de décision est un outil d'aide à la décision qui utilise un modèle arborescent de décision, où chaque nœud interne correspond à un test sur un attribut, chaque branche dénote un résultat de ce test et chaque nœud terminal correspond à une classe. Les arbres de décision permettent de partitionner, récursivement, un ensemble de données en utilisant une approche en profondeur ou en largeur et s'arrêtent lorsque toutes les transactions ont été affectées à une classe particulière [2].

2.3. Types de classifications

Il existe trois types de classification ;

- ✓ **Classification binaire** : dans la classification binaire l'algorithme classe les données selon deux classes que on appelle (**classe négative et classe positive**) on aura :

  Un ensemble de classes $C = \{C1, C2\}$. On déduit ses paramètres selon une fonction paramétrée de la forme $f : X \to \{C1, C2\}$. Le modèle de prédiction dans la classification binaire prend la forme suivante :

$$f(x) = \begin{cases} \boldsymbol{C1} & \boldsymbol{if\ \ g(x) < 0}, \\ \boldsymbol{C2} & \boldsymbol{if\ \ g(x) \geq 0}, \end{cases}$$

  Avec : $\boldsymbol{g : x \to R}$ représente une application paramétrée de $x \in IR^n$

  Dans notre cas les transactions par carte de crédit sont classifiées en deux classe transaction frauduleuse et transaction habituel respectivement classe positif et classe négative.

- ✓ **Classification multi-classes :** elle consiste à classer les données selon plusieurs catégories où le nombre de classes > 2, dans cette classification, chaque exemple est décrit avec plusieurs caractéristiques. On peut traiter la classification multi-classe en divisent la classification en plusieurs classes binaires [25].

Classification binaire **classification multi-classes**

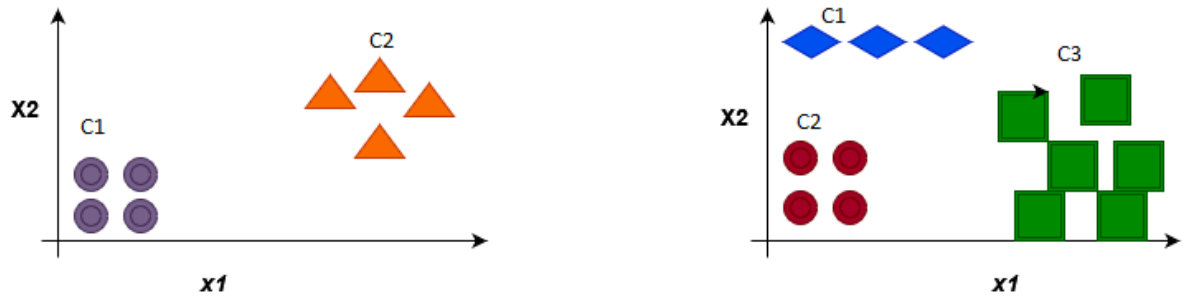

Figure II. 3: Représentation des diagrammes de la classification binaire et de la classification multi-classes

La figure II.3 représente les deux graphes de la classification binaire et de la classification multi-classe, comme on peut le voir à gauche sur la classification binaire les ronds représente la classe C1 et les triangles représente la classe C2 exemple ; une transaction frauduleuse et une transaction valide, tandis que sur la droite la classification multi-classe comporte trois classes distribuer comme suis ; (losange pour la classe C1, ronds pour la classe C2 et les carrées pour la classe C3) [3].

- ✓ **Classification multi-étiquettes :** c'est une classification où chaque instance peut appartenir à plusieurs classes en même temps, Dans la classification multi-étiquette, les exemples sont associés à un ensemble d'étiquettes $Z \in Y$.

  De nos jours, nous remarquons que les méthodes de classification multi-étiquette sont de plus en plus requises par les applications modernes telles que ; la classification des fonctions des protéines et la catégorisation musicale [23].

Dans le cadre de ce mémoire, on s'intéresse à la classification binaire car on parle de deux cas la transaction normale et la transaction frauduleuse.

3.Apprentissage automatique pour la détection des anomalies

3.1. Spécificités du problème de détection des anomalies

La détection d'anomalie dans la fraude bancaire consiste à utiliser des algorithmes d'apprentissage automatique pour détecter les comportements inhabituels du client (les transactions qui sont différentes des comportements normaux), par exemple une transaction effectuée d'un endroit très éloigner de l'emplacement habituel du titulaire du compte, ou une anomalie de fréquence si un grand nombre de transaction sont effectuées en peu de temps ou une anomalie de montant, si le montant de la transaction est supérieur à un certain seuil. Les techniques d'apprentissage supervisé et les algorithmes tels que la détection des densités [2].

anormales ou la détection des clusters sont aussi utilisés. Mais la détection peut être difficile lorsque le nombre de cas positifs est réduit par rapport au nombre de cas négatif très élevé, et si le nombre de types d'instances positives est grand, les nouvelles instances positives peuvent aussi être totalement différentes des instances existantes dans la base de données d'apprentissage automatique. En effet, plus il y a d'instances, plus il peut être difficile de trouver des caractéristiques distinctives pour chaque classe. Cela peut également rendre la tâche de validation des résultats plus complexe, car il peut y avoir des ambiguïtés dans la classification des instances, par exemple, une instance peut être considérée comme positive par un observateur et négative par un autre ce qui peut rendre la tâche de prédiction plus difficile car les algorithmes d'apprentissage automatique ont besoins de données équilibrées pour identifier les schémas d'anomalies avec précision [2].

Il est important de s'assurer que le modèle d'apprentissage automatique est suffisamment sensible pour détecter les cas positifs même s'ils sont très rares dans l'ensemble de données.

3.2. Problème de classes déséquilibrées

3.2.1. Description du problème

Les transactions effectuées par carte de crédit étant très déséquilibrées car la quantité des transactions frauduleuses est très petite par rapport à celle des transactions normales, qu'on appelle respectivement classe minoritaire et classe majoritaire. Une anomalie peut être vue comme un évènement rare et exclus, donc il est difficile de dire si une transaction est frauduleuse ou non seulement par l'observation des attributs de celle-ci , l'utilisation de modèle d'apprentissage automatique pour la prédiction peut s'avérer peu performants avec des classes déséquilibrées ce qui ne permet pas de détecter toutes les anomalies avec précision, c'est pour ce fait que des modèles spéciaux ont été mis en place pour résoudre le problème de déséquilibre des classes.

3.2.2. Solutions au problème de classes déséquilibrées

3.2.2.1. Techniques de re-échantillonnage : on peut citer trois classes : technique de sous-échantillonnage, technique de sur-échantillonnage, technique hybrides.

- **Technique de sous-échantillonnage (undersampling)**

Le sous-échantillonnage est une méthode couramment utilisée pour traiter le déséquilibre des classes, elle consiste à réduire le nombre de cas de la classe dominante pour équilibrer la

distribution. Cette technique est utilisée lorsque la base de données est volumineuse car elle permet de réduire le temps processeur et la ressource mémoire utilisée.

- **Technique de sur-échantillonnage (oversampling)**

Le sur-échantillonnage est une technique qui consiste à dupliquer les instances de la classe minoritaire afin d'équilibrer le rapport entre les deux classes [27].

- **Technique hybride**

L'hybride combine entre le sur-échantillonnage et le sous-échantillonnage, une technique hybride peut impliquer la duplication aléatoire d'observations de la classe minoritaire et la suppression aléatoire d'observations de la classe majoritaire pour équilibrer la distribution des classes [1].

3.2.2.2. Quelques approches basées sur l'échantillonnage

- ✓ **Sous-échantillonnage aléatoire**

  Le sous-échantillonnage aléatoire est une technique de sous-échantillonnage qui consiste à réduire le nombre d'observations de la classe majoritaire en sélectionnant aléatoirement des échantillons (il supprime des instances au hasard). Cette technique peut avoir des inconvénients car elle peut éliminer des informations utiles lors du sous-échantillonnage [1].

- ✓ **Tomek link removals**

  Tomek link removal est une technique de sous-échantillonnage qui consiste à supprimer les paires de points de deux classes différentes qui sont les plus proches voisins l'un de l'autre.5

$$\text{Soient } (E_1, E_2) \in X^2 \; est\; dite\; Tomek\; Link\; si \nexists\, E_3 \in X: \|(E_1, E_2)\| < \|(E_1, E_2)\|\; V\; \|(E_2, E_3)\| < \|(E_1, E_2)\|$$

  Tomek Link Removals consiste à supprimer tous les Tomek link en éliminant de chaque Tomek Link l'observation appartenant à la classe majoritaire.

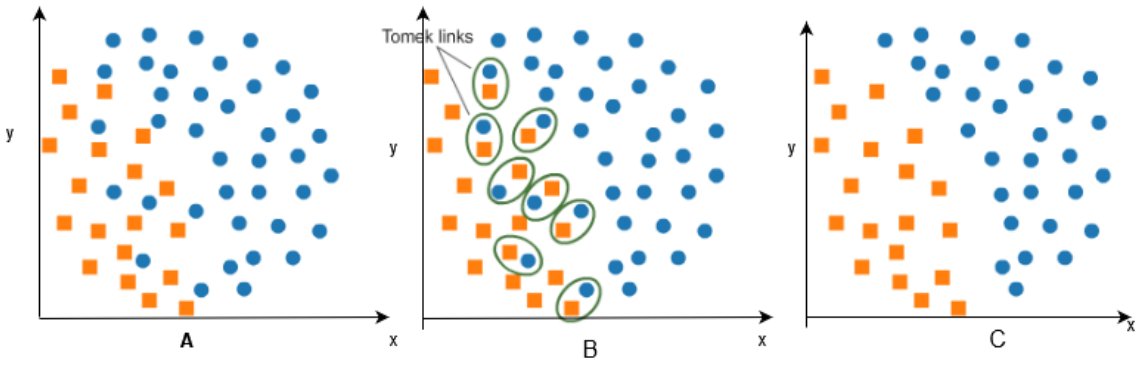

Figure II. 4: Représentation du modèle Tomek link removals

La figure II.4 représente les observations de deux classes différentes comme on peut le voir sur la figure II.4 **A**, la figure II.4 **B** représente les paires de Tomek link regroupé, quant à la figure **C** elle représente la représentation des classes après avoir supprimé tous les Tomek link en éliminant les observations de la classe majoritaire.

- ✓ **Sur-échantillonnage aléatoire**

  Le sur-échantillonnage aléatoire permet d'équilibrer un ensemble de données en dupliquant des exemples de la classe minoritaire jusqu'à ce qu'un ratio de classe souhaité soit atteint. Parmi ses inconvénients la possibilité d'overfitting [27].

- ✓ **SMOTE (Synthetic Minority over-sampling Technique)**

  Cette technique permet d'équilibrer les données provenant des diverses classes en agrandissant la taille de la classe minoritaire. Plus précisément, elle permet de créer de nouvelles instances à partir de celles de la classe minoritaire.

  SMOTE est plus efficace avec les données de faible dimension mais devient moins efficace lorsque les données sont de grande dimension car il n'atténue pas le biais de la classification dans la classe majoritaire [9].

3.2.2.2. Apprentissage ensembliste

Cet apprentissage rassemble plusieurs modèles et les combine afin d'obtenir une classification plus fiable des prédictions.

3.2.2.2.1. Bagging

- ✓ Le concept du bagging trouve son application dans le domaine du data mining prédictif, le mot Bagging est une contraction de Bootstrap Aggregation. Il utilise le bootstrapping qui est un processus d'échantillonnage aléatoire avec remise des données d'apprentissage, l'échantillonnage est effectué de telle sorte que chaque échantillon soit différent des autres.

- ✓ Le bagging utilise le même algorithme d'apprentissage automatique sur chacun des N échantillons obtenus par le bootstrapping afin d'entrainer les N modèles.
- ✓ Les prédictions obtenues des différents modèles sont ensuite agrégées (par vote) pour obtenir une classification plus précise.
- ✓ En pratique, la méthode de bagging donne d'excellents résultats notamment sur les arbres de décision utilisés en « forêts aléatoires ».
- ✓ L'avantage du bagging est qu'il permet de réduire la variance de l'estimateur, et permet donc de corriger l'instabilité des arbres de décision [6].

3.2.2.3. Méthodes basées sur le modèle gaussien

- **Introduction**

✓ Le modèle gaussien est un processus d'apprentissage automatique qui utilise la distribution gaussienne pour modéliser les données dénommées également Gaussian Processes for Machine Learning (GPML).

✓ Une loi normale est une loi de probabilité absolument continue qui dépend des paramètres ; un nombre réel noté μ sa moyenne, son écart type un nombre réel positif noté σ, La densité de probabilité de la loi normale est donnée formellement par :

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}}\, e^{-\frac{1}{2}\left(\frac{x-\mu\cdot}{\sigma}\right)^2}$$

3.2.2.3.2. Estimation des paramètres

✓ L'estimation des paramètres est une étape importante dans l'apprentissage automatique, Dans le cadre des modèles gaussiens, l'estimation des paramètres est effectuée en utilisant la fonction de vraisemblance.

✓ Ensemble de données : D= $\{x^{(1)}, \dots, x^{(m)}\}, x^{(i)} \in \mathbb{R}^n, x^{(i)} = ({x_1}^{(i)}, \dots, {x_n}^{(i)})$.

