ROYAUME DU MAROC
Université Mohammed V
- RABAT -
جامعة محمد الخامس
- الرباط -
Faculté des sciences
كلية العلوم

CENTRE D'ETUDES DOCTORALES - SCIENCES ET TECHNOLOGIES

N° d'ordre : 3772

THESE DE DOCTORAT

Structure de Recherche : Intelligent Processing and Security of Systems (I.P.S.S)
Discipline : Informatique
Spécialité : Sécurité Informatique et Machine Learning

Présentée et soutenue le 20/05/2023 par :

IBTISSAM BENCHAJI

Vers un Deep Learning Système de Détection des Fraudes sur cartes bancaires

Devant le jury :

Abderrahim SEKKAKI	PES, Université Hassan II, Faculté des Sciences Ain Chock, Casablanca	Président
Abdelkrim HAQIQ	PES, Université Hassan I, Faculté des Sciences et Techniques, Settat	Rapporteur/Examinateur
Mohamed LAHBY	PH, Université Hassan II, Ecole Normale Supérieure, Casablanca	Rapporteur/Examinateur
Hicham LAANAYA	PH, Université Mohammed V, Faculté des Sciences, Rabat	Rapporteur/Examinateur
Samira DOUZI	PH, Université Mohammed V, Faculté de Médecine et de Pharmacie, Rabat	Co-directrice de thèse
Bouabid EL OUAHIDI	PES, Université Mohammed V, Faculté des Sciences, Rabat	Directeur de thèse

Année Universitaire : 2022/2023

Dédicace

À mon regretté père qu'Allah lui accorde sa Miséricorde, dont la mémoire continue de briller dans mon cœur, et à ma chère mère dont la force et la dévotion ont guidé mon chemin. Votre amour et votre héritage de bienveillance et de générosité sont tissés dans chaque page de cette thèse.

A mon mari qui m'a soutenu tout au long de ce parcours. Ta patience, ton encouragement et ta présence ont été les piliers sur lesquels j'ai construit ce travail. Cette thèse n'est pas seulement la mienne, mais aussi la tienne. Elle symbolise notre dévouement mutuel à la croissance, au partage des connaissances et à la réalisation de nos aspirations.

A mes chers enfants Nada, Hiba et Jad, pour qui je me suis lancée dans cette quête de connaissance, afin de vous montrer qu'aucun rêve n'est hors de portée. Que ces pages vous rappellent l'importance de la curiosité, du travail acharné et de la détermination dans la réalisation de vos propres rêves, et que vous sachiez à quel point vous êtes une source de fierté et d'inspiration pour moi.

À ma sœur et mes frères, vos liens ont été ma source de soutien, d'encouragement et d'inspiration tout au long de ce parcours. Votre présence dans ma vie est un rappel constant de l'importance de la famille et je vous suis infiniment reconnaissante pour tout l'amour et le soutien inconditionnel que vous m'avez offerts. Merci pour tout ce que vous avez fait et pour être toujours à mes côtés.

À tous les membres chers de ma famille et à mes amis exceptionnels, cette thèse est dédiée avec une profonde gratitude. Votre amour inconditionnel, votre soutien constant et vos encouragements ont illuminé chacune des étapes de ce parcours académique. Que ces pages reflètent l'appréciation que j'ai pour chacun de vous, et que notre lien continue de grandir au fil du temps. Merci pour tout ce que vous avez apporté dans ma vie.

Enfin, cette thèse est dédiée à tous les esprits curieux et passionnés qui croient en l'importance de l'exploration intellectuelle. Puissions-nous continuer à explorer, à questionner et à innover, pour que les frontières du savoir continuent de s'étendre et de nourrir notre quête collective de découverte. Avec gratitude envers le passé et espoir pour l'avenir, je dédie ce travail à tous ceux qui partagent cette aspiration.

Remerciements

L'accomplissement de ce travail a été possible grâce à plusieurs personnes qui ont eu une influence positive à travers leurs contributions scientifiques et morales. C'est avec plaisir et reconnaissance que je leur réserve ces lignes.

Je voudrais tout d'abord exprimer ma plus sincère gratitude envers mon directeur de thèse, Monsieur **Bouabid EL OUAHIDI**, Professeur d'Enseignement Supérieur à la Faculté des Sciences de Rabat, pour la confiance qu'il m'a accordée en acceptant d'encadrer ce travail, pour ses précieux conseils et pour toutes les heures qu'il a consacrées à diriger cette recherche. Sans ses encouragements, cette thèse n'aurait pas pu être achevée. Je suis donc ravie d'avoir travaillé en sa compagnie.

J'adresse un remerciement particulier à ma co-directrice de thèse, Madame **Samira DOUZI**, Professeure Habilitée à la Faculté de Médecine et de Pharmacie de Rabat, pour son encadrement et engagement. Son aide, ses conseils et son enthousiasme pour la recherche scientifique m'ont beaucoup apporté tout au long de l'élaboration de ce travail. Cette thèse restera dans ma mémoire comme un moment de stimulation scientifique et de bonne humeur grâce à elle.

Ma profonde gratitude s'adresse au président du jury Monsieur **Abderrahim SEKKAKI**, Professeur d'Enseignement Supérieur à la Faculté des Sciences Ain Chock de Casablanca, pour sa confiance et ses encouragements ainsi que pour la bienveillance dont il a fait preuve à mon égard.

Je remercie Monsieur **Abdelkrim HAQIQ**, Professeur d'Enseignement Supérieur à la Faculté des Sciences et Techniques de Settat, pour l'intérêt qu'il a porté à mes travaux et ses précieux retours ainsi que pour les connaissances complémentaires qu'il m'a apporté lors de nos discussions. Vous m'avez fait honneur en faisant partie de mon jury.

Je remercie Monsieur **Mohamed LAHBY**, Professeur Habilité à l'école Normale Supérieure de Casablanca, pour le temps engagé dans l'évaluation de mon travail, pour ses lectures attentives et ses précieux compléments. Vous m'avez fait honneur en faisant partie de mon jury.

Je remercie aussi Monsieur **Hicham LAANAYA**, Professeur Habilité à la Faculté des Sciences de Rabat, pour son implication et ses nombreux conseils et remarques et pour les compléments éclairés qu'il a pu me fournir. Vous m'avez fait honneur en faisant partie de mon jury.

J'exprime également toute ma gratitude à mes collègues de la structure de recherche **I**ntelligent **P**rocessing and **S**ecurity of **S**ystems **(I.P.S.S)** pour leur amitié et leur soutien, et pour avoir créé un environnement de travail cordial.

Enfin, ma profonde et sincère gratitude à ma famille pour leur amour, leur aide et leur soutien continus et sans égal. Je suis à jamais redevable à tous les membres de ma famille de m'avoir donné les opportunités et les expériences qui ont fait de moi ce que je suis. Ils m'ont encouragé de manière désintéressée à explorer de nouvelles directions dans la vie et à chercher mon propre destin. Ce voyage n'aurait pas été possible sans eux, et je leur dédie cette étape importante.

Résumé

L'objectif de cette thèse est d'apporter des contributions significatives à la recherche scientifique en proposant de nouvelles approches de machine learning visant à améliorer la détection des fraudes sur cartes bancaires. Elle aborde principalement les défis complexes auxquels est confronté un système de détection de fraudes en mettant l'accent sur le problème de déséquilibre des classes, la définition du contexte d'achat frauduleux à partir des données historiques et l'exploitation des informations pertinentes pour la tâche de classification en utilisant les mécanismes d'attention.

Tout d'abord, les données sur les transactions par carte de crédit souffrent d'un fort déséquilibre vu que le nombre des transactions frauduleuses est beaucoup plus réduit que celui des transactions légitimes (moins de 1% des transactions sont frauduleuses). Nous proposons dans cette thèse une nouvelle méthode de ré-échantillonnage qui consiste à générer de nouvelles données, à partir d'une classe minoritaire d'un dataset, en se basant sur la méthode de clustering k-Means et l'algorithme génétique.