✓ Estimation des paramètres de chaque caractéristique j (dimension) : $x_J \sim N(\mu_j, \sigma_j^2), \mu_j = \frac{1}{m}\sum_{i=1}^{m}\left(x_j^{(i)} - \mu_j\right)^2$.

3.2.2.3.3. Algorithme de détection d'anomalies basé sur la distribution gaussienne

✓ Sélectionner les caractéristiques X_i les plus pertinentes (les plus indicatives d'anomalies).

✓ Soit le nouvel ensemble de données d'apprentissage :

$$D = \{x^{(1)}, \dots, x^{(m)}\}, x^{(i)} \in \mathbb{R}^n.$$

- ✓ Estimer/ adapter les paramètres $(\mu_1, \sigma^2), \dots, (\mu_n, \sigma_n^2)$.

$$x_j \sim N(\mu_j, \sigma^2).$$

$$\mu_j = \frac{1}{m}\sum_{i=1}^{m} x_j^{(i)}$$

$$\sigma^2 = \frac{1}{m}\sum_{i=1}^{m}\left(x_j^{(i)} - \mu_j\right)^2$$

- ✓ Pour un nouvel exemple $x = (x_1, \dots, x_n)$, calculer p(x) :

$$p(x) = \prod_{j=1}^{n} p(x_j; \mu_j, \sigma^2) = \prod_{j=1}^{n} \frac{1}{\sqrt{2\pi}\sigma_j} exp\left(-\frac{(x_j-\mu_j)^2}{2\sigma_j^2}\right)$$

- ✓ Si $p(x) < \varepsilon$ alors l'instance est une anomalie [19].

Problèmes :

- ✓ Une distribution peut ne pas suivre une distribution gaussienne.
- ✓ P(x) est comparable pour les instances normales et anomalies.

Solutions :

- ✓ Transformation d'une caractéristique d'une distribution non-gaussienne à une distribution gaussienne (exemple : la transformée de Box-Cox) ;
- ✓ Utilisation de la distribution gaussienne multivariée.

3.2.2.3.4. Distribution gaussienne multivariée

- ✓ La distribution gaussienne multivariée (également appelée distribution normale multivariée) est une distribution de probabilité qui décrit la distribution de plusieurs variables aléatoires continues corrélées les unes avec les autres. [4].
- ✓ Le principe consiste à ne pas modéliser la distribution des caractéristiques pertinentes $p(x_1), \dots, p(x_n)$ séparément, mais de modéliser $p(x), x \in \mathbb{R}^n$ en entier. Ainsi, les paramètres à estimer sont $\mu \in \mathbb{R}^n, \sum \in \mathbb{R}^{n \times n}$

✓ Soit l'ensemble des données d'apprentissage : $\{x^{(i)}, \dots, x^{(m)}\}$[4].

$$\mu = \frac{1}{m} \sum_{i=1}^{m} x^{(i)}$$

$$CV = \frac{1}{m} \sum_{i}^{m} \left((x^{(i)} - \mu) \left(x^{(i)} - \mu \right)^T \right)$$

$$CV = \begin{bmatrix} \sigma_1^2 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_n^2 \end{bmatrix}$$

CV= matrice de covariance.

3.2.2.4. Algorithme de détection d'anomalie dans le cas d'une distribution gaussienne multivariée

a) Adapter le modèle $\boldsymbol{p(x)}$ en estiment le paramètres $\boldsymbol{\mu}$ *et* $\boldsymbol{\Sigma}$

$$\boldsymbol{\mu = \frac{1}{m} \sum_{i=1}^{m} x^{(i)}}$$

$$\boldsymbol{\Sigma = \frac{1}{m} \sum_{i}^{m} (x^{(i)} - \mu)(x^{(i)} - \mu)^T}$$

b) Pour une nouvelle instance x, calculer $p(x)$:

$$\boldsymbol{p(x) = \prod_{j=1}^{n} p(x_j, \mu_j, \sigma_j^2) = \prod_{j=1}^{n} \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left(- \frac{(x_j - \mu_j)}{2\sigma_j^2} \right)}$$

c) Si $p(x) < \in$ alors x est une anomalie [4].

3.2.2.5. Méthodes basées sur la densité locale (LOF)

Local Outlier Factor (LOF) est un algorithme d'apprentissage automatique utilisé pour la détection d'anomalies dans une base de données.

On parle d'une valeur locale aberrante, quand un point est considéré comme aberrant de ses voisins [18].

Le LOF permet d'identifier les outliers en tenant compte de la densité des points avoisinant

Le LOF utilise les concepts suivants :

- K-distance et K-voisins
- Distance d'accessibilité (RD)
- Densité de Régularité Locale (LRD)
- Facteur D'anomalie locale (LOF)

a) K-distance et K-voisins

La distance K représente la distance entre le point et son k-ième voisin le plus proche. Les k-voisins, désignés par $N_k(A)$, comprennent un ensemble de points qui se trouvent à l'intérieur ou sur le cercle de rayon de la distance K. Le nombre de k-voisins peut être supérieur ou égal à la valeur de K. Comme le montre l'exemple ci-dessous :

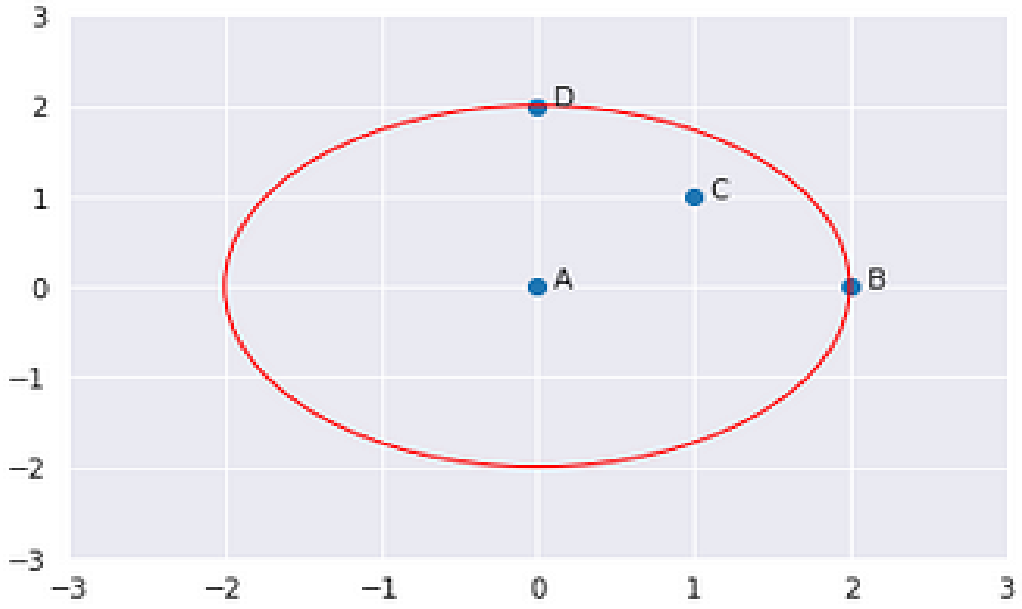

Figure II. 5*:* Représentation des N-neighbors de A un avec k-distance=2 [7]

Dans la figure II.5 nous avons les 4 points suivants ; A, B, C et D. On suppose que K=2, les k-voisins de A seront B, C et D. Ici, la valeur de K = 2, mais $||N_2(A)||$ est égal à 3. Par conséquent, $||N_k(point)||$ sera toujours supérieur ou égal à K.

b) Distance d'accessibilité (RD)

Définie comme étant le maximum entre la distance K de X_J et la distance entre X_i et X_j.

$$RD(X_i, X_j) = \max(k - distance(X_j), distance(X_i, X_j)$$

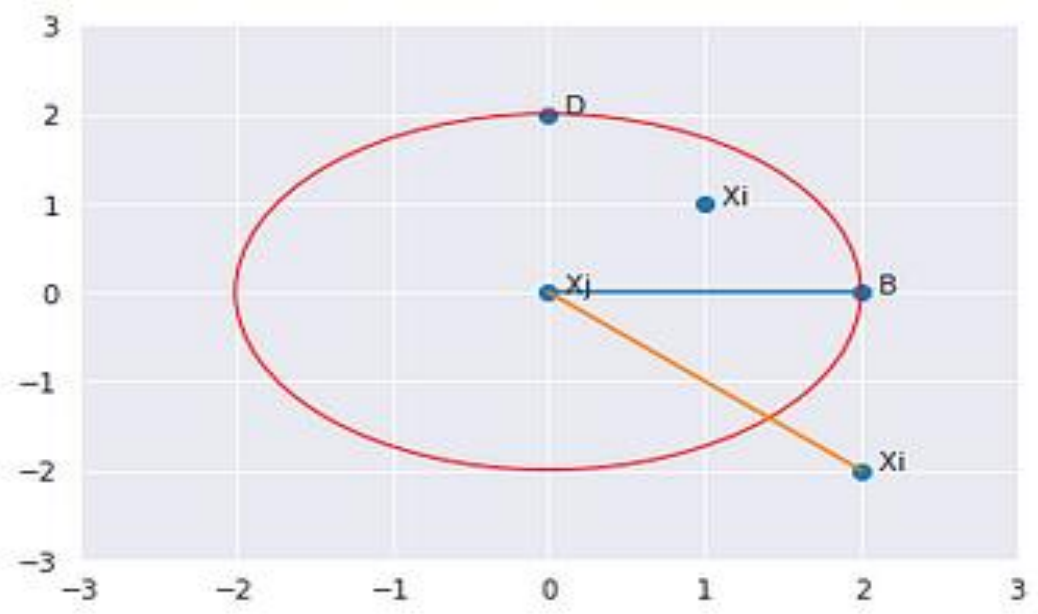

Figure II. 6: Représentation de la Distance d'accessibilité avec k=2 [7]

Comme la montre la figure II.6 si un point X_i se trouve parmi les k-voisins de X_j, la distance d'accessibilité sera la distance K de X_i, sinon la distance d'accessibilité sera la distance entre X_i et X_i.

c) Densité de Régularité Locale (LRD)

Le LRD est une mesure inverse de la distance moyenne d'accessibilité entre un point A et ses voisins. Selon la formule LRD, plus la distance d'accessibilité moyenne est grande (c'est-à-dire que les voisins sont éloignés du point), moins la densité de points autour d'un point donné est élevée. Cela indique à quelle distance un point se trouve du cluster de points le plus proche. Des valeurs faibles de LRD indiquent que le cluster le plus proche est éloigné du point.

$$LRD_K(A) = \frac{1}{\sum_{X_j} \in N_k(A) \frac{RD(A, X_j)}{|N_K(A)|}}$$

d) Facteur D'anomalie locale (LOF)

$$\boldsymbol{LOF_k(A) = \frac{\sum_{Xj \in N_k(A)} LRD_k(X_j)}{|N_K(A)|} \times \frac{1}{LRD_{k\,(A)}}}$$

Le LOF est le rapport entre la LRD moyenne des K voisins d'un point A et la LRD de ce point A. Si le point n'est pas une valeur aberrante, le rapport entre la LRD moyenne

des voisins est approximativement égal à la LRD du point (car la densité d'un point et celle de ses voisins sont généralement similaires). Dans ce cas, le LOF est proche de 1. En revanche, si le point est une valeur aberrante, la LRD du point sera inférieure à la LRD moyenne de ses voisins. Dans ce cas, la valeur du LOF sera élevée.

Les valeurs du facteur local d'aberration (LOF) permettent d'identifier une valeur aberrante en se basant sur le voisinage local. Il donne de meilleurs résultats que l'approche globale pour trouver les valeurs aberrantes [7].

3.2.2.6. Forêt d'Isolation

La Forêt d'Isolation est un algorithme d'apprentissage automatique utilisé pour la détection d'anomalies, il utilise des arbres de décision aléatoires pour isoler les anomalies dans les ensembles de données. Dans ces arbres, les partitions sont créées en sélectionnant d'abord aléatoirement une caractéristique, puis en choisissant une valeur de séparation aléatoire entre la valeur minimale et maximale de la caractéristique sélectionnée. Comme pour les autres méthodes de détection de valeurs aberrantes, un score d'anomalie est nécessaire pour la prise de décision. Dans le cas de l'Isolation Forest, il est défini comme suit :

$$S(x,n) = 2^{-\frac{E(h(x))}{c(n)}}$$

h(x) : représente la longueur du chemin de l'observation x dans l'arbre de l'Isolation Forest. Plus précisément, il s'agit du nombre d'arêtes que l'observation x doit traverser pour atteindre un nœud terminal dans l'arbre.

c(n) : représente la longueur moyenne d'un chemin dans un arbre binaire de recherche lorsqu'une recherche échoue.

n : est le nombre de nœuds externes dans l'arbre de l'Isolation Forest. Les nœuds externes sont les nœuds terminaux de l'arbre.

Plus le score d'anomalie d'une observation est proche de 1, plus celle-ci est considérée comme une anomalie. Un score inférieur à 0,5 indique que l'observation est probablement normale [16].