Par ailleurs, les attributs décrivant une transaction bancaire ignorent les informations séquentielles qui se sont avérées très pertinentes pour la définition des comportements d'achat et des stratégies de fraudes. Dans cette thèse, nous avons montré que la capturation de l'historique des achats à partir de données séquentielles en utilisant les réseaux de neurones récurrents LSTM, a conduit à une amélioration significative de la prédiction des fraudes sur cartes bancaires. Ensuite, nous avons utilisé les mécanismes d'attention pour améliorer les performances des réseaux de neurones récurrents en se focalisant sur les informations pertinentes à la tâche de classification.

Enfin, nous avons exploré un nouveau modèle de deep learning pour la définition du comportement d'achat frauduleux en se basant sur l'approche PV-DM (Paragraph Vector-Distributed Memory). Les résultats obtenus révèlent que l'utilisation du modèle PV-DM permet d'obtenir de bonnes performances et est considéré plus robuste et plus simple que le modèle LSTM couramment utilisé pour le traitement séquentiel des données.

En conclusion, ces travaux permettent de considérer les connaissances contextuelles dans le cadre de la détection de fraudes par carte de crédit afin d'améliorer la tâche de classification. Les méthodes proposées peuvent être étendues à toute tâche supervisée comportant des datasets déséquilibrés ou séquentiels.

Mots clés : Big Data, Fraud detection, Deep Learning, Imbalanced datasets, Sequence learning, Attention mechanism, PV-DM.

Abstract

The objective of this thesis is to make significant contributions to the scientific research by proposing new machine learning approaches to improve credit card fraud detection. It mainly addresses the complex challenges faced by a fraud detection system by focusing on the class imbalance problem, the definition of the fraudulent purchase context from historical data and the exploitation of relevant information for the classification task using attention mechanisms.

First, the credit card transaction data suffer from a strong imbalance since the number of fraudulent transactions is much smaller than the number of legitimate transactions (less than 1% of transactions are fraudulent). In this thesis, we propose a new resampling method that consists in generating new data, from a minority class of a dataset, based on the k-Means clustering method and the genetic algorithm.

Moreover, the attributes describing a credit card transaction ignore the sequential information that has been shown to be very relevant for the definition of purchasing behaviors and fraud strategies. In this thesis, we showed that capturing purchase history from sequential data using LSTM recurrent neural networks led to a significant improvement in the prediction of bank card fraud. Next, we used attention mechanisms to improve the performance of these recurrent neural networks by focusing on the relevant information for the classification task.

We also explored a new deep learning model for the definition of fraudulent purchase behavior based on the PV-DM (Paragraph Vector-Distributed Memory) approach. The results obtained reveal that the use of the PV-DM model allows to obtain good performances and is considered more robust and simpler than the LSTM model commonly used for sequential data processing.

In conclusion, this work allows to consider contextual knowledge in the context of credit card fraud detection in order to improve the classification task. The proposed methods can be extended to any supervised task involving unbalanced or sequential datasets.

Keywords : Big Data, Fraud detection, Deep Learning, Imbalanced datasets, Sequence learning, Attention mechanism, PV-DM.

Table des figures

Liste des tableaux

Table des matières

Introduction générale

1 Contexte

La fraude est une manœuvre criminelle pratiquée intentionnellement par une personne pour obtenir de manière illégale un profit financier. Elle peut également se produire dans le seul but de tromper une autre personne ou entité, par exemple en fournissant de fausses informations. La fraude n'est pas un phénomène récent propre à la société contemporaine, les fraudeurs pratiquent des activités frauduleuses depuis bien des décennies [Baesens2015] [Akhilomen2013].

En dépit du développement de technologies avancées pour prévenir la fraude, telles que la norme EMV (**E**uropay **M**astercard **V**isa) et la protection 3D-SECURE, un volume non négligeable de transactions par carte de crédit reste illégitime. En effet, selon les statistiques publiées en 2019 par Nilson Report [Report2019], les pertes financières mondiales causées par la fraude à la carte de crédit ont atteint 27,85 milliards de dollars en 2018 et devraient atteindre 35,67 milliards de dollars d'ici cinq ans et 40,63 milliards de dollars dans dix ans. La Figure 1 illustre l'évolution des dommages mondiaux en milliards de dollars causés par la fraude à la carte de crédit. Il est à noter que les pertes pourraient être plus élevées en réalité, car de nombreux cas n'ont pas été signalés aux instances extérieures et ont été résolus en interne pour éviter toute publicité négative [Robinson2016].

FIGURE 1 – *Dommages en milliards de dollars à l'échelle mondial. 2010-2027*

En outre, une enquête mondiale a été menée par KPMG en 2018 et a révélé que 61 % des participants ont indiqué que le volume total des fraudes avait augmenté et 59 % ont affirmé que la valeur a augmenté [Hicks2019]. Il a aussi été constaté que la tendance de la plupart des types de fraude est en hausse. La Figure 2 montre la progression des différents types de fraude dans différentes zones géographiques.

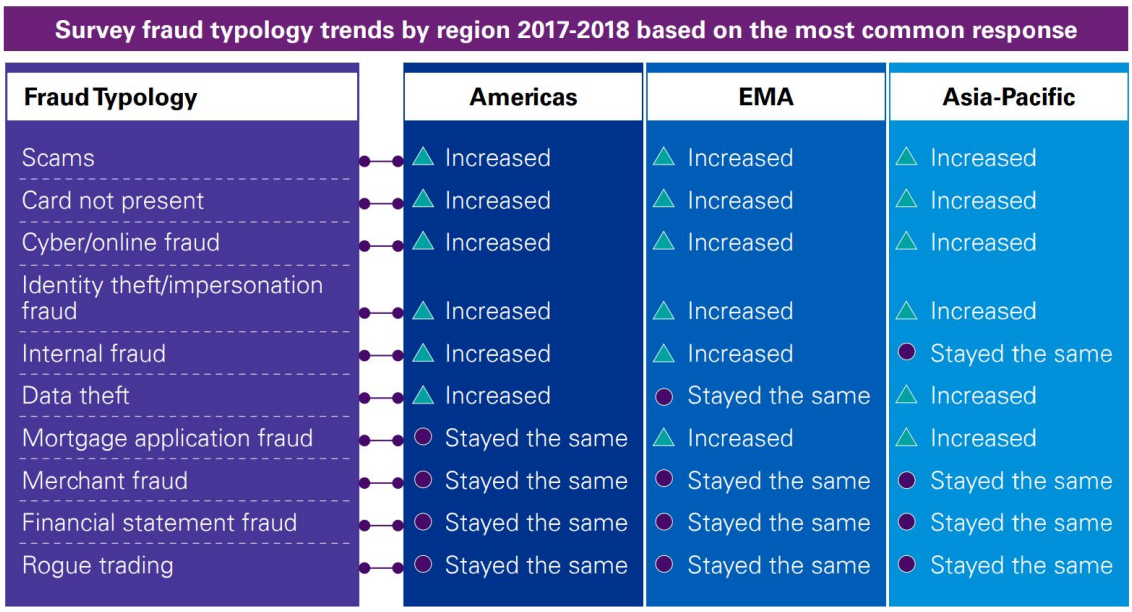

FIGURE 2 – *Tendance mondiale des différents types de fraudes*

Dans ce contexte, la fraude à la carte de crédit est devenue une préoccupation majeure pour les systèmes bancaires modernes. La charge financière que représente cette fraude est un défi de taille pour les institutions financières et les fournisseurs de services, les obligeant à adapter et à améliorer en permanence leurs systèmes de prévention et de détection de fraude. Leurs efforts ne se limitent pas à atténuer les pertes directes subies par les transactions frauduleuses, mais aussi à veiller à ce que les clients légitimes ne soient pas impactés par les contrôles de vérification automatisés ou manuels.

2 Types de fraudes

Dans le secteur des paiements, la fraude par carte de crédit a pour origine soit le vol d'une carte physique, soit la manipulation d'informations sensibles relatives au compte bancaire d'un titulaire légitime de carte, telles que le numéro de la carte de crédit, la date d'expiration ou même le code CVC de vérification de la carte (Card Validation Code). En cas de vol d'une carte physique, le titulaire de la carte est averti à temps et peut rapidement signaler le vol à la banque émettrice. En revanche, la compromission d'informations sensibles peut facilement passer inaperçue pour le titulaire de la carte pendant des semaines, jusqu'à ce que le fraudeur utilise les informations d'identification pour effectuer une transaction frauduleuse. Ce n'est qu'à ce moment-là que le titulaire de la carte, l'émetteur ou le commerçant peut éventuellement repérer l'acte frauduleux et prendre les contre-mesures nécessaires.