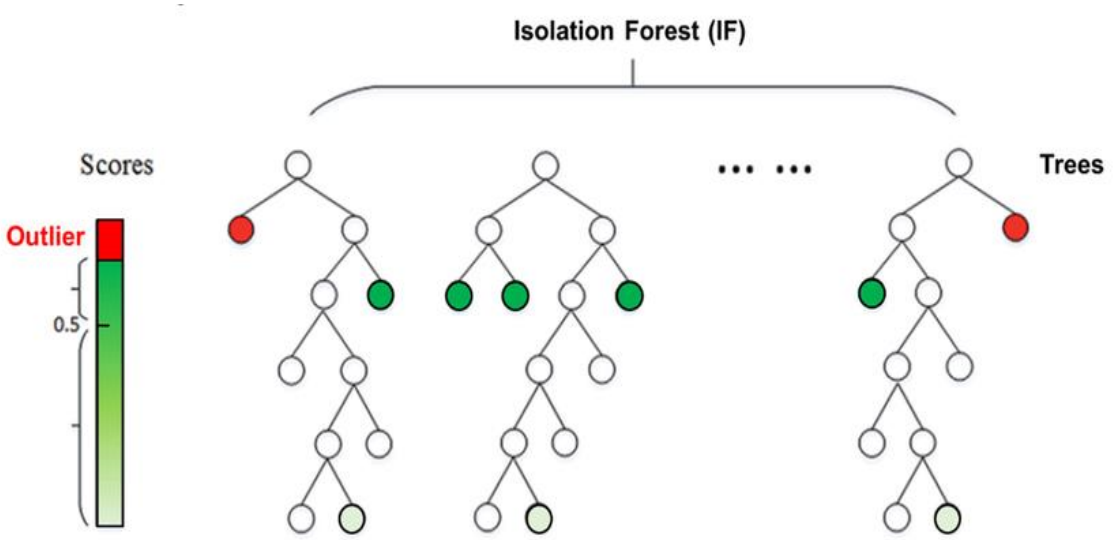

Figure II. 7:Représentation de la méthode isolation forest.

Comme on peut le voir sur la figure II.8, Les cercles clairs représentent des échantillons normaux communs, les cercles verts représentent des échantillons normaux inhabituels et les cercles rouges représentent les anomalies.

L'idée derrière l'Isolation Forest est que, en moyenne, les valeurs aberrantes sont plus proches du nœud racine avec une profondeur plus faible que les instances normales [16].

4. Métrique de performances

Une métrique de performances est une mesure qui permet d'évaluer la qualité d'un modèle ou d'un algorithme d'apprentissage automatique. Les métriques de performances permettent d'estimer la capacité d'un modèle à prédire correctement les sorties pour les données d'entrée, dans notre cas déterminer si une transaction est frauduleuse ou pas.

- ✓ **Matrice de confusion :** une matrice de confusion est un outil d'analyse prédictive qui permet d'évaluer la qualité d'un modèle de classification souvent binaire (pas forcement) en comparant les prédictions du modèle avec les vraies valeurs.

  La matrice de confusion dans notre cas est carrée de taille 2×2 où chaque ligne représente les instances dans une classe réelle et chaque colonne représente les instances dans une classe prédite, on les représente dans la détection de fraude sur un tableau comme suis :

Ground truth / Modèle	Négatif	Positif
Négatif (0)	Vrai négatif (True negative : TN)	Faux positif (False positive : FP)
Positif (1)	Faux négatif (False negative : FN)	Vrai positif (True positif :TP)

Tableau 1: Matrice de confusion

b) Le Vrais négatifs (TN) représente le nombre de transactions qui étaient légitimes et qui ont également été classées comme légitimes par le modèle.

- Le Faux négatifs (FN) représente le nombre de transactions qui étaient frauduleuses mais qui ont été incorrectement classées comme légitimes par le modèle.
- Le Faux positifs (FP) représente le nombre de transactions qui étaient légitimes mais qui ont été incorrectement classées comme transactions frauduleuses.
- Le Vrais positifs (TP) représente le nombre de transactions qui étaient frauduleuses et qui ont également été classées comme frauduleuses par le système [22].

✓ **Recall**

Le rappel également appelé sensibilité est une métrique de performance, qui mesure la proportion d'exemples positifs réels qui ont été correctement identifiés par le modèle dans l'ensemble des données. Le recall est important car il évalue la capacité d'un modèle de détection d'anomalies à trouver toutes les véritables valeurs aberrantes présentes dans un ensemble de donnée.

Le rappel est calculé comme le rapport entre le nombre de vrais positifs (TP) détectées et la somme des vrais positifs (TP) et des faux négatifs (FN) c'est-à-dire toutes les instantes.

$$\boldsymbol{Recall = \frac{TP}{TP + FN}}$$

✓ **Precision**

La précision est une valeur qui caractérise les performances d'un modèle en termes de classification des exemples de la classe positive. Contrairement au rappel, la précision concerne le nombre d'exemples que le modèle a étiquetés comme positifs et qui étaient vraiment positifs. Elle est calculée comme suis :

$$\boldsymbol{Precision = \frac{TP}{TP + FP}}$$

✓ **F1-Score**

F1-Score est utilisé dans les classifications binaires avec des classes déséquilibrées, il prend en considération le rappel et la précision. La valeur de F1-Score varie entre 0 et 1, elle est calculée comme suis : $$\boldsymbol{F1score = \frac{2\times(Precision\times Recall)}{Precision+Recall}}$$

✓ **Specificity**

La spécificité, également appelée taux de vrais négatifs elle mesure le taux que le modèle a définis comme négatifs qui étaient vraiment négatifs [14]. Elle est calculée comme suis :

$$\boldsymbol{Specificity = \frac{TN}{TN+FP}}$$

✓ **Area under Precision-recall curve**

L'aire sous la courbe de précision-rappel (ou AUPRC) est utiliser pour mesurer la performance des modèles. Il est le compromis entre la précision et le rappel, par rapport aux autres métriques l'AUPRC est insensible aux distributions de données.

L'AUPRC est calculée en calculant l'aire sous la courbe du tracé précision-rappel, qui est obtenu en traçant le rappel sur l'axe des x et la précision sur l'axe des y pour différents seuils de probabilité [28].

5. Conclusion

Dans ce chapitre nous avons présenté le fonctionnement des modèles d'apprentissage automatique utilisé pour la détection d'anomalies dans la fraude par carte de crédit. Nous avons abordé le problème de déséquilibre de classes et les solutions utilisées pour y remédier. Nous avons également cité quelques métriques de performances.

Chapitre III

Implémentations des méthodes et Evaluation des performances

1. Introduction

Après avoir défini les concepts théoriques liés à la fraude bancaire et à ses méthodes de détection, ainsi que tous les modèles d'apprentissage automatique nous expliquons dans ce chapitre, les différents algorithmes utilisés pour la détection d'anomalie dans une base de données déséquilibré.

L'objectif de cette étude est de trouver un modèle qui permet aux banques détectées la fraude bancaire avec des méthodes basée sur l'apprentissage automatique en prenant en compte une base de données déséquilibré.

2. Environnement de développement

PYTHON :

Python étant un langage de programmation très répandu et facile à apprendre de par sa syntaxe lisible et sa variété d'usages, il use un style de programmation ouvert et évolutif. Python est un langage polyvalent utilisé dans de nombreux domaines [29], dans le cadre de ce mémoire la version utilisée est 3.9.13.

Spyder :

Spyder est un environnement de développement intégré (IDE), il est conçu pour la programmation scientifique et l'analyse de données leurs exécutions et le débogage du code en utilisant Python, il facilite l'exploration, l'analyse et la visualisation des données. Spyder est inclus dans la distribution Anaconda, ce qui facilite son installation et son utilisation avec les autres packages et outils inclus dans Anaconda [20].

Anaconda est une distribution libre et open source des langages de programmation Python disponible pour MacOS, Linux et Microsoft Windows, elle comprend un ensemble de packages et d'outils populaires parmi eux on cite spyder. Il est généralement appliqué au développement d'applications dédiées à la science des données et à l'apprentissage automatique [29].

Figure III. 1: Logo d'anaconda

Figure III. 2: Logo de spyder

Pandas :

Pandas est une bibliothèque open-source pour la manipulation et l'analyse de données en Python. Elle offre des structures de données puissantes et flexibles, ainsi que des outils pour le nettoyage, la transformation, l'agrégation et la visualisation de données [20]. Dans le cadre de ce mémoire la version utilisée est 1.4.4.

Numpy :

NumPy (Numerical Python) est une bibliothèque python utilisée pour la manipulation des tableaux multidimensionnels, ainsi que des fonctions mathématiques efficaces pour effectuer des opérations numériques sur ces tableaux. Il a également des fonctions pour travailler dans le domaine de l'algèbre linéaire, de la transformée de Fourier et des matrices [29]. Dans le cadre de ce mémoire la version utilisée est 1.21.5.

Matplotlib :

Matplotlib est une bibliothèque du langage de programmation Python destinée à tracer et visualiser des données sous forme de graphiques domaine de l'analyse de données et de la science des données. Elle peut être combinée avec les bibliothèques python de calcul scientifique NumPy et SciPy.

Matplotlib offre une grande variété de graphes et de visualisations avec une précision de contrôle il s'intègre facilement avec les tableaux NumPy et les structures de données Pandas, ce qui facilite la création de graphiques à partir de données stockées dans ces formats [20]. Dans le cadre de ce mémoire la version utilisée est 3.5.2.

SciPy :

SciPy (Scientific Python) est une bibliothèque open-source en Python qui fournit des fonctionnalités avancées pour les calculs scientifiques et l'analyse de données. SciPy étend les fonctionnalités de NumPy en fournissant des outils avancés pour les calculs mathématiques, l'optimisation, le traitement du signal, les statistiques et bien d'autres domaines [20]. Dans le cadre de ce mémoire la version utilisée est 1.9.1.

Seaborn :

Seaborn est une bibliothèque Python destinée à la visualisation de données statistiques. Elle est basée sur Matplotlib, une autre bibliothèque de visualisation, elle simplifie la

création de graphes et propose des styles esthétiques par défaut [20]. Dans le cadre de ce mémoire la version utilisée est 0.11.2.

Sys :

Sys est un module intégré de la bibliothèque standard de Python, il fournit un accès à certaines fonctionnalités et paramètres spécifiques au système. Il permet d'interagir avec l'environnement d'exécution Python et de gérer des fonctionnalités système [20].

3. Data Set

On se base principalement sur les données bancaires, dans le cadre de notre étude, on utilise le data set credit card fraud detection :

- **Description de la base de données de Kaggle (Credit Card Fraud Detection) :**

Il est important pour les compagnies et les banques de déterminer si une transaction est frauduleuse ou pas. La base de données de Kaggle contient les transactions effectuées en Europe durant le mois de septembre durant 2 jours où nous constatons que 492 transactions sont frauduleuses parmi 284 807 transactions au totale, comme on le voit bien la base de données est déséquilibré la classe positive représente seulement 0.0172% du total [30].

Cette base de données contient seulement les variables numériques en entrée qui sont le résultat de l'analyse en composantes principales (PCA) qui permet de transformer des variables liées entre elles (corrélées) en statistique en nouvelles variables décorrélées les unes des autres nommées composantes principales. Pour des raisons de confidentialité les caractéristiques originales et les informations sur les bases de données ne sont pas fourni. Les caractéristiques V1, V2, … V28 sont les principales composantes qui décrivent les transactions obtenues avec PCA, les seules caractéristiques qui n'ont pas été transformés sont le temps (qui représente les secondes écoulées entre chaque transaction et la première transaction dans l'ensemble de données) et le montant (qui est le montant de la transaction).

La classe est la variable cible indiquant si la transaction est frauduleuse (1) ou valide (0) [30].