Selon les experts en détection des fraudes par carte de crédit [Ghosh1994] [Delamaire2009] [Laleh2009], les stratégies de fraudes sont divisées en cinq types différents :

- **Cartes de crédit perdues/volées (≈1% des transactions frauduleuses)** : Ce type de fraudes se produit lorsqu'une personne perd sa carte de crédit physique ou que celle-ci est volée puis utilisée par un fraudeur comme étant sa propre carte. Étant donné que les fraudeurs disposent d'une carte physique et des numéros CVC, ils peuvent utiliser alors la carte pour acheter illégitimement des biens ou des services au nom du propriétaire légitime de la carte. Le propriétaire n'est pas en mesure d'être informé de la transaction, à moins qu'il ne reçoive le relevé mensuel des charges.

- **Cartes non reçues (<1% des transactions frauduleuses)** : Cartes de crédit interceptées durant la production ou la remise par voie postale. Afin de lutter contre ce type de fraude, les banques exigent une authentification supplémentaire du destinataire avant d'activer la carte (Récupération de la carte à l'agence bancaire, contact par téléphone afin que la carte soit activée, etc.).

- **Vol d'identité (Marginal)** : Ce type de fraude découle d'une usurpation d'identité. Le fraudeur cible une personne qui ne dispose d'aucune carte de crédit et essaie de se procurer des informations telles que sa date de naissance et son numéro de carte d'identité nationale par le biais d'appels ou de faux courriels. Après avoir obtenu ces informations, il adresse une demande à une banque émettrice avec ces fausses informations d'identité. Il peut s'agir d'une identité partiellement ou entièrement artificielle, dénommée fraude à l'identité, ou de l'identité volée d'une autre personne, dite vol d'identité. La fraude à la demande est un cas particulier de délit d'identité. L'élément clé de la fraude à la demande est l'adresse. C'est l'endroit où la carte de crédit sera envoyée et récupérée par le fraudeur. Les contre-mesures déployées par les banques contre ce type de fraude comprennent le rapprochement des données de la demande pour découvrir les éventuels doublons, la tenue et le partage de listes noires avec d'autres banques [Phua2005].

- **Cartes contrefaites (<10% des transactions frauduleuses)** : La carte est dupliquée lors de l'utilisation d'une carte authentique ou lors du piratage d'une base de données et elle est ensuite reproduite sur de fausses cartes par des groupes de criminalité organisée. Les fraudeurs récupèrent et recopient les informations sensibles et les caractéristiques de sécurité d'une carte existante. Ce type de fraude était prédominant dans le passé, mais il a été partiellement résolu par la norme **EMV**. Cette norme tire son nom des organismes fondateurs (**E**uropay **M**astercard **V**isa) et représente le standard international de sécurité des cartes de paiement qui consiste essentiellement à remplacer la carte à piste par une carte à puce plus sécurisée.

- **Fraudes par carte non présente (>90% des transactions frauduleuses)** : La majorité des fraudes par carte de crédit se produisent sur des transactions de commerce électronique (e-commerce). Les informations d'identification (numéro de carte, date d'expiration et CVC) sont généralement obtenues lors d'un piratage de bases de données organisé par des groupes criminels internationaux et sont ensuite vendues sur le Dark Web. British airways, Mariot Hotels et Ticket Master ont par exemple été victimes de violations de données en 2018. La plupart des commerçants (90%) utilisent la technologie 3D SECURE qui protège le titulaire de la carte par une double identification. Un autre problème qui entrave la lutte contre la fraude par carte non présente est que les institutions financières ne signalent pas les attaques qui ont provoqué des violations de données en raison de la mauvaise publicité que cela entraînerait [Robinson2016].

3 Détection de fraudes par cartes

Dans [Van Vlasselaer2015], la fraude est décrite comme un phénomène aux multiples facettes regroupant les caractéristiques suivantes : un crime peu commun, dissimulé, évoluant dans le temps et souvent soigneusement organisé. Ces caractéristiques sont résumées dans [Baesens2015] :

- **Peu commun** : Indépendamment du contexte ou de l'application exacte, seule une minorité des cas concernés sont typiquement des fraudes, et seul un nombre limité d'entre eux sont connus pour être des fraudes. En outre, les fraudeurs masquent les activités frauduleuses par des activités non frauduleuses, ce qui explique le caractère peu commun de la fraude.
- **Dissimulé** : Le comportement des fraudeurs ne se distingue pas des autres. Ils adoptent un comportement normal pour passer inaperçu et rester couverts par les non-fraudeurs. Cela rend la fraude imperceptiblement dissimulée, dans la mesure où les fraudeurs parviennent à se cacher en étudiant et en planifiant soigneusement la manière de commettre la fraude.
- **Évoluant dans le temps** : Les fraudeurs changent continuellement de méthodes car leur objectif est de ne pas être détectés autant que possible. Cela signifie que les techniques et les astuces utilisées par les fraudeurs évoluent dans le temps en même temps que les mécanismes de détection de fraude, ou mieux, en avance sur eux.
- **Soigneusement organisé** : Les fraudeurs organisent les activités frauduleuses avec le plus grand soin, à l'instar d'autres crimes organisés. Très souvent, ils n'opèrent pas de manière isolée mais s'associent avec d'autres criminels. Ces actes frauduleux sont commis moyennant une grande variété de stratégies et d'outils techniques afin de recueillir des informations personnelles sur les titulaires de cartes et sur leurs cartes de crédit : Key-loggers, sniffers, clonage de sites, faux sites marchands, vol physique de cartes ou même la production illégale de cartes artificielles. Ainsi, la fraude n'est pas un événement isolé [Akhilomen2013].

Ces caractéristiques constituent un grand défi pour les industries, en particulier l'industrie financière, pour détecter les activités frauduleuses telles que les transactions frauduleuses effectuées par carte de crédit. Par conséquent, la détection et la prévention de fraude sont les éléments les plus importants d'une stratégie efficace de lutte contre les fraudeurs. La détection de fraude désigne la capacité de découvrir des activités frauduleuses et définir les actions à entreprendre après qu'une fraude ait lieu, tandis que la prévention de fraude désigne les mesures qui peuvent être prises pour empêcher la fraude de se produire ou réduire sa survenance [Vona2017].

Un système type de détection de fraude se compose d'un dispositif automatique et d'un dispositif manuel. Le dispositif automatique est basé sur des règles de détection de fraude. Il analyse toutes les nouvelles transactions entrantes et leur attribue des scores de fraude. Le processus manuel comprend l'expertise et les efforts des investigateurs de fraude. Ils se concentrent sur les transactions présentant des scores de fraude élevés et fournissent un retour d'information sur les transactions analysées pour mettre à jour et améliorer l'outil automatique (Figure 3).

Les systèmes de détection des fraudes se basent soit sur des règles établies par des experts, soit des règles établies à partir des données ou une combinaison des deux. Les règles établies par des experts visent à identifier des scénarios spécifiques de fraude qui ont été découverts précédemment par les investigateurs de fraude. Un exemple de scénario de fraude pourrait être le suivant : *"Un titulaire de carte réalise une transaction dans le pays A et, dans l'heure qui suit, il réalise une deuxième transaction avec exactement le même montant dans le pays B."* Si un tel scénario est détecté dans le flux de transactions, alors le système de détection des fraudes produira une alerte. Par contre, les règles établies à partir des données reposent sur des algorithmes d'apprentissage automatique (machine learning). Ils extraient des modèles de fraude (fraudulent patterns) à partir de données historiques et visent à détecter ces modèles au sein du flux de données des nouvelles transactions entrantes.