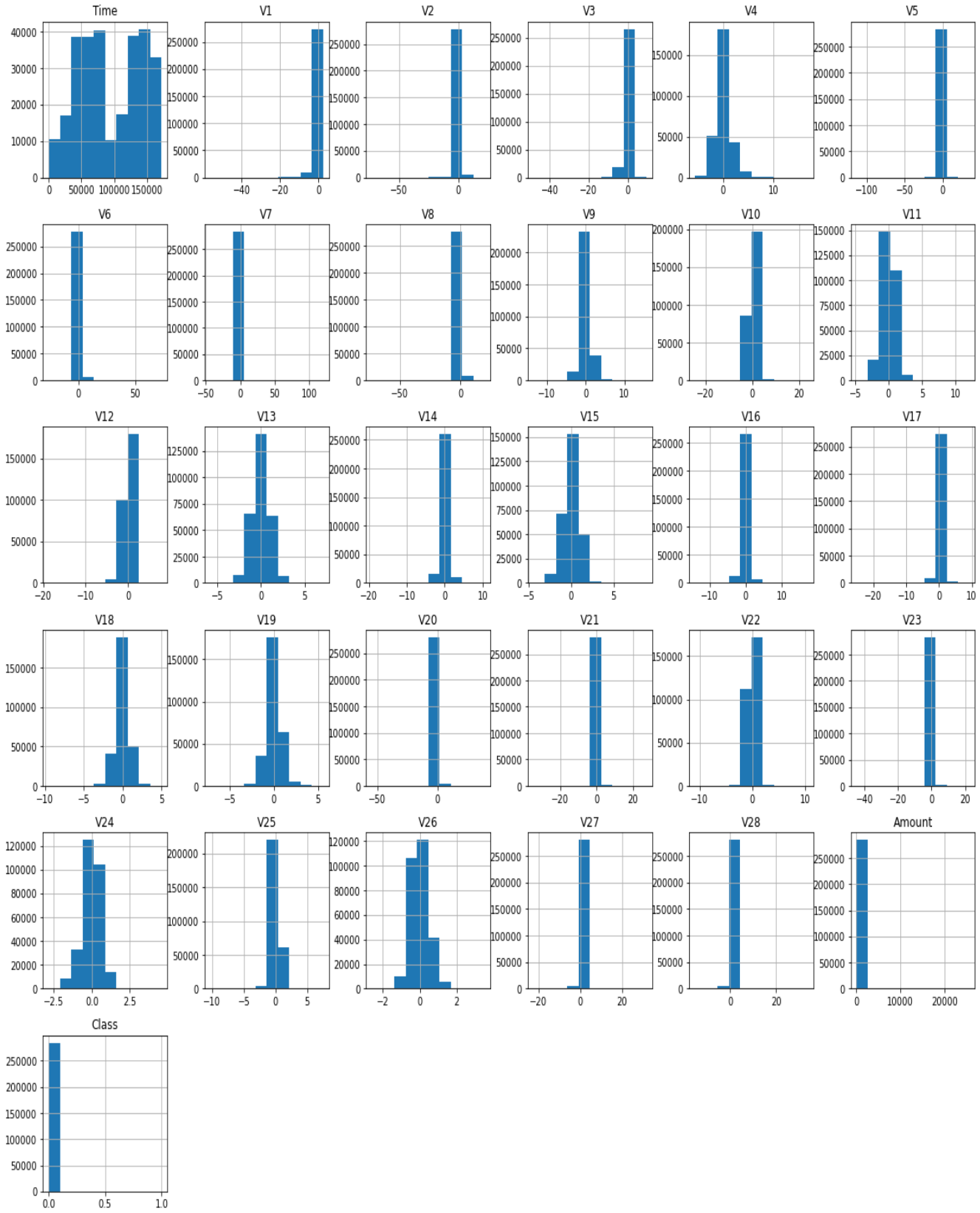

Figure III. 3: Histogrammes de distribution des différentes caractéristiques dans les données

Comme on peut le voir sur la figure III.3 toutes les caractéristiques suivent principalement une distribution gaussienne et celles qui ne le sont pas, sont sous forme fat tail (queue grasse) qui peuvent être transformé en une distribution gaussienne.

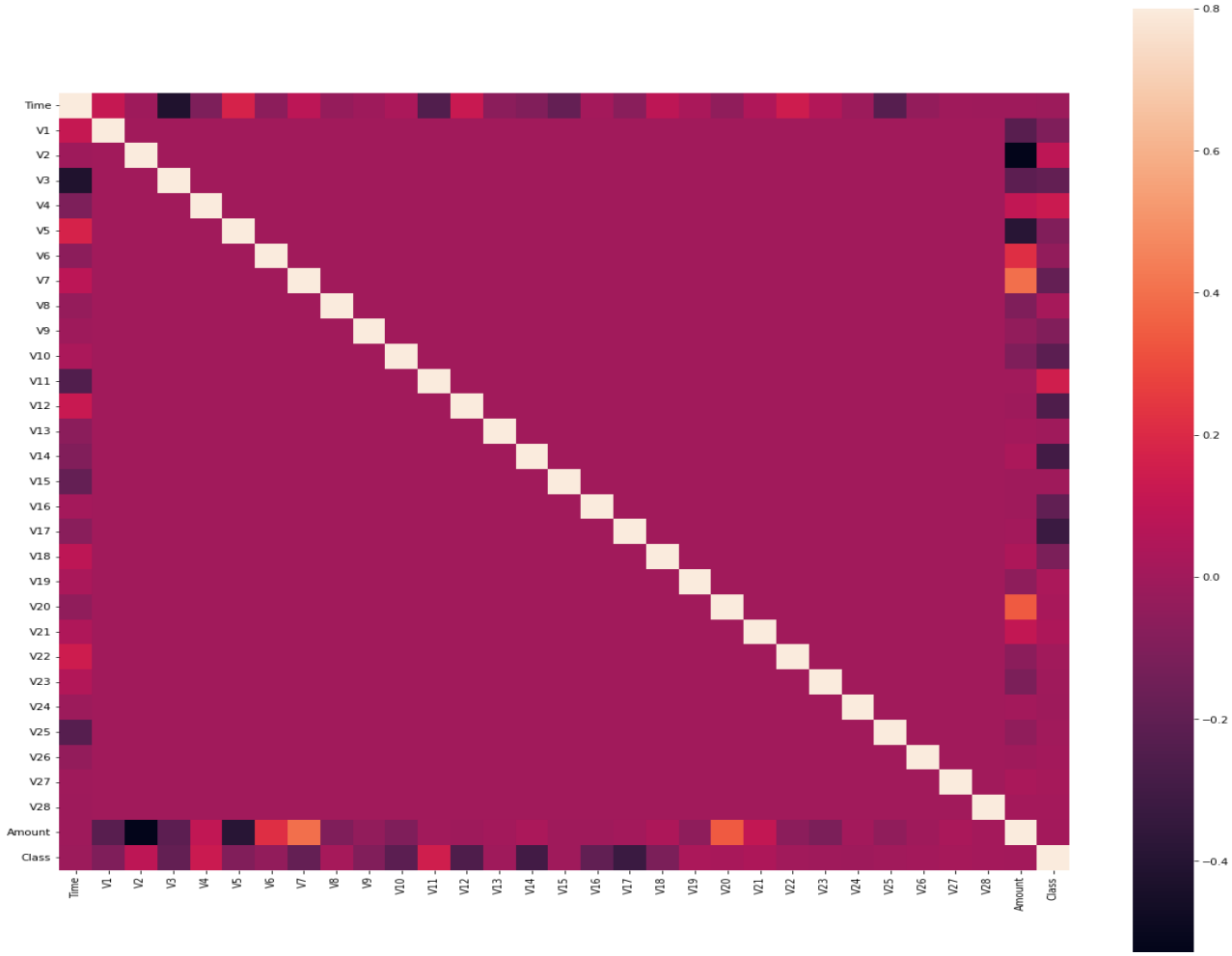

Figure III. 4: Heatmap représentant la corrélation entre les différentes caractéristiques dans les données

Comme on peut le voir la figure III.4, une heatmap qui représente la corrélation entre les différentes caractéristiques avec une échèle de couleurs de la claire pour les décorrélées au foncé pour les plus corrélées. On peut voir que la majorité des caractéristiques sont décorrélés entre elles ce qui facilite notre étude.

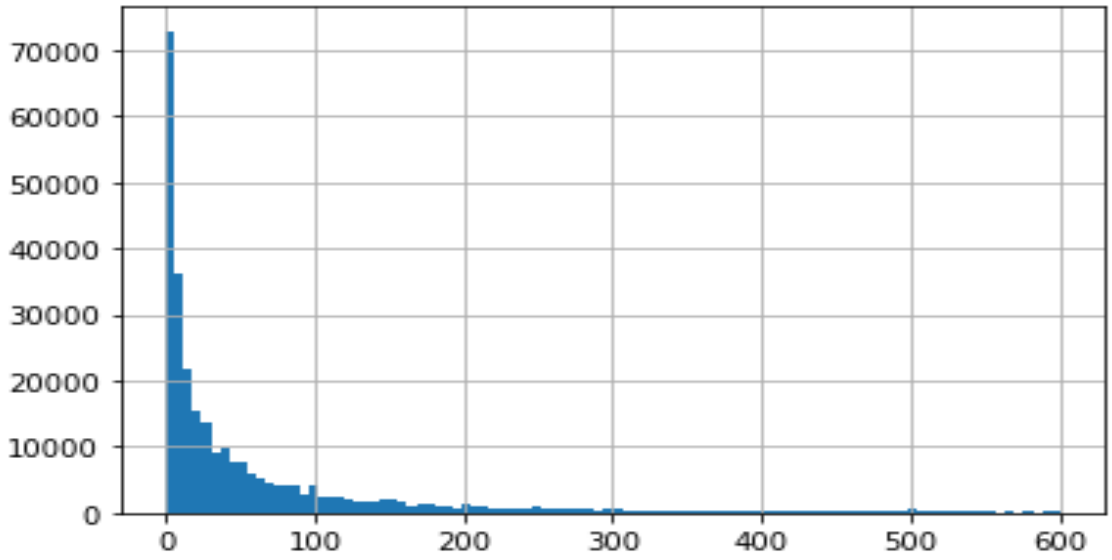

Figure III. 5:Histogramme représentant la distribution des montants des transactions valides

Comme la montre la figure III.5 la plupart des transactions valides sont inférieure à 500$, car pour la plupart des transactions, elles sont faites pour des achats de la vie quotidienne qui n'excède pas les 500$.

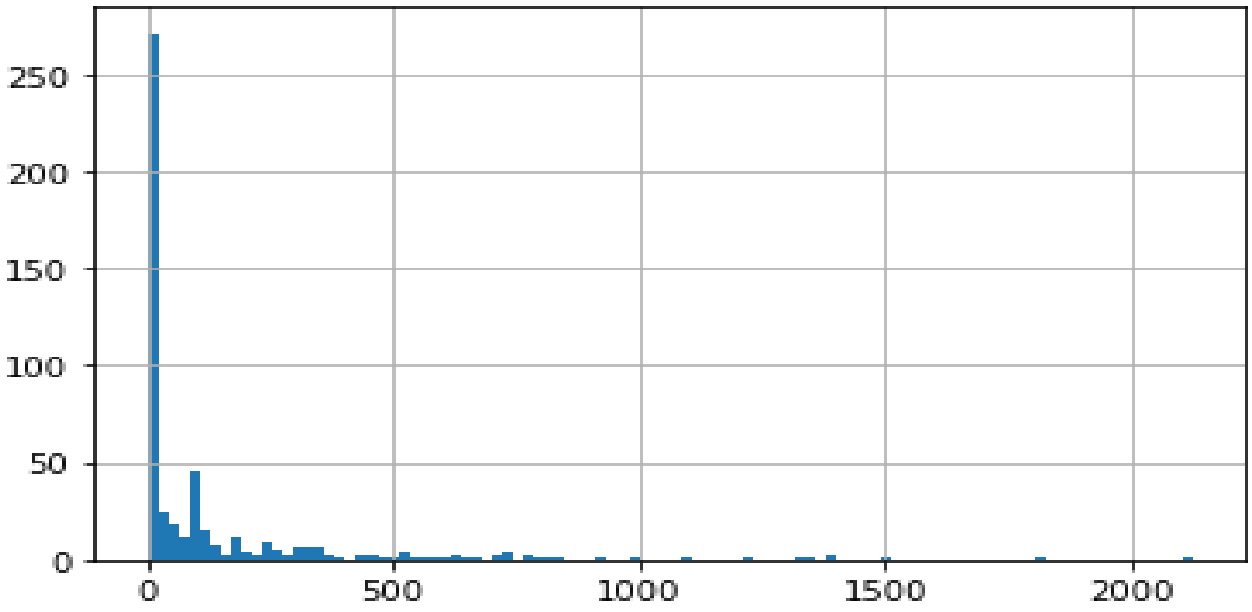

Figure III. 6:Histogramme représentant la distribution des montants des transactions frauduleuses

Comme on peut le voir sur la figure III.6 les transactions frauduleuses sont accentuées sur les petites montants la plupart sont inférieure à 500$ tout comme les transactions valides.

- **Prétraitement de la base de données :**

On a une base de données de 284807 échantillons parmi eux 284315 sont des transactions valides et 492 sont des transactions frauduleuses avec le traitement de données on divise la base de données en 60% pour l'apprentissage, 20% pour le test et 20% pour la validation après avoir préparé, nettoyé et mélangé les données de manière aléatoire.

N_test = 93823 représente les instances dans l'ensemble test qui sont utilisé pour évaluer les performances du modèle de détection de fraude.

N_outlier = 246 représente les instances frauduleuses dans l'ensemble test.

X_train shape = (96669, 28) représente la forme de l'ensemble d'entraînement contenant des instances valides.

X_valid shape = (94069, 28) représente l'ensemble de validation utilisé pour ajuster le seuil de détection de fraude

X_test shape = (94069, 28) représente la forme de l'ensemble de test utilisée pour évaluer les performances du modèle de détection.

4. Méthodes implémentées

Nous avons utilisé deux approches pour contourner le problème de déséquilibre de classes la première basée sur le re-échantillonnage et l'utilisation des algorithmes classiques (SVM, KNN et DT) et la deuxième basé sur l'utilisation d'algorithme appropriés au problème de classe (LOF, Gaussienne multivariée et isolation forest).

4.1 méthodes basé sur le re-échantillonnage

On utilise SMOTE qui permet d'effectuer un sur-échantillonnage en augmentent le nombre d'instance positifs. Il génère de nouvelles instances synthétiques de la classe minoritaire en interpolant les caractéristiques des instances existantes comme suit :

On sélectionne d'abord une instance de la classe minoritaire (un point) à partir de l'ensemble de données, on cherche ses k voisins les plus proches parmi les instances de la classe minoritaire, on les sélectionne aléatoirement l'un des k voisins les plus proches. On générer une instance (un nouveau point) entre l'instance sélectionnée et son voisin choisi au hasard en effectuant une interpolation linéaire dans l'espace des caractéristiques. Enfin on répète les étapes plusieurs fois.