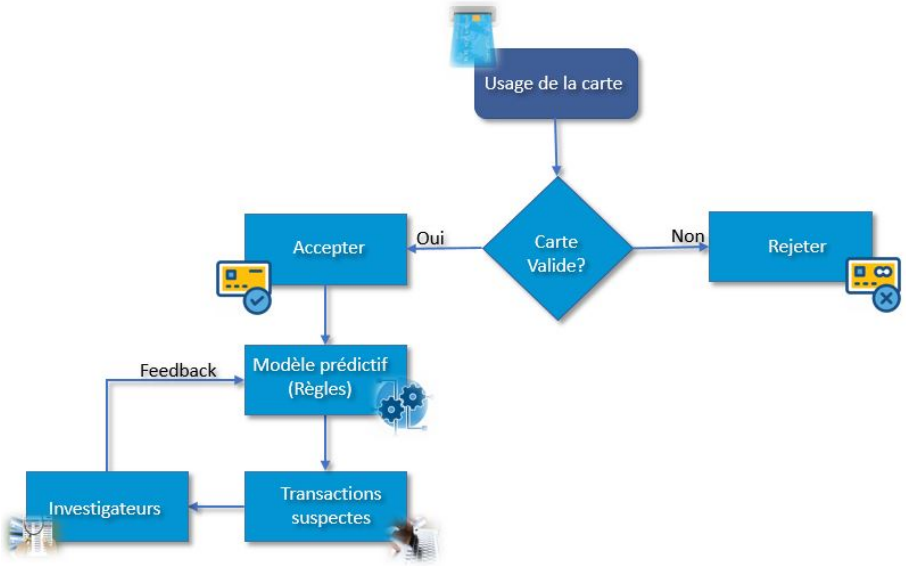

FIGURE 3 – *Processus de détection des fraudes sur cartes bancaires*

3.1 Détection de fraudes par des experts

L'approche fréquemment utilisée pour la détection de fraude est l'approche basée sur les experts. Il s'agit d'une approche classique fondée sur l'expérience, l'intuition et la connaissance de l'activité ou du domaine de l'analyste des fraudes [Baesens2015]. Typiquement, cette approche basée sur l'expertise consiste en une investigation manuelle d'un cas suspect, qui peut avoir été signalé, par exemple, par un client se plaignant d'être facturé pour des transactions qu'il n'a pas effectuées. Une telle transaction contestée peut indiquer qu'un nouveau mécanisme de fraude a été découvert ou développé par les fraudeurs, et nécessite donc une investigation détaillée pour que l'institution financière puisse comprendre et traiter ce nouveau mécanisme [Vona2017].

3.2 Détection de fraudes à partir des données

Au cours des dix dernières années, le monde a connu une croissance sans précédent des données. Plus de 2,5 exaoctets de données sont générés chaque jour et s'accumuleront pour atteindre 175 zettaoctets de données d'ici 2025 [Goughlin2018]. Selon IBM, 90% des données présentes dans le monde aujourd'hui ont été créées au cours des huit dernières années [Jacobson2013]. Ces données sont générées de partout : capteurs, sites de médias sociaux, images et vidéos numériques, transactions d'achat, téléphones portables, signaux GPS, etc. Ces données se caractérisent par : leur masse, leur évolution rapide et leur diversité, mieux connues sous le nom de Volume, Vitesse et Variété respectivement. Ces caractéristiques ont fait naître la notion de Big Data et d'applications axées sur les données, en particulier l'analytique, qui exploite le pouvoir profond des données pour en extraire l'intelligence et prendre les meilleures décisions avec une grande précision.

Même si les approches classiques de détection de fraude basées sur des experts sont encore très répandues et constituent un bon point de départ et un outil complémentaire pour permettre à une organisation de développer un système efficace de détection et de prévention de fraude, on assiste à une évolution vers une analyse de la fraude axée sur les données [Baesens2015]. Aujourd'hui, l'analyse axée sur les données est la solution privilégiée par tous les secteurs à forte utilisation de données, y compris le secteur bancaire. Ce dernier explore les opportunités et les défis des différents types d'analyse, y compris l'analyse des fraudes, en s'appuyant sur des technologies Big Data avancées extrêmement évolutives et performantes.

À la lumière de la discussion ci-dessus, il est évident que l'analyse de la fraude axée sur les données présente plusieurs avantages par rapport à l'approche traditionnelle basée sur les experts pour la détection de fraude. Cependant, différents défis dans les approches guidées par les données doivent être relevés afin de renforcer le système de détection de fraudes.

4 Motivation

Bien que les approches de détection de fraude aient nettement amélioré leur efficacité ces dernières années en recourant à des techniques statistiques performantes et en analysant des quantités massives de données pour découvrir des tendances et des stratégies de fraude, la fraude reste extrêmement difficile à détecter [Baesens2015]. En effet, le problème de la détection de fraude mérite une attention particulière étant donné qu'il est intrinsèquement lié à plusieurs défis complexes qui empêchent une application directe des méthodes de prédiction classiques. Ces défis ne sont pas nécessairement spécifiques à la détection de la fraude par carte de crédit mais leurs effets sont particulièrement visibles pour les institutions financières qui sont confrontées à un volume élevé de transactions effectuées à des fréquences élevées et dont le but est de repérer des événements rares et coûteux dans un vaste domaine d'activités légitimes. Il s'agit donc des facteurs clés qui ont motivé cette initiative de recherche.

Cette section décrit en détail ces défis afin de fournir une vue d'ensemble sur l'importance de cette recherche dans le monde réel et dans le domaine de la science :

- **Classes déséquilibrées** : Avec l'utilisation très répandue des paiements électroniques, les opérateurs financiers doivent faire face à de grandes difficultés pour détecter les comportements frauduleux, rares mais coûteux, dans le vaste flux de transactions générées en continu par des millions de clients répartis dans le monde. En effet, la majorité des transactions entrantes sont légitimes et seule une très faible minorité a été émise par des fraudeurs. Il n'est pas rare que cette fraction ne représente que 0,5% de toutes les transactions [Phua2004]. Pour les algorithmes d'apprentissage, ce déséquilibre entre la classe majoritaire et la classe minoritaire peut entraver la capacité de l'algorithme à apprendre un modèle approprié des données, dans le sens où l'algorithme n'est pas capable de découvrir des modèles (patterns) au sein de la classe minoritaire ou même d'ignorer totalement cette dernière [Abdallah2016].
- **Les modèles dynamiques de fraude** : Représentent un réel défi, en particulier pour les systèmes qui se basent sur des modèles d'apprentissage supervisé. Ces systèmes ne peuvent détecter des modèles de fraude que sur la base d'un dataset d'apprentissage constitué de modèles observés dans le passé. Cependant, les fraudeurs ne cessent de produire de nouvelles méthodes et stratégies frauduleuses pour contourner les systèmes. Une telle variation des activités frauduleuses ne peut donc pas être traitée par les systèmes basés sur l'apprentissage supervisé puisque les datasets d'apprentissage ne contiennent pas les nouveaux modèles de fraude. C'est ce qu'on appelle fréquemment l'apprentissage de nouvelles classes dans l'apprentissage automatique [Muhlbaier2008]. Les systèmes de détection de fraude doivent être capables de mettre à jour leurs modèles régulièrement et efficacement à mesure que de nouvelles observations arrivent.
- **La détection de fraude en temps réel** : Est une solution idéale pour les institutions financières, car les systèmes en temps réel peuvent éviter d'énormes pertes financières. Cependant, la détection des activités frauduleuses en temps réel pose différents défis au système. Les trois principaux défis sont les suivants : (i) La vitesse à laquelle les données circulent aujourd'hui rend difficile leur traitement et leur analyse, (ii) La complexité informatique de l'analyse des fraudes est élevée, et (iii) la conception d'un algorithme performant pour les applications à forte densité de données est une tâche délicate.

- **L'intégration d'un vaste volume de données avec une grande variété** : Représente un énorme défi. L'un des principaux avantages du Big Data est qu'il permet aux utilisateurs de collecter des données provenant d'une grande variété de sources, y compris des sources financières. Ces sources représentent un énorme volume de datasets contenant de nombreux attributs et enregistrements. Par conséquent, les données provenant d'une grande variété de sources posent des problèmes d'intégration au système d'analyse de fraude.
- **La mesure de la performance** : Est un autre défi qui est causé par la distribution déséquilibrée des données. Le taux d'exactitude (observations correctement classées) est celui utilisé pour un problème de classification typique. Cette mesure n'est pas toujours appropriée pour la détection des fraudes [He2009]. Par exemple, si les données contiennent 1% d'observations de fraude, un taux de précision inférieur à 99% est inacceptable. La raison en est simple : le système peut classer tous les enregistrements comme légitimes et donner un taux de précision de 99%. Il faut donc envisager des mesures de performance sensibles aux coûts, qui tiennent compte des observations mal classées, sans déclencher de nombreuses fausses alertes, qui feraient perdre énormément de temps et de ressources à investiguer. Ces statistiques, qu'il est recommandé de prendre en compte dans l'analyse de la détection de fraude, sont notamment la précision, le score F1, la sensibilité, la courbe de Precision-Recall (AUC) et le coefficient de corrélation de Matthew.

Les défis expliqués ci-dessus ont motivé ce projet de recherche, étant donné que le problème se situe au cœur des scénarios du monde réel, et que les défis précités doivent être relevés, ce qui pourrait être très utile pour améliorer la qualité de service des institutions financières et optimiser l'expérience client.

5 Objectif

Dans le domaine de l'analyse des fraudes, l'efficacité constitue un aspect primordial sur lequel il faudrait se focaliser pour obtenir un taux élevé de détection de fraude. L'efficacité garantit la performance des modèles de détection de fraude. Une augmentation de 1% du taux d'exactitude est cruciale car elle aura un impact énorme sur la détection des activités frauduleuses et des fraudeurs.

L'objectif de cette thèse est d'analyser et de développer des méthodes basées sur l'apprentissage automatique qui peuvent être utilisées pour réduire les faux positifs ainsi que les faux négatifs afin de détecter correctement les transactions frauduleuses sans affecter l'expérience client.

6 Contributions

Nous présentons dans cette section, les différentes contributions apportées par cette thèse composée de quatre chapitres, en plus du chapitre introductif, qui décrit le contexte de la fraude financière et notre problème de recherche.

- **Chapitre 1** : Dans ce chapitre, notre objectif est de résoudre le problème du déséquilibre fort des classes auquel est confronté un système d'apprentissage automatique de détection de fraude, en utilisant une nouvelle méthode de génération de la classe minoritaire basée sur la méthode de clustering k-Means et les opérateurs génétiques [Benchaji2019].

  D'abord, la méthode k-Means est utilisée pour répartir, dans des clusters distincts, les observations de la classe minoritaire en fonction de leurs similitudes. L'objectif de cette méthode de regroupement est d'avoir, au niveau de chaque cluster, des observations qui sont les plus similaires possible, ce qui garantira une meilleure représentation des nouvelles observations à générer.

Ensuite, au moyen d'opérateurs génétiques de croisement et de mutation, nous générons au niveau de chaque cluster de nouvelles observations synthétiques appartenant à la classe minoritaire et qui imitent le plus fidèlement possible les observations initiales, puis nous les fusionnons avec le dataset initial pour obtenir un jeu d'apprentissage augmenté. La figure 4 illustre le schéma général de notre méthode.

Nous avons mené une expérience de validation comparative au cours de laquelle nous avons démontré l'efficacité de notre méthode au moyen des mesures de performance les plus appropriées pour la détection de fraude sur cartes bancaires.

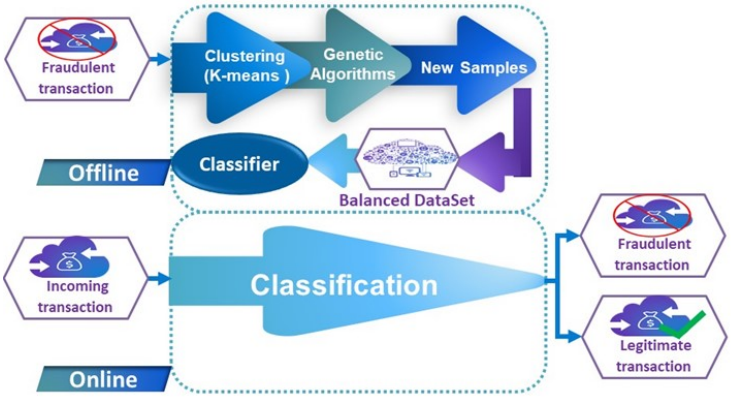

FIGURE 4 – *Méthode de génération des nouvelles observations à partir de la classe minoritaire*

- **Chapitre 2** : Dans ce chapitre, nous proposons un système de détection de fraude basé sur la modélisation séquentielle des données, en utilisant les réseaux de neurones récurrents. En effet, des comportements d'achat similaires peuvent à la fois représenter un comportement tout à fait légitime dans le contexte de certains consommateurs ou des anomalies évidentes dans le contexte d'autres consommateurs. En effet, une fraude n'est pas uniquement une caractéristique de la transaction elle-même, mais une caractéristique à la fois de la transaction et du contexte particulier dans lequel elle s'est produite.

  Pour construire un tel contexte qui résume l'historique des comportements d'achat, nous utilisons les réseaux à mémoire court-terme et long terme LSTM (Long Short Term Memory) comme classifieur dynamique, afin de capturer la dépendance séquentielle entre des transactions consécutives [Benchaji2021a]. L'objectif est de permettre à un classifieur de mieux détecter des transactions très dissemblables dans les achats d'un consommateur. Les résultats obtenus suggèrent que la modélisation basée sur des réseaux de neurones récurrents est une stratégie prometteuse pour caractériser des séquences de transactions et améliorer l'efficacité de la détection de fraude.
- **Chapitre 3** : L'objectif de ce chapitre est d'améliorer les résultats obtenus dans le chapitre précédent en utilisant les mécanismes d'attention [Benchaji2021b], capables de se focaliser sur les informations les plus pertinentes pour la tâche de classification. Le modèle proposé, comparé aux études précédentes, tient compte de la nature séquentielle des données transactionnelles et permet au classifieur d'identifier les transactions les plus importantes dans la séquence d'entrée et qui prédisent avec une plus grande précision les transactions frauduleuses (Figure 5).

  Précisément, la robustesse de ce modèle est construite en combinant la force de trois sous-méthodes : l'approximation et la projection de collecteurs uniformes UMAP (Uniform Manifold Approximation and Projection) pour sélectionner les caractéristiques prédictives les plus utiles, les réseaux LSTM pour incorporer les séquences de transactions et les mécanismes d'attention visant à améliorer la classification du modèle. Les performances de ce modèle présentent de bons résultats en termes d'efficience et d'efficacité.

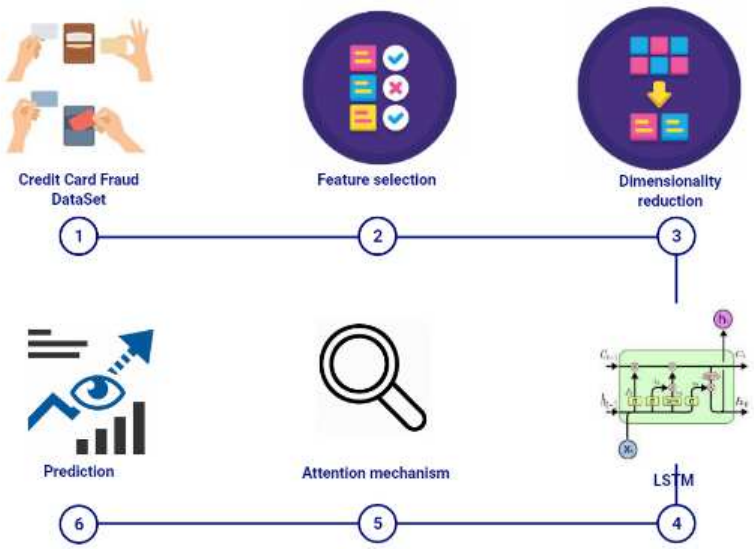

FIGURE 5 – *Utilisation des mécanismes d'Attention pour l'amélioration du modèle proposé*

- **Chapitre 4** : Dans les travaux précédents, nous avons modélisé les comportements d'achat frauduleux en utilisant les réseaux de neurones récurrents. Les expériences ont montré que la caractérisation d'une transaction entrante, en tenant compte de l'historique des transactions précédentes, améliore les performances de prédiction de façon significative, mais nécessite beaucoup de calculs pour fonctionner en raison de leur structure complexe avec des mémoires à court et à long terme pour stocker l'information à différents moments dans le temps, ce qui peut rendre leur utilisation coûteuse en termes de puissance de calcul.