Apres avoir rééquilibré les classes nous pouvant maintenant utiliser les algorithmes d'apprentissage automatiques standard pour la détection d'anomalies.
Avec l'utilisation de 0.1 de la base donnée nous trouvons les résultats suivants :

- **Les résultats obtenus avec le SVM** :

Pour Les deux classes	**Spécificité**	**Recall**	**F1 score**	**Support**
0	1.00	0.98	0.99	8530
1	0.07	0.87	0.13	15

Tableau 2: Comparaison des résultats des différentes métriques pour le SVM

- **Les résultats obtenus avec le KNN :**

Pour Les deux classes	**Spécificité**	**Recall**	**F1 score**	**Support**
0	1.00	0.98	0.99	8530
1	0.07	0.87	0.13	15

Tableau 3: Comparaison des résultats des différentes métriques pour le KNN

- **Les résultats obtenus avec la DT :**

Pour Les deux classes	**Spécificité**	**Recall**	**F1 score**	**Support**
0	1.00	0.98	0.99	8530
1	0.07	0.87	0.13	15

Tableau 4: Comparaison des résultats des différentes métriques pour la DT

Les résultats pour le KNN sont similaires à ceux du SVM et ceux de la DT.

D'après les tableaux 1,2 et 3 les trois modèles (SVM, KNN et Decision Tree) ont des performances élevées pour la classe des transactions valides (classe 0). Pour la classe des transactions frauduleuses (classe 1), les modèles présentent une spécificité faible de 0,07, ce qui indique qu'ils ont mal classés de nombreux exemples de cette classe en tant que classe négative. Les modèles ont obtenu un rappel élevé de 0,87 pour la classe 1, ce qui signifie qu'ils ont réussis à trouver la grande majorité des exemples de cette classe. En termes de F1 score, qui est une mesure globale de performance, les modèles ont obtenu une valeur de 0,13 ce qui indique une performance relativement faible pour cette classe spécifique.

Les modèles ont trouvé 8530 transactions valide et 15 transactions frauduleuses.

4.2. Méthodes basées sur les algorithmes spécialisé pour les classes déséquilibré

- **Facteur d'Anomalie Locale (LOF) :** est un algorithme d'apprentissage non supervisé utilisé pour la détection des valeurs aberrantes dans les données. Il est basé sur l'écart local de densité d'un échantillon donné par rapport à ses voisins, en comparant la densité locale d'un échantillon aux densités locales de ses voisins, les instances ayant des densités inférieures à leurs voisins seront considérés comme des instances aberrantes (outliers) [7].

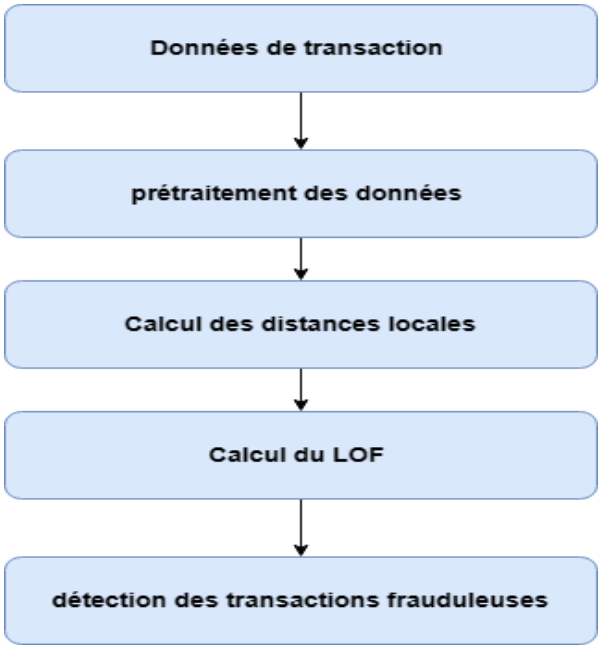

Figure III. 7:Représentation des différentes étapes de la méthode LOF

- **Forêt d'Isolation :** c'est une méthode d'apprentissage automatique utilisée dans la détection des valeurs aberrantes, par ailleurs la détection des fraudes dans un jeu de données. L'isolation Forest permet d'isoler les observations anormales en les séparant du reste des données normaux [26].

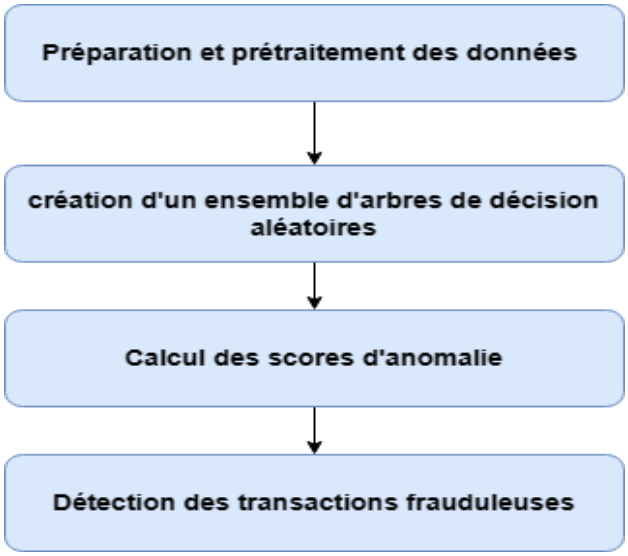

Figure III. 8:Représentation des différentes étapes de la méthode isolation forest

- **Gaussienne Multivariée** : la distribution gaussienne multivariée est largement utilisée en statistiques et en apprentissage automatique pour modéliser des ensembles de données multidimensionnels elle est définie par un vecteur moyen (μ) qui représente le centre de la distribution et par une matrice de covariance (Σ) qui décrit la relation entre différentes dimensions des variables aléatoires [4].

  La fonction de densité de probabilité (PDF) de la distribution gaussienne multivariée est donnée par l'expression :

$$f(x) = (2\pi)^{\left(\frac{-k}{2}\right)} * |\Sigma| * \exp(-0.5 \times (x-\mu)^T * \Sigma^{-1} * (x-\mu))$$

  Où :

  K : est le nombre de variable de la distribution

  X : est le vecteur des variables aléatoire

  $|\Sigma|$: est le déterminant de la matrice de covariance Σ

  T : est la transposée

  Σ^{-1} : est l'inverse de la matrice de covariance Σ

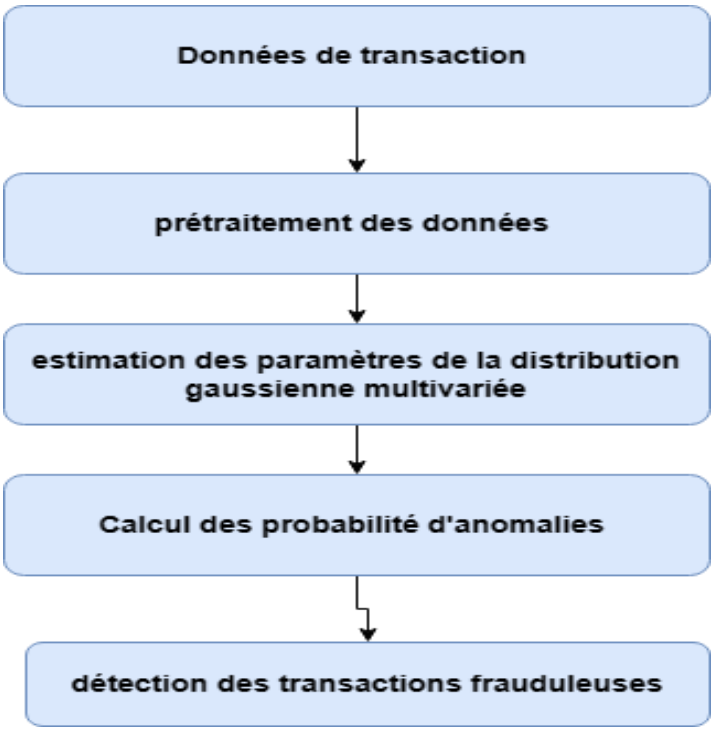

Figure III. 9:Représentation des différentes étapes de la méthode gaussienne multivariée

4.2.1 Les métriques utilisées :

$$Recall = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$F1score = \frac{2 \times (Precision \times Recall)}{Precision + Recall}$$

Avec :

TP : Vrais positif

FN : faux négatif

FP : false positif

4.2.2. Implémentation des méthodes et analyse des résultats obtenus :

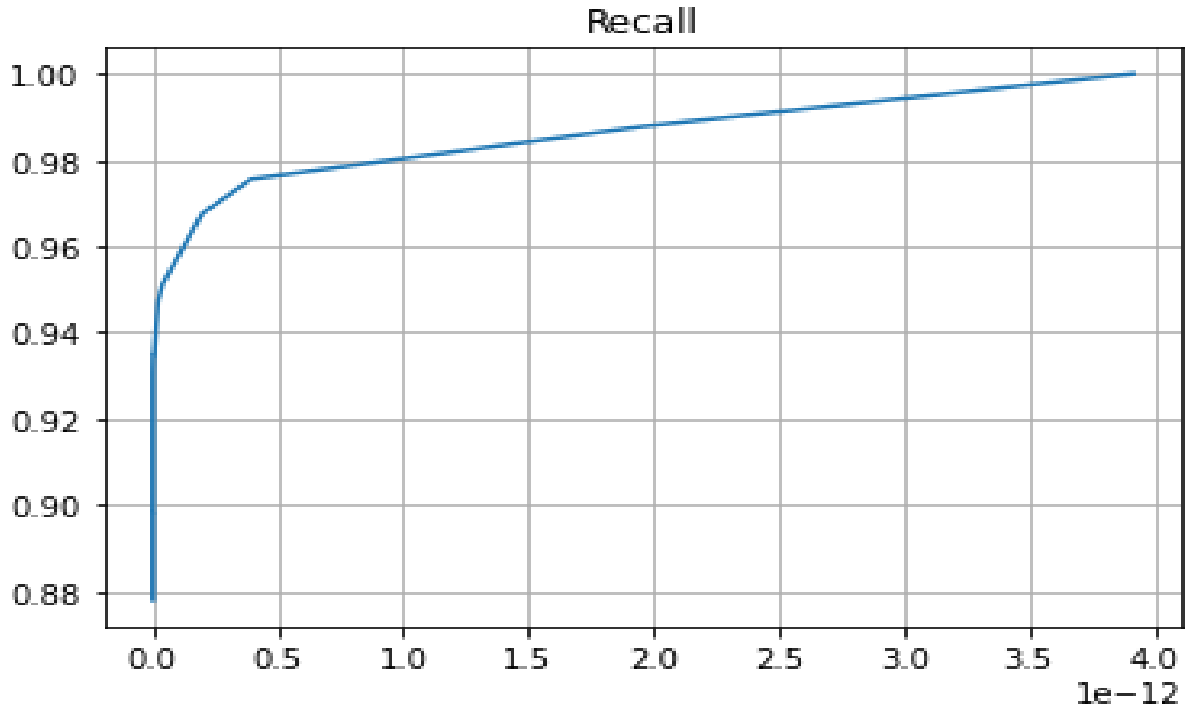

Figure III. 10: Graphe de variation du rappel (recall) en fonction du seuil ε

La figure III.10 montre que en choisissant ε à 10^{-12} le recall =1 , donc (100%) la meilleure performance.

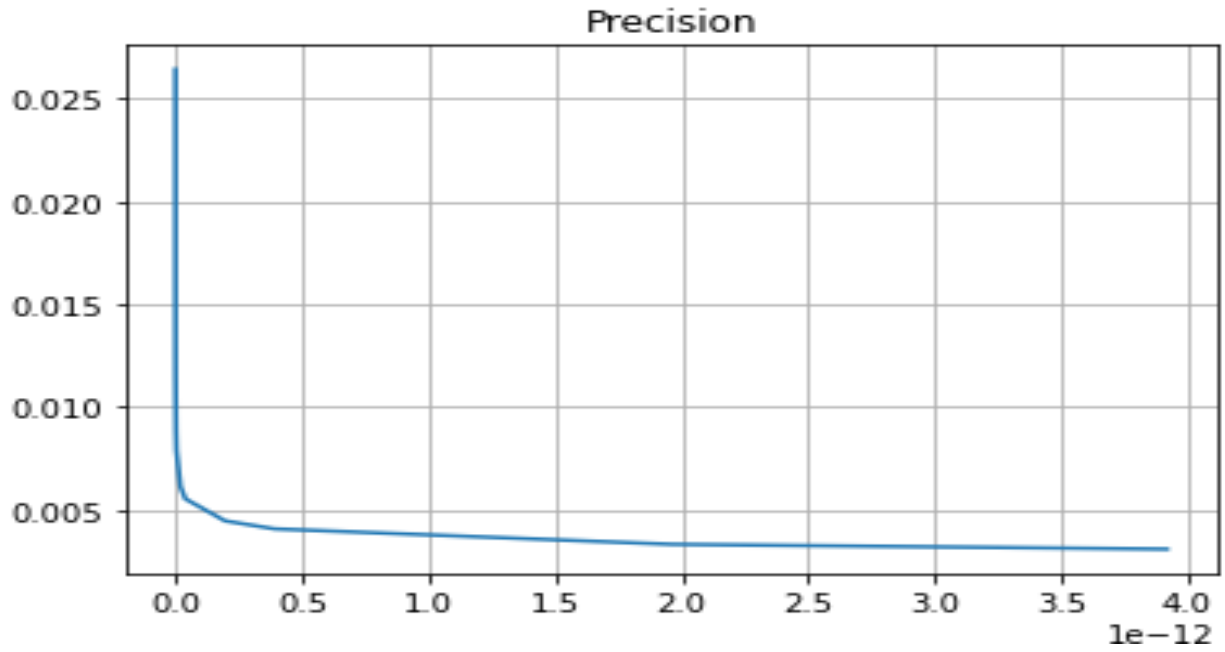

Figure III. 11: Graphe de variation de la précision en fonction du seuil ε.