  Dans ce dernier chapitre, nous proposons d'explorer un nouveau modèle de deep learning pour la définition du comportement d'achat frauduleux en se basant sur l'approche PV-DM (Paragraph Vector-Distributed Memory). L'objectif est la génération d'une représentation vectorielle à partir des transactions et des séquences, où les indices contiennent implicitement des informations sur le contexte global de la séquence et les contextes locaux des transactions, ensuite ces vecteurs sont utilisés pour entraîner un modèle de détection de fraudes. Les valeurs expérimentales obtenues révèlent que l'utilisation du modèle PV-DM permet d'obtenir de bonnes performances et est considéré plus robuste et plus simple que le modèle LSTM couramment utilisé pour le traitement séquentiel des données.

Enfin, nous fournissons une conclusion pour le travail effectué dans cette thèse, et donnons des perspectives pour des recherches et travaux futurs.

Le lecteur est averti que nous avons adopté une rédaction de ce document où chaque chapitre contient les concepts de base, la problématique traitée, les approches et modèles proposés et les implémentations réalisées. Ainsi chaque chapitre se suffit à lui même et ce dans un but de faciliter la lecture du document.

1

Modèle de classification efficace des fraudes sur cartes à partir des classes déséquilibrées

Le système de détection des fraudes bancaires est confronté à un défi majeur : les datasets des fraudes bancaires sont fortement déséquilibrés vu que le nombre des transactions frauduleuses est beaucoup plus réduit que celui des transactions légitimes. Ainsi, la plupart des classifieurs traditionnels échouent souvent à détecter les objets de la classe minoritaire pour ces datasets déséquilibrés. Dans ce chapitre, nous proposons une nouvelle méthode de ré-échantillonnage qui consiste à générer de nouvelles données, à partir d'une classe minoritaire d'un dataset, en se basant sur la méthode de clustering k-Means (après avoir determiné le k adéquat par la méthode Elbow) et l'algorithme génétique. Notre approche consiste à utiliser l'algorithme k-Means pour répartir dans des clusters distincts les données de la classe minoritaire, et sur chaque cluster, nous utilisons les opérateurs génétiques (crossover et mutation) pour générer de nouvelles observations, et construire ainsi un dataset augmenté et équilibré. Dans nos implémentations, nous appliquons l'approche proposée à un dataset fortement déséquilibré afin de démontrer son efficacité au moyen des mesures de performance les mieux appropriées.

1 Introduction

L'amélioration de la technologie et l'avènement de nouvelles solutions de paiement ont apporté de nombreux avantages à la société et ont fait de la carte de crédit le mode de paiement le plus populaire pour les achats en ligne et hors ligne, mais elles ont aussi malheureusement entraîné une augmentation des activités frauduleuses. Ces activités illégales, qui visent à obtenir des biens sans payer ou à retirer des fonds illégitimes d'un compte, ont causé de graves dommages aux utilisateurs et aux prestataires de services.

La détection automatique des fraudes par carte de crédit, à l'aide des algorithmes de classification, est un défi à relever dans le domaine de machine learning. Pour cela, différentes techniques ont été proposées dans la littérature pour le résoudre [Aleskerov1997] [Whitrow2009] [Sánchez2009] [Bhattacharyya2011] [Sahin2013] [Dal Pozzolo2014a] [Bahnsen2015]. Toutefois, plusieurs caractéristiques rendent cette tâche difficile et doivent être considérées en même temps lors de la conception d'un système de détection de fraudes bancaires. Tout d'abord, les comportements d'achat et les stratégies de fraude peuvent changer au fil du temps, rendant une fonction de décision apprise par un classifieur non pertinente si celui-ci n'est pas mis à jour. Deuxièmement, le système doit répondre dans des délais très brefs pour être utile lors de scénarios réels. Troisièmement, les données sur les transactions par carte de crédit souffrent d'un fort déséquilibre (class imbalance) en ce qui concerne les effectifs des classes (moins de 1% des transactions sont frauduleuses), comme le montre [Dal Pozzolo2014a] [Krivko2010].

Le problème de dataset déséquilibré se produit lorsque l'une des classes dans les données a beaucoup plus d'observations que l'autre classe. Ce problème est plus visible lorsque l'on considère dans le contexte de données massives (Big Data). Les données qui sont utilisées pour construire les modèles contiennent une très petite partie de la classe minoritaire (Instances positives) par rapport à la classe majoritaire (Instances négatives) [Chawla2004]. Dans les applications du monde réel, il est crucial de classer correctement la classe minoritaire, car cette classe est généralement celle qui présente le plus d'intérêt, comme dans les cas de fraude. Dans cet exemple, la fraude est la classe minoritaire et il est plus important de détecter un cas de fraude en raison de ses conséquences dangereuses qu'une situation normale.

Cette disproportion de classes dans les données rend très difficile à l'algorithme de machine learning de généraliser le comportement de la classe minoritaire et d'apprendre ses caractéristiques et modèles (Patterns). Par conséquent, les performances de ces algorithmes seront biaisées vers la classe majoritaire en raison de leurs nombreux exemples dans le dataset [Japkowicz2002]. A cet effet, des approches spéciales d'exploration de données sont utilisées, soit par les algorithmes de classification traditionnels, soit au niveau du prétraitement des données, afin de résoudre ce problème [Chawla2004].

Dans ce chapitre, nous abordons principalement l'enjeu relatif au dataset déséquilibré et proposons une méthode d'oversampling qui consiste à augmenter la taille de la classe minoritaire en utilisant la méthode de clustering k-Means et les opérateurs génétiques. Ainsi, de nouvelles observations mimées sont générées au niveau de chaque cluster, et sont ensuite fusionnées avec les données d'apprentissage initiales jusqu'à avoir un dataset équilibré, ce qui permettra d'améliorer l'efficacité du classifieur lors de la détection. Nous avons mené une expérience de validation comparative au cours de laquelle nous avons prouvé l'efficacité de notre méthode en termes de taux de classification correcte et de détection de fraude.

Ce chapitre est organisé comme suit. La section 2 présente le défi des datasets déséquilibrés auquel sont confrontés les systèmes de détection. Dans la section 3, les concepts de base, utilisés dans cette étude, sont présentés. Nous abordons, dans la section 4, l'organisation de notre méthodologie proposée. La section 5 décrit le dataset utilisé et discute les résultats obtenus. Enfin, la section 6 conclut le document et propose des idées pour de futures recherches.

2 Le problème du déséquilibre des classes

Comme introduit dans la première section, le défi majeur à relever lors de la conception d'un système de machine learning de détection de fraude est le déséquilibre des classes. Un dataset est dit déséquilibré lorsque le nombre d'instances négatives (majoritaires) est supérieur au nombre d'instances positives (minoritaires). La figure 1.1 illustre la distribution extrêmement déséquilibrée du dataset des transactions sur cartes utilisé dans nos implémentations.

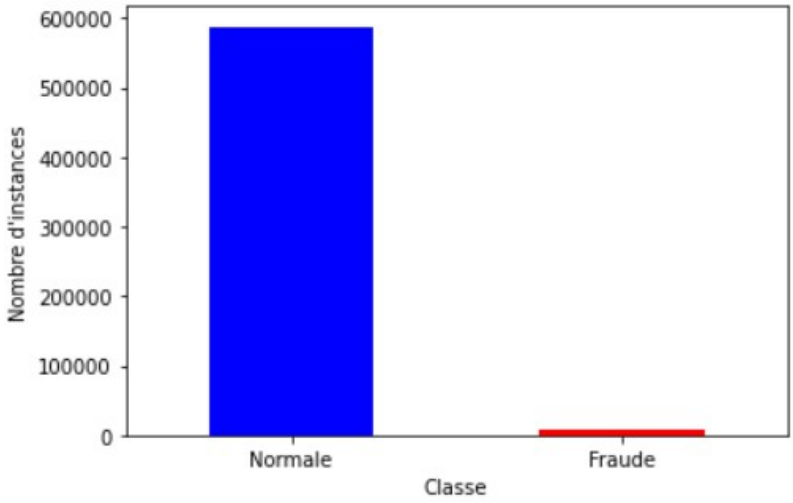

FIGURE 1.1 – *Distribution extrêmement déséquilibrée des classes*

Par conséquent, les algorithmes d'apprentissage automatique qui cherchent à maximiser l'exactitude globale ont souvent tendance à classer toutes les observations comme des instances de la classe majoritaire et obtiennent des résultats médiocres en termes de précision prédictive sur la classe minoritaire (Low recall) [Batista2000]. Cela se produit parce que les méthodes d'apprentissage traditionnelles ne sont pas en mesure d'effectuer une classification correcte des nouvelles observations dans de tels contextes, en fait elles enregistrent une bonne précision uniquement pour les observations qui appartiennent à la classe majoritaire, en reportant des valeurs de précision inacceptables pour les autres cas de la classe minoritaire. En d'autres termes, cela signifie qu'il est possible que, en présence d'un déséquilibre des données, un classifieur prédit toutes les nouvelles observations comme appartenant à la classe majoritaire, en ignorant la classe minoritaire.

Plusieurs approches ont été proposées dans la littérature pour faire face au problème des datasets déséquilibrés. Ces approches peuvent être classées en deux grandes catégories d'approches comme illustré au niveau du schéma 1.2 :

- **Approche de ré-échantillonnage (Data level)** : Cette catégorie de méthodes résout le problème des classes déséquilibrées au niveau des données, c'est-à-dire que des techniques de rééchantillonnage sont appliquées pour modifier directement le dataset initial qui n'est pas équilibré soit en supprimant des données de la classe majoritaire (undersampling) ou en répliquant des données d'apprentissage de la classe minoritaire (oversampling) ou en combinant les deux [Dal Pozzolo2015] [Drummond2003].
- **Approche d'apprentissage d'Ensemble (Algorithmic level)** : Cette catégorie de méthodes résout le problème des classes déséquilibrées au niveau de l'algorithme de classification. En effet, les méthodes d'Ensemble visent à combiner plusieurs classifieurs faibles en un seul classifieur fort, ce qui permet d'améliorer les performances de la classification des datasets déséquilibrés [Cieslak2012] [Bhowan2012] [Wasikowski2009]. Les méthodes d'Ensemble communément utilisées sont les stratégies de bagging [Breiman1996a] et de boosting [Friedman2002] qui seront abordées en détail dans les sections suivantes.

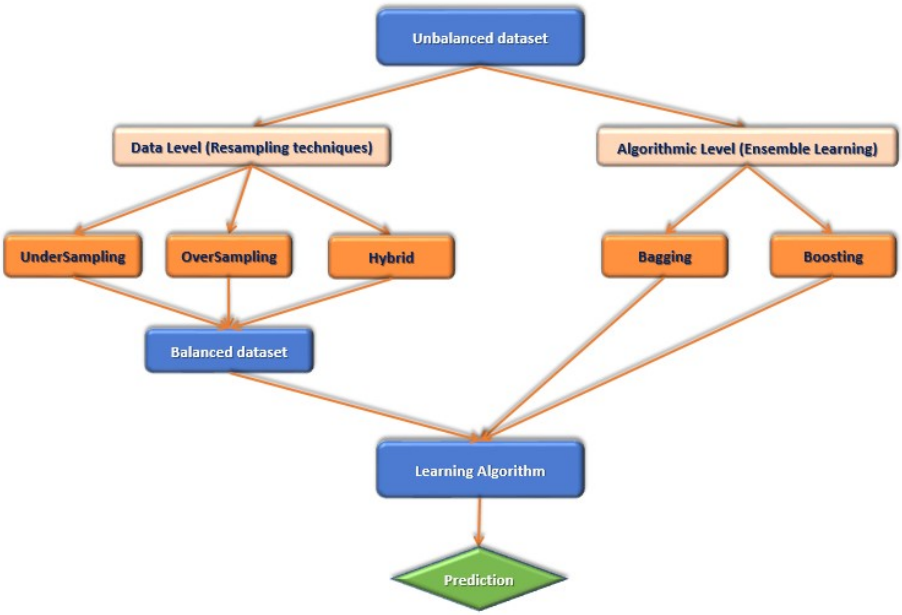

FIGURE 1.2 – *Approches utilisées pour faire face au déséquilibre des datasets*

2.1 Approche de ré-échantillonnage (Data level)

En présence d'une distribution déséquilibrée des classes, certaines tâches de pré-traitement des données doivent être effectuées avant de fournir les données en entrée au modèle prédictif. Ce pré-traitement des données est effectué en utilisant une approche au niveau des données, appelée approche de ré-échantillonnage. On distingue essentiellement trois approches de rééchantillonnage : (a) le sous-échantillonnage (Undersampling), (b) le sur-échantillonnage (Oversampling) et (c) l'approche hybride qui combine les deux [Whitrow2009] [Drummond2003].

2.1.1 Méthodes de sous-échantillonnage (Undersampling)

Les méthodes de sous-échantillonnage [Drummond2003] consistent à supprimer des observations appartenant à la classe majoritaire afin de rendre le dataset équilibré, comme illustré dans la figure 1.3. Ces méthodes sont utilisées lorsque la taille du dataset est importante et que la réduction des observations majoritaires peut améliorer considérablement le temps d'exécution et réduire les problèmes de stockage. Différentes méthodes existent pour choisir les données à supprimer. Parmi les méthodes les plus utilisées, on cite : Random Undersampling [Japkowicz2002] et Tomek Links [Tomek1976].

FIGURE 1.3 – *Méthode d'undersampling*

- **Random Undersampling** :
  « Random Undersampling » [Japkowicz2002] [Estabrooks2004] consiste à éliminer de manière aléatoire un nombre d'observations de la classe majoritaire afin de rendre le dataset équilibré, et ce en fonction du taux souhaité de la classe majoritaire et de la classe minoritaire. Dans un problème non équilibré, il est souvent réaliste de supposer que de nombreuses observations de la classe majoritaire sont redondantes et qu'en enlevant certaines d'entre elles au hasard, la distribution des données ne changera pas de manière significative [More2016]. Cependant, le risque de supprimer des observations pertinentes du dataset est toujours présent, puisque la suppression est effectuée de manière non contrôlée. Par conséquent, les performances du classifieur risque de se détériorer. La figure 1.4 montre un exemple d'undersampling lorsque 50% de la classe majoritaire est supprimée.

FIGURE 1.4 – *Random Undersampling (50%)*

- **Tomek Links** :
  La méthode de "Tomek Links" est plus complexe que le random undersampling. Elle a été développée par Tomek [Tomek1976] et repose sur un calcul de distance. Elle consiste à rechercher des paires de points de chaque classe en bordure de zone séparant la classe majoritaire et la classe minoritaire. Cette paire de points est appelée "Tomek links" [Batista2004]. Une fois l'ensemble de ces paires trouvées, les observations de la classe majoritaire appartenant à une paire "Tomek Links" sont supprimées pour rééquilibrer les données. Cette méthode permet de créer de l'espace et de faciliter la séparation entre les deux classes. Cependant, en raison de la rareté de la classe minoritaire dans le dataset, il est difficile de supprimer un grand nombre d'observations de la classe majoritaire en utilisant la méthode Tomek links. La figure 1.5 illustre cette méthode d'undersampling où les paires de Tomek sont entourées en orange.