La figure III.11 montre la variation du taux de la précision en variant le seuil ε.

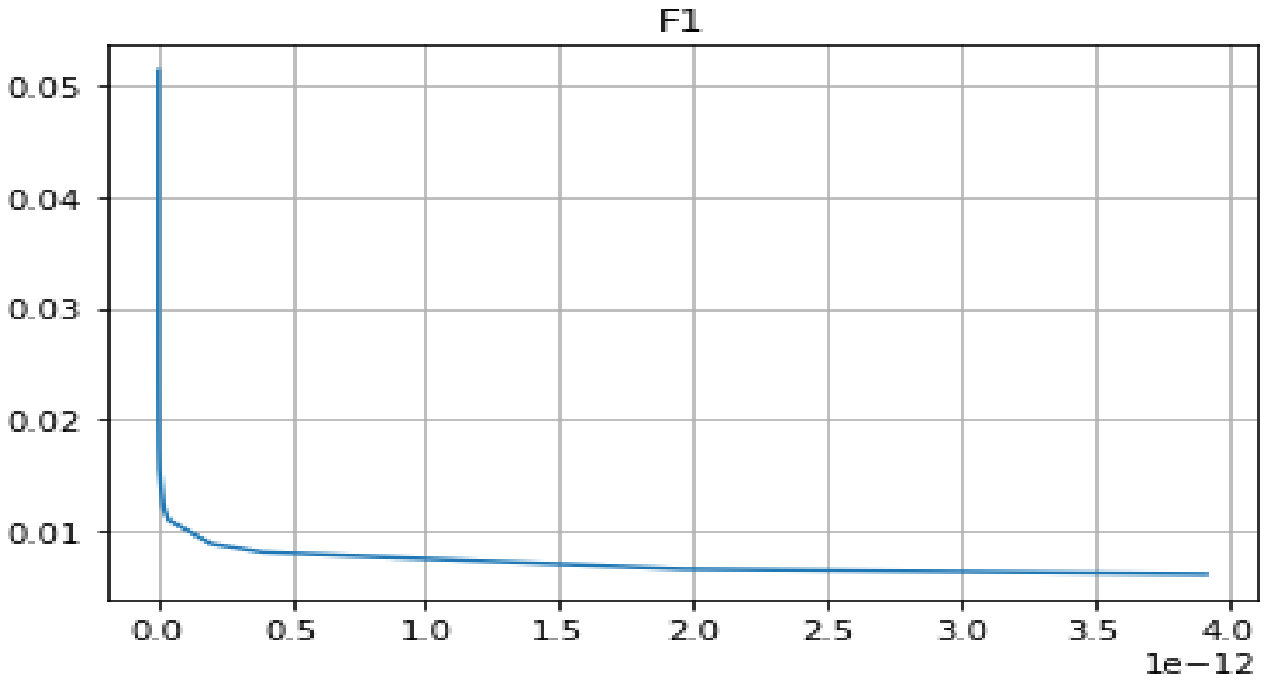

Figure III. 12: Graphe de variation du F1 score en fonction du seuil ε

La figure III.12 montre la variation du taux de F1 score en variant le seuil ε.

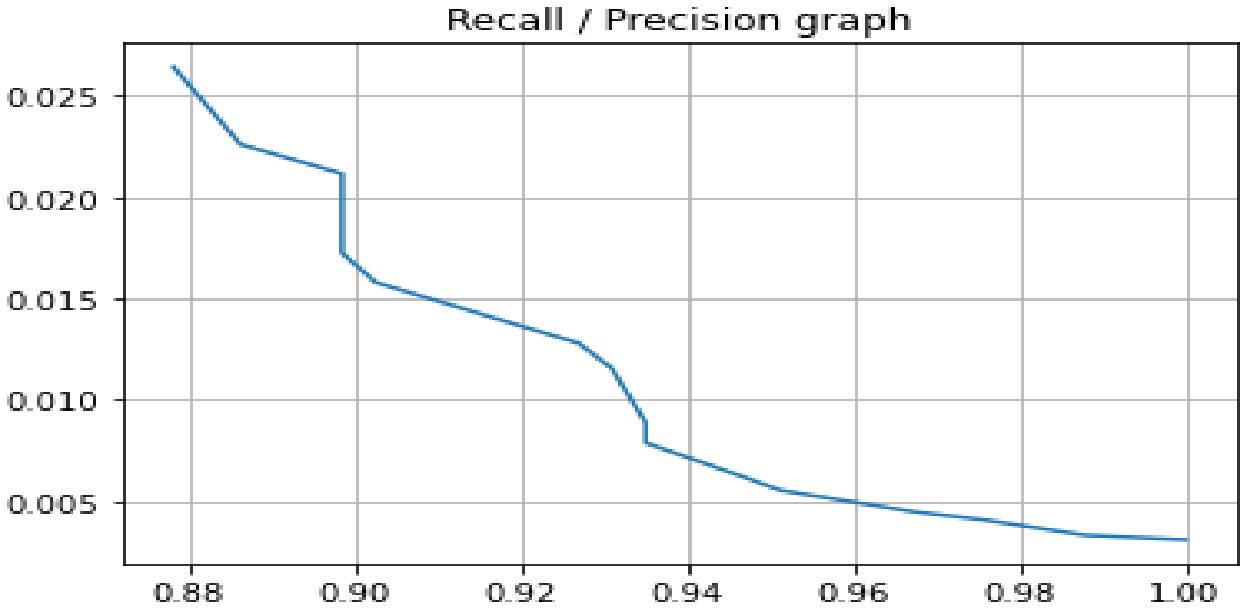

Figure III. 13: Graphe illustrant la relation entre le rappel et la précision (Recall / Precision) pour différent seuil.

La figure III.13 représente l'évolution du rappel et de la précision à mesure que le seuil de décision change, l'axe des abscisses représente le rappel, également appelé taux de vrais positifs. Tandis que l'axe des ordonnées représente la précision, qui est le nombre d'exemples positifs correctement classés par le modèle. Lorsque le seuil est optimal la précision est maximale tandis que le rappel est minimal.

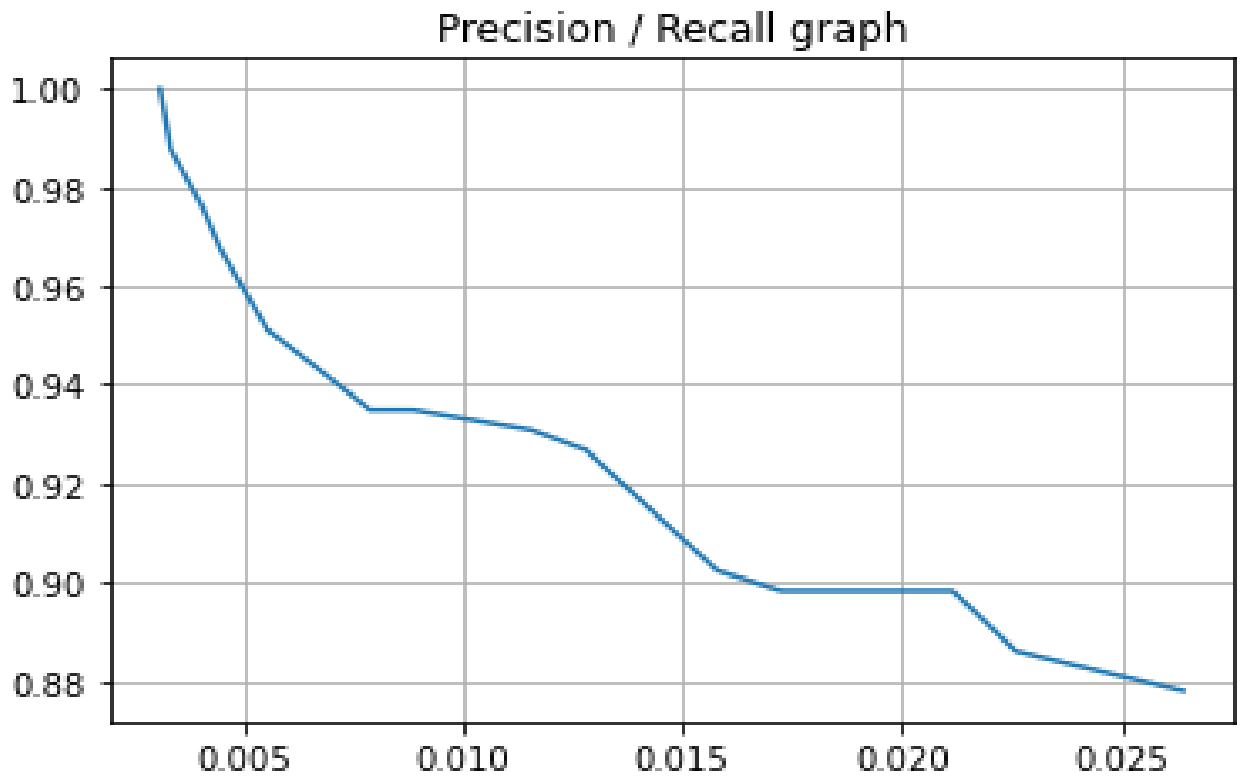

Figure III. 14:Graphe illustrant la relation entre le rappel et la précision (Precision/ Recall) pour différentes valeurs seuil.

La figure III.14 représente l'évolution de la précision et du rappel à mesure que le seuil de décision change, l'axe des abscisses représente la précision, qui est le nombre d'exemples positifs correctement classés par le modèle. Tandis que l'axe des ordonnées représente le rappel, également appelé taux de vrais positifs. Lorsque le seuil est faible le rappel est faible et la précision est maximal.

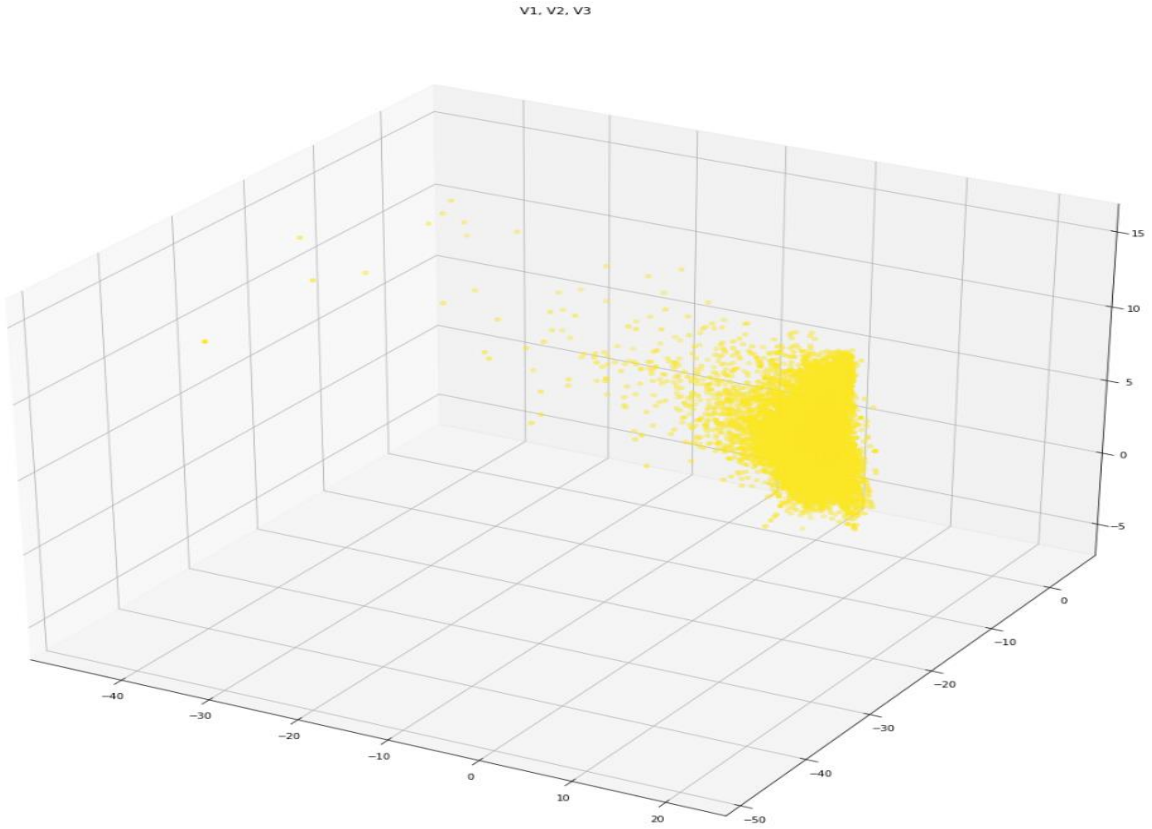

Figure III. 15:Nuage de points 3D avec les prédictions

- **Les résultats pour la gaussienne multivariée**

On aura les résultats suivants en choisissant le seuil ε = 3.9180295965029286e-12 :

Recall : 1.0 est un résultat excellant ce qui signifie que tous les cas positifs réels ont été correctement identifiés par le modèle il n'y a pas de faux négatifs.

Précision : 0.0030807378742908668, signifie que parmi toutes les transactions identifiées comme frauduleuses par le modèle de détection, seulement 0.31% sont réellement des fraudes, les autres sont des faux positifs.

F1 score : 0.006142552155511442, il combine à la fois le rappel (recall) et la précision, le faible score de 0.006142552155511442 indique que le modèle a du mal à trouver un équilibre entre la capacité à détecter les vrais positifs (rappel) et la capacité à minimiser les faux positifs (précision).

La méthode de la gaussienne multivariée est une méthode très efficace pour détecter les anomalies (transaction frauduleuse) car elle détecte tous les cas positifs mais elle présente également l'inconvénient de détecter les cas de faux positif ceux qui peux freiner certaines transactions en les classifiant comme frauduleuses alors qu'elles ne le sont pas

- **Les résultats pour Isolation Forest et local outlier factor :**

D'après les résultats suivants :

Isolation Forest : 647

Accuracy score : 0.9977282861727416

- ✓ Le modèle isolation forest a prédit 647 erreurs, il a prédit 647 anomalies alors que ce sont des transactions valide.
- ✓ L'accuracy score qui mesure la précision indique une performance élevée étant de 0.99.

Pour Les deux classes	Spécificité	Recall	F1 score	Support
0	1.00	1.00	1.00	284315
1	0.34	0.34	0.34	492

Tableau 5: Comparaison des résultats des différentes métriques

- ✓ La précision qui permet de calculer taux des transactions frauduleuses prédite par rapport aux nombres total de transactions frauduleuses étant seulement de 34% de vrais positif.
- ✓ Le taux du rappelle et de F1 score étant de 34% ce qui signifie que le modèle a une capacité limitée à détecter les cas de fraude.
- ✓ Bien que l'accuracy soit de 1.00, les résultats suggèrent que le modèle peut être efficace dans la détection des cas de frauduleux.
- ✓ Les deux modèles Isolation Forest et Local Outlier Factor présentent de faibles performances pour les métriques car le modèle a du mal à identifier les transactions frauduleuses.

4.2.3. La comparaison

Une comparaison entre les performances des trois techniques (gaussienne multivariée, local outliers factor et isolation forest).

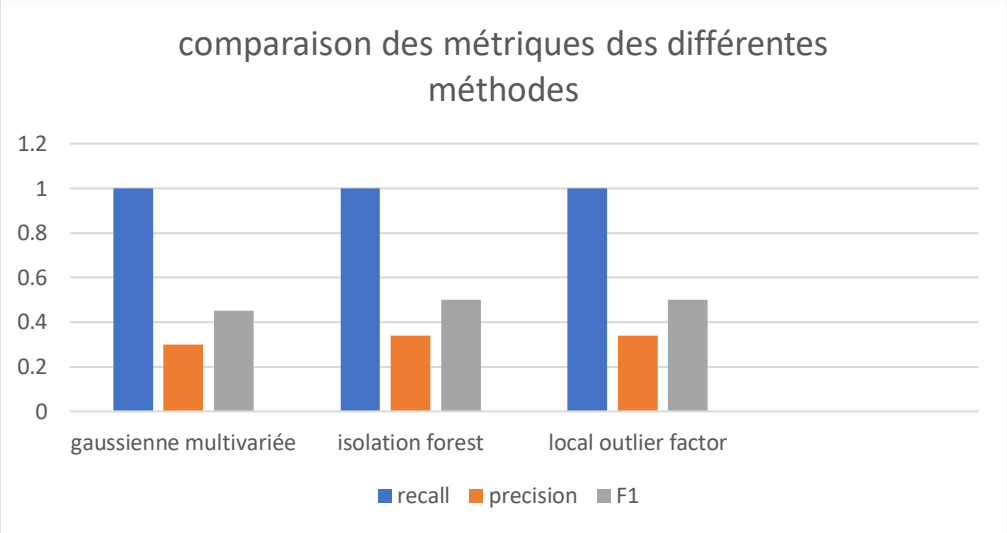

Figure III. 16**:: Histogramme comparative des différentes métriques pour chaque méthode utilisée**

La détection d'anomalies avec la gaussienne multivariée est basée sur la modélisation de distribution des données, tandis que l'Isolation Forest et le Local Outlier Factor est une approche basée sur des mesures de séparation ou de densité des données. Chaque technique a ses propres forces et faiblesses, et le choix dépendra des caractéristiques des données et du contexte spécifique de détection d'anomalies.

Dans notre cas la meilleure technique ou le modèle qui offre de meilleures performances est le modèle gaussien.

4.2.4. Variation du nombre de k-voisins pour Local Outlier Factor :

En variant la valeur des n_neighbors pour n= 6, n=12, n=20 et n=24 on remarque que les résultats restent les mêmes, en trouve des performances similaires dans la détection des transactions frauduleuses, avec une précision et un rappel d'environ 34 %. Dans certain cas choisir une plus petite de valeur de **n_neighbors** qui signifie que seuls quelques voisins proches seront pris en compte pour évaluer la densité locale peut conduire à une détection d'anomalies plus sensible et détecter des anomalies locales plus petites ou isolée mais pas dans notre cas.

5. Conclusion :

Dans ce chapitre nous avons etudié deux approches l'une en utilisant le sur-échantillonnage avec smote et les modèles classique (SVM, KNN et DT) et l'autre en utilisant trois méthodes compatible avec le déséquilibre de classe (LFO, gaussien multivariée et isolation forest) pour la détection d'anomalies, puis nous avons interprété leurs résultats et étudiés les performances de chaque méthode, enfin nous avons comparé entre les résultats des différentes métriques pour chaque méthode.

Dans notre cas les méthodes basées sur le sur-échantillonnage en utilisant partiellement la base de données donnent des résultats très faibles pour la détection des transactions frauduleuses, seulement 0.7% de spécificité. Tandis que les autres méthodes donnent de meilleurs résultats plus exactement le modèle gaussien qui donnent les meilleurs résultats avec une spécificité de 34% qui représente un excellent taux.

Conclusion Générale

Conclusion Générale :

La détection des transactions frauduleuses par cartes de crédit grâce à l'utilisation d'algorithmes d'apprentissage automatique représente une avancée majeure dans la lutte contre la fraude financière. Elle contribue à renforcer la confiance dans les transactions financières et à protéger les consommateurs ainsi que les institutions contre les pertes causées par la fraude.

Avec l'utilisation d'algorithmes sophistiqués pour analyser et identifier les schémas et les comportements anormaux dans les données de transaction. Les modèles d'apprentissage automatique sont entraînés à partir d'un large éventail de données, comprenant à la fois des transactions légitimes et frauduleuses. Ces modèles sont capables de repérer des schémas subtils et des signaux d'alerte qui pourraient indiquer une activité frauduleuse.

Dans le cadre de ce PFE nous avons proposé différentes méthodes : avec sur-échantillonnage (SVM, KNN, DT) et d'autres sans échantillonnage préalable comme ; Le Facteur d'Anomalie Locale, la gaussienne multivariée et la Forêt d'Isolation, afin d'obtenir les meilleurs résultats pour la détection de transactions frauduleuses dans une base de données à 284 807 transactions.

Les résultats montrent une bonne performance de notre approche en matière de détection de transactions frauduleuses, bien que les performances en précision des modèles soit différents, on constate que notre approche permet d'avoir d'excellent résultats avec le modèle gaussien.

Cependant, il est important de noter que la détection de transactions frauduleuses par cartes de crédit reste un défi constant. Les fraudeurs adaptent constamment leurs techniques pour échapper aux systèmes de détection, ce qui nécessite de développer des stratégies et des technologies pour contrer les fraudeurs en constante évolution. Et compte tenu des délais, nous n'avons pu analyser l'ensemble des modèles d'apprentissage automatique existants il nous semblerait intéressant, dans l'avenir, de les explorer.

Bibliographie :

1. Aitken, A. C. (1935). Note on selection from a multivariate normal population. Proceedings of the Edinburgh Mathematical Society, 4(2), 106-110.
2. Avoce, J., J. (novembre 2021). Apprentissage profond distribué sécurisé : application à la détection de fraudes bancaire sur internet. Université du QUÉBEC EN OUTAOUAIS.
3. Benzaki, Y. (2018). Introduction à l'algorithme k Nearest Neighbors (KNN). Mr.Mint: Apprendre le Machine Learning de A à Z, 2.
4. Bishop, C. M., & Nasrabadi, N. M. (2006). Pattern recognition and machine learning (Vol. 4, No. 4, p. 738). New York: springer.
5. Bounie, D., Bourreau, M. (avril 2004). Sécurité des paiements et développement du commerce électronique. Revue économique, vol.55, 689-710.
6. Breiman, L. (1996). Bagging predictors. Machine learning, 24, 123-140.
7. Breunig, M. M., Kriegel, H. P., Ng, R. T., & Sander, J. (2000, May). LOF: identifying density-based local outliers. In Proceedings of the 2000 ACM SIGMOD international conference on Management of data (pp. 93-104).
8. Chattopadhyay, A., Mishra, S., Gonzàlez-Briones, A. (2021). Integration of machine learning and iot in healthcare domain. In Hybrid artificial intelligence and IoT in healthcare. Springer 223-244.
9. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16, 321-357.
10. Chergui, S., Souici, L. (2021/2022). Méthodes d'apprentissage automatique pour la détection des défaillances d'oléoducs. Mémoire Université de Bejaia..
13. Géron, A. (2017). MACHINE LEARNING AVEC SCIKIT-LEARN.
14. Goutte, C., & Gaussier, E. (2005). A probabilistic interpretation of precision, recall and F-score, with implication for evaluation. In Advances in Information Retrieval: 27th European Conference on IR Research, ECIR 2005, Santiago de

Compostela, Spain, March 21-23, 2005. Proceedings 27 (pp. 345-359). Springer Berlin Heidelberg.

15. Josephe, P. T. (2020). E-COMMERCE : an Indian perspective. Sixth Edition, 6.

16. Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008, December). Isolation forest. In 2008 eighth ieee international conference on data mining (pp. 413-422). IEEE.

17. Maddali, A. (mai 2012). Le paiement électronique (expérience québécoise et française). Revue nouvelle économie, N°3.

18. Maïga, A. (2010). Détection et correction automatique des défauts de conception au moyen de l'apprentissage automatique pour l'amélioration de la qualité des systèmes. Rapport technique n° EPM-RT-2010-12.

19. Marthi, B. (2007, June). Automatic shaping and decomposition of reward functions. In Proceedings of the 24th International Conference on Machine learning (pp. 601-608).

20. Nagpal, A., & Gabrani, G. (2019, February). Python for data analytics, scientific and technical applications. In 2019 Amity international conference on artificial intelligence (AICAI) (pp. 140-145). IEEE.

21. Patel, H. (30 août 2021). What is Feature Engineering — Importance, Tools and Techniques for Machine Learning. For towards data science.

22. Powers, D. M. (2020). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. arXiv preprint arXiv:2010.16061.

23. Tsoumakas, G., Katakis, I., & Dufour, O. Vue d'ensemble de la classification multi-étiquette. Journal International du Stockage et de l'Exploitation de Données.

25. Waegeman, W., Verwaeren, J., Slabbinck, B., & De Baets, B. (2011). Supervised learning algorithms for multi-class classification problems with partial class memberships. Fuzzy Sets and Systems, 184(1), 106-125.

26. Watin-Augouard, M. (2014). Le Big data. Les Notes du CREOGN, N° 5. ffhal-03097019ff.

27. Weiss, G. M., & Provost, F. (2003). Learning when training data are costly: The effect of class distribution on tree induction. Journal of artificial intelligence research, 19, 315-354.

28. Wen, P., Xu, Q., Yang, Z., He, Y., & Huang, Q. (2022). Exploring the Algorithm-Dependent Generalization of AUPRC Optimization with List Stability. arXiv preprint arXiv:2209.13262.

Webographie :

11. Egan, J. (2021). Credit card fraud statistics. Bankrate. https://www.bankrate.com/finance/credit-cards/credit-card-fraud-statistics/#fraud

12. Fraude par carte de crédit : comment la prévenir https://www.americanexpress.com consulté le 28/02/2023.

24. Urbano Mateos, S. (consulté février 2023). Fraudes de carte de crédit. Economiafinanza.com.

29. Welcome to Python.org

Liens vers la base de données :

[30] La base de données de Kaggle (Credit Card Fraud) ; https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

Codes source :

```
import sys
import numpy as np
import pandas as pd
import matplotlib
import seaborn as sns
import scipy
import matplotlib.pyplot as plt
from mpl_toolkits import mplot3d
from math import *
from scipy import stats
from math import *
from sklearn.metrics import confusion_matrix

# View libraries versions
print('Python:{}'.format(sys.version))
print('numpy: {}'.format(np.__version__))
print('pandas: {}'.format(pd.__version__))
print('matplotlib: {}'.format(matplotlib.__version__))
print('seaborn: {}'.format(sns.__version__))
print('scipy: {}'.format(scipy.__version__))

# Data reading
data = pd.read_csv('Dataset_credit_card.csv')
print('\n\nData head: \n', data.head())
print('\n\nData columns: \n', data.columns)
print('\n\nData shape: \n', data.shape)
print('\n\nData description: \n', data.describe()

# Data sampling
```

Gaussienne multivariée


```
# Sample the data
#data = data.sample(frac=0.1, random_state=1)
data = data.sample(frac=1, random_state=1)
print('\n\nSampled data shape: ', data.shape)
# Plot histograms of features
data.hist(figsize=(20, 20))
plt.show()

# Determine fraudulent instances ration
Fraud = data[data['Class'] == 1]
Valid = data[data['Class'] == 0]
outlier_fraction = len(Fraud) / len(Valid)
print('Outlier fraction: {}'.format(outlier_fraction))
print('Fraud cases: {}'.format(len(Fraud)))
print('Valid cases: {}'.format(len(Valid)))



# Correlation matrix
Corrmat = data.corr()
fig = plt.figure(figsize=(20, 20))
sns.heatmap(Corrmat, vmax=.8, square=True)
plt.show()

# Get all the columns from the Dataframe
Columns = data.columns.tolist()
Columns = [c for c in Columns if c not in ["Class"]]
target = "Class"
X = data[Columns]
Y = data[target]

# Plot example 1
plt.scatter(X["V1"], X["V2"])
plt.xlabel('V1')
```

```
plt.ylabel('V2')
plt.show()
# Plot example 2 3D)
fig = plt.figure(figsize=(20, 20))
ax = plt.axes(projection="3d")
ax.scatter3D(X["V1"], X["V2"], X["V3"], color="blue")
plt.title("V1, V2, V3")
plt.show()

# Analysis of valid transactions
# Amount
Valid["Amount"].loc[Valid["Amount"] < 600].hist(bins=100)
plt.show()
Fraud["Amount"].hist(bins=100)
plt.show()
pca_columns = list(data)[1:-2]
Valid_pca = Valid[pca_columns]
Fraud_pca = Fraud[pca_columns]
N_test = int(Valid_pca.shape[0] * 0.2 ) # for all dataset
N_outlier = int( Fraud_pca.shape[0] * 0.5 ) # for all dataset
print('N_test: ',N_test)
print('N_outlier: ',N_outlier)
N_test = int(Valid_pca.shape[0] * 0.20)
N_outlier = int(Fraud_pca.shape[0] * 0.50)
shuffled_data = Valid_pca.sample(frac=1)[:].values
X_train = shuffled_data[:-2*N_test]
X_valid = np.concatenate([shuffled_data[-2*N_test:-N_test], Fraud_pca[:N_outlier]])
y_valid = np.concatenate([np.zeros(N_test), np.ones(N_outlier)])
# X_test = np.concatenate([shuffled_data[-N_test:], Fraud_pca[N_outlier:]])
X_test = np.concatenate([shuffled_data[-N_test:], Fraud_pca[N_outlier:2*N_outlier]])
y_test = np.concatenate([np.zeros(N_test), np.ones(N_outlier)])

print(X_train.shape)
```

```
print(X_test.shape)
print(X_valid.shape)


def covariance_mat(X):
    m, n = X.shape
    A = np.zeros((n, n))
    mu = X.mean(axis=0)
    for i in range(m):
        A += np.outer(X[i]-mu, X[i]-mu)
    return A/m
cov_mat = covariance_mat(X_train)
print(cov_mat)
cov_inv = np.linalg.pinv(cov_mat)
cov_det = np.linalg.det(cov_mat)
def multivariate_gauss(x):
    n = len(cov_mat)
    A = (np.exp(-0.5 * np.dot(x, np.dot(cov_inv, x.T))) / (2. * np.pi)**(n/2.) / np.sqrt(cov_det))
    return A
def metrics(X_test, Y_test, epsilon):
    predict = np.array([(multivariate_gauss(x) <= epsilon) for x in X_test], dtype=bool)
    grd_truth = np.array(Y_test, dtype=bool)
    tn, fp, fn, tp = confusion_matrix(Y_test, predict).ravel()
    recall = tp / (tp + fn)
    precision = tp / (tp + fp)
    F1 = 2 * recall * precision / (recall + precision)
    return recall, precision, F1
eps = max([multivariate_gauss(x) for x in Fraud_pca.values])
print(eps)
recall, precision, f1 = metrics(X_test, y_test, eps)


print('For eps = {} :'.format(eps))
print('Recall: ', recall)
print('Precision: ', precision)
```

```
print('F1: ', f1)


# Validation process
validation = []
for e in np.array([0.0000001, 0.0000005, 0.000001, 0.000005, 0.00001, 0.00005, 0.0001, 0.0005,
0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1]) * eps:
    recall, precision, f1 = metrics(X_valid, y_valid, e)
    validation.append([e, recall, precision, f1])
print(validation)
x = np.array(validation)[:,0]
y_recall = np.array(validation)[:,1]
y_precision = np.array(validation)[:,2]
y_f1 = np.array(validation)[:,3]
plt.plot(x, y_recall)
plt.title('Recall')
# plt.xscale('log')
plt.grid(True)
plt.show()


plt.plot(x, y_precision)
plt.title('Precision')
plt.grid(True)
plt.show()
#
print('x=', x)
plt.plot(x, y_f1)
plt.title('F1')
plt.grid(True)
plt.show()
plt.plot(y_recall, y_precision)
plt.title('Recall / Precision graph')
plt.grid(True)
plt.show()
```

```
plt.plot(y_precision,y_recall)
plt.title('Precision / Recall graph')
plt.grid(True)
plt.show()
fig = plt.figure(figsize=(20, 20))
ax = plt.axes(projection="3d")
predictions = np.array([multivariate_gauss(x) <= eps for x in X_test], dtype=int)
ax.scatter3D(X_test[:,1], X_test[:,2], X_test[:,3], c=predictions)
plt.title("V1, V2, V3")
plt.show()
```

Isolation forest

Codes

```
import sys
import numpy as np
import pandas as pd
import matplotlib
import seaborn as sns
import scipy
import matplotlib.pyplot as plt

# import sklearn
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score
from sklearn.ensemble import IsolationForest
from sklearn.neighbors import LocalOutlierFactor
print('Python:{}'.format(sys.version))
print('numpy: {}'.format(np.__version__))
print('pandas: {}'.format(pd.__version__))
print('matplotlib: {}'.format(matplotlib.__version__))
print('seaborn: {}'.format(sns.__version__))
print('scipy: {}'.format(scipy.__version__))

# Data reading
data = pd.read_csv('Dataset_credit_card.csv')
print(data.columns)
print(data.shape)
print(data.describe())

# Sample the data
data = data.sample(frac=0.1, random_state=1)
print(data.shape)

# Plot histograms of features
data.hist(figsize=(20, 20))
```

```
plt.show()

# Fraction of fraudulent instances
Fraud = data[data['Class'] == 1]
Valid = data[data['Class'] == 0]
outlier_fraction = len(Fraud) / len(Valid)
print('Fraud cases: {}'.format(len(Fraud)))
print('Valid cases: {}'.format(len(Valid)))
print('Outlier fraction'.format(outlier_fraction))

# Correlation matrix
Corrmat = data.corr()
fig = plt.figure(figsize=(20, 20))
sns.heatmap(Corrmat, vmax=.8, square=True)
plt.show()

# Get all the columns from the Dataframe
Columns = data.columns.tolist()
Columns = [c for c in Columns if c not in ["Class"]]
target = "Class"
X = data[Columns]
Y = data[target]
print('Shape of X: ', X.shape)
print('Shape of Y: ',Y.shape)

# Define the outlier detection methods
state = 1
classifiers = {
    "Isolation Forest" : IsolationForest(max_samples=len(X),
                                         contamination=outlier_fraction,
                                         random_state=state),
    "Local Outlier Factor" : LocalOutlierFactor(n_neighbors=20,
                                                contamination=outlier_fraction)
```

```
}

# Learn the model
n_outlier = len(Fraud)

for i, (clf_name, clf) in enumerate(classifiers.items()):
    if clf_name == "Local Outlier Factor":
        Y_pred = clf.fit_predict(X)
        score_pred = clf.negative_outlier_factor_
    else:
        clf.fit(X)
        Score_pred = clf.decision_function(X)
        Y_pred = clf.fit_predict(X)

    # Reshape the prediction value 0 for valid, 1 for fraud
    Y_pred[Y_pred == 1] = 0
    Y_pred[Y_pred == -1] = 1
    n_errors = (Y_pred != Y).sum()

    # Run classification metric
    print('{}: {}'.format(clf_name, n_errors))
    print('Accuracy score:', accuracy_score(Y, Y_pred))
    print(classification_report(Y, Y_pred))
import sys
import numpy as np
```

Codes

```
import pandas as pd
import matplotlib
import seaborn as sns
import scipy
import matplotlib.pyplot as plt
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import classification_report
from imblearn.over_sampling import SMOTE
from sklearn.model_selection import train_test_split



print('Python:{}'.format(sys.version))
print('numpy: {}'.format(np.__version__))
print('pandas: {}'.format(pd.__version__))
print('matplotlib: {}'.format(matplotlib.__version__))
print('seaborn: {}'.format(sns.__version__))
print('scipy: {}'.format(scipy.__version__))

# Data reading
data = pd.read_csv(r'C:\Users\mokra\Desktop\Outils\Dataset_credit_card.csv')
print(data.columns)
print(data.shape)
print(data.describe())

# Sample the data
data = data.sample(frac=0.1, random_state=1)
print(data.shape)
# Fraction of fraudulent instances
```

```
Fraud = data[data['Class'] == 1]
Valid = data[data['Class'] == 0]
outlier_fraction = len(Fraud) / len(Valid)

print('Fraud cases: {}'.format(len(Fraud)))
print('Valid cases: {}'.format(len(Valid)))
print('Outlier fraction'.format(outlier_fraction))

# Correlation matrix
Corrmat = data.corr()
fig = plt.figure(figsize=(20, 20))
sns.heatmap(Corrmat, vmax=.8, square=True)
plt.show()

# Get all the columns from the Dataframe
Columns = data.columns.tolist()
Columns = [c for c in Columns if c not in ["Class"]]
target = "Class"
X = data[Columns]
Y = data[target]

print('Shape of X: ', X.shape)
print('Shape of Y: ',Y.shape)

# Analysis of valid transactions
# Amount
Valid["Amount"].loc[Valid["Amount"] < 600].hist(bins=100)
plt.show()
Fraud["Amount"].hist(bins=100)
plt.show()
X_train,X_test, Y_train, Y_test = train_test_split(X,Y,test_size=0.3, shuffle=True, stratify=Y)
```

```
print(X_train.shape)
print(X_test.shape)

# Analysis of valid transactions
# Amount
Valid["Amount"].loc[Valid["Amount"] < 600].hist(bins=100)
plt.show()
Fraud["Amount"].hist(bins=100)
plt.show()
oversampling = SMOTE()
X_train2, Y_train2 = oversampling.fit_resample(X_train,Y_train)
print(X_train2.shape)
print(Y_train2.shape)
sc = StandardScaler()
sc.fit(X_train2)
X_train2_std = sc.transform(X_train2)
X_test_std = sc.transform(X_test)

# SVM Classifier ================================================

print("SVM:")

svc = SVC(C=1.0, random_state=1, kernel='linear')
svc.fit(X_train2_std, Y_train2)
Y_pred = svc.predict(X_test_std)
print(classification_report(Y_test, Y_pred))

# KNN====================================

print("KNN")

knn = KNeighborsClassifier(n_neighbors=7)
knn.fit(X_train2, Y_train2)
```

```
y_pred = knn.predict(X_test)
print(classification_report(Y_test, Y_pred))

# Decision Tree ==============================

print("DT")

DT = DecisionTreeClassifier()
DT = DT.fit(X_train2,Y_train2)
y_pred = DT.predict(X_test)
print(classification_report(Y_test, Y_pred))
```

Résumé

Titre : Détection et classification des transactions frauduleuses par cartes de crédit à base des algorithmes d'apprentissage automatique.

Mots clés : transaction par carte de crédit, apprentissage automatique, gaussienne multivariée, SVM, KNN, LOF, Isolation forest.

Résumé : le commerce électronique a pris une grande ampleur ces dernières décennies, ce qui a entraîné l'émergence de nouvelles formes de fraude dans les transactions par carte de crédit, entrainant de grandes pertes financières pour institutions financières. Pour pouvoir détecter de manière automatique les transactions frauduleuses, plusieurs approches basées sur l'apprentissage automatique sont utilisées. Dans ce travail nous proposons les 3 modèles suivants : la gaussienne multivarié, l'isolation forest et local outlier factor, Ces modèles consistent à l'analyse de base de données, à la détection d'anomalies et au calcul des performances de chaque modèle.

Abstract

Title: Detection and classification of fraudulent credit card transactions based on machine learning algorithms.

keywords: machine learning, multivariate gaussian, SVM, KNN, LOF, Isolation Forest.

Abstract: The rise of e-commerce has lately resulted new types of fraud in credit card transactions, leading to significant financial losses for financial institutions. To resolve this issue, various approaches based on machine learning have been employed to automatically detect fraudulent transactions. In this project, we propose three models: multivariate Gaussian, isolation forest, and local outlier factor. These models involve analyzing databases, identifying outliers, and evaluating the performance of each model.