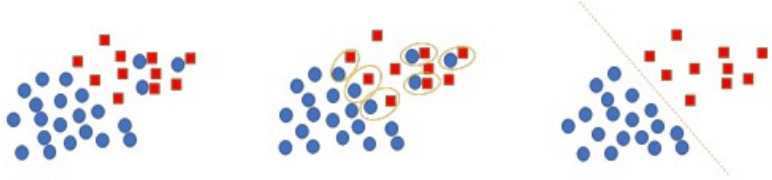

FIGURE 1.5 – *Tomek links*

2.1.2 Méthodes de sur-échantillonnage (Oversampling)

Les méthodes d'undersampling cherchent à équilibrer les deux classes d'un dataset en supprimant des observations de la classe majoritaire afin d'améliorer la connaissance de la classe minoritaire. Le principal inconvénient de ces méthodes est que, lorsque le déséquilibre est trop important, il est nécessaire de supprimer un grand nombre d'observations de la classe majoritaire, ce qui entraîne une détérioration des performances de l'algorithme d'apprentissage en raison de la perte d'information sur la classe majoritaire.

Une autre manière d'équilibrer les deux classes en évitant de perdre un trop grand nombre d'informations sur la classe majoritaire est d'augmenter le nombre d'observations dans la classe minoritaire (Figure 1.6). Comme pour les méthodes d'undersampling, on procède en choisissant aléatoirement les observations que l'on souhaite dupliquer. Des stratégies d'oversampling contrôlées ont également été développées pour dupliquer ou créer des instances dans les parties appropriées de l'espace des données. Quelques exemples de méthodes d'oversampling seront abordés dans les sous-sections suivantes.

FIGURE 1.6 – *Méthode d'Oversampling*

- **Random Oversampling** :

  La méthode « Random Oversampling » [Drummond2003] consiste à augmenter le nombre d'observations de la classe minoritaire en les dupliquant de manière aléatoire. L'information de la classe minoritaire est donc augmentée sans perte d'information sur la classe majoritaire. Cependant, cette duplication n'apporte aucune information et peut entraîner un surapprentissage (Overfitting) des algorithmes. En outre, le random oversampling augmente le temps d'apprentissage étant donné que le dataset équilibré devient artificiellement plus grand.

- **Synthetic Minority Oversampling Technique (SMOTE)** :

  Parmi les techniques de suréchantillonnage les plus utilisées, on retrouve le célèbre algorithme SMOTE (Synthetic Minority Oversampling Technique) [Chawla2002]. Cet algorithme, présenté dans Algorithme 1, utilise les k plus proches voisins (k-NN) [Altman1992] appartenant à la classe minoritaire pour en créer de nouvelles. L'un des plus proches voisins, de l'observation considérée, est sélectionné aléatoirement et une nouvelle observation synthétique est alors créée le long du segment de ligne joignant l'observation considérée et le plus proche voisin sélectionné, comme le montre la figure 1.7.

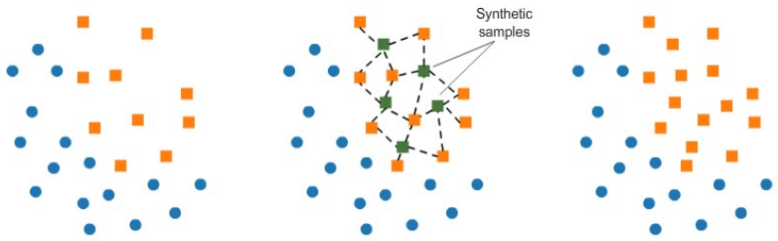

FIGURE 1.7 – *Génération d'exemples synthétiques à l'aide de SMOTE*

Entrée: $S_{\min}$ la classe minoritaire, κ le nombre de voisins à considérer

Début: $S_{new} = S_{\min}$

for *x in* $S_{\min}$ **do**

Calculer les κ voisins de x par : V = κ-NN(x) soit V = $\{v_i\}_{i=1}^{k}$

Choisir un voisin aléatoire $\hat{x} \in V$

Calculer le vecteur distance D par : D = distance $(\{\hat{x}, v_i\})$, i = 1 à κ

Multiplier le vecteur de distance D par un nombre aléatoire $\delta \in [0, 1]$

$x_{new} = x + \delta.D$

$S_{new} = S_{new} + x_{new}$

end

retourner S_{new}

Algorithm 1: L'algorithme SMOTE

Le principal avantage de SMOTE par rapport aux autres méthodes traditionnelles est la création d'observations synthétiques au lieu de réutiliser des observations existantes, ce qui permet d'obtenir un classifieur qui risque moins d'être sur-ajusté (Overfit). Cependant, les observations synthétiques sont créées entre deux points de la classe minoritaire sans considérer les points de la classe majoritaire, ce qui peut conduire à un chevauchement (Overlap) entre les classes minoritaire et majoritaire, et donc augmenter les erreurs de classification. ADASYN [He2008] et Borderline-SMOTE [Han2005], deux dérivations de SMOTE, abordent ce problème en ne prenant en compte que les observations voisines de la classe minoritaire. Pour une liste exhaustive des algorithmes basés sur SMOTE, le lecteur intéressé est invité à consulter une revue réalisée par [Fernández2018].

2.1.3 Méthode hybride

Une méthode hybride combine les approches de sur-échantillonnage et celles de sous-échantillonnage en éliminant des données dans la classe majoritaire et ajoutant des données dans la classe minoritaire afin de rééquilibrer la distribution des classes [Batista2003]. Ci-après un exemple de méthode hybride SMOTE-TL combinant les approches SMOTE et Tomek Links.

- **Combinaison de SMOTE et Tomek Links** :

  SMOTE est une approche très efficace visant à balancer les distributions de classes. Cependant, lors de la génération de nouvelles observations minoritaires, le groupe de points minoritaires peut envahir l'espace de la classe majoritaire. Fournir de telles données au modèle peut conduire à un sur-apprentissage (Overfitting). Par conséquent, pour atténuer une telle situation, les approches SMOTE et Tomek Links sont combinées pour équilibrer la distribution des classes. Dans ce processus, le dataset d'apprentissage initial est d'abord sur-échantillonné à l'aide de SMOTE, puis la suppression des liens de Tomek est appliquée au dataset résultant pour produire un dataset équilibré [Santoso2017].

2.2 Approche d'apprentissage d'Ensemble (Algorithmic level)

Bien souvent, dans la problématique des datasets déséquilibrés tels que le cas des fraudes, l'utilisation d'un bon modèle prédictif et de ré-échantillonnage ne suffisent pas à obtenir des résultats satisfaisants. En effet, le sous-échantillonnage aura tendance à diminuer l'apprentissage de la classe majoritaire et le sur-échantillonnage peut avoir des effets de sur-apprentissage (Overfitting) de la classe minoritaire. Pour pallier à ces problèmes, il est fait recours à l'approche algorithmique appelée méthode d'Ensemble ou « Agrégation de classifieurs » décrite dans cette section.

La méthode d'Ensemble consiste à améliorer les performances d'apprentissage en combinant un grand nombre de classifieurs de base. Ces algorithmes sont basés sur des stratégies adaptatives (Boosting [Friedman2002]) ou aléatoires (Bagging [Breiman1996a]). De nombreux articles comparatifs montrent leur efficacité sur des exemples de problèmes réels [Ghattas2000] [Valentini2002]. Le succès des méthodes d'Ensemble tient du fait qu'elles garantissent une erreur plus faible que le meilleur des classifieurs qu'ils agrègent. Les sous-sections suivantes détaillent ces deux stratégies d'Ensembles.

2.2.1 Bagging

Bagging, abréviation de Bootstrap Aggregation, est une méthode d'agrégation de classifieurs faibles pour obtenir un classifieur performant, proposée par L. Breiman [Breiman1996a]. Le principe du bagging est d'entraîner séparément chaque classifieur de base C_k sur un échantillon bootstrap D_k (k indique le nombre d'échantillons bootstrap) du dataset d'apprentissage. Un échantillon bootstrap est obtenu par tirage aléatoire avec remise de n' instances du dataset initial D de taille n. Cette technique d'échantillonnage garantit que chaque bootstrap est indépendant de ses pairs, puisqu'il ne dépend pas des observations choisies précédemment lors de l'échantillonnage. La figure 1.8 illustre l'approche Bagging.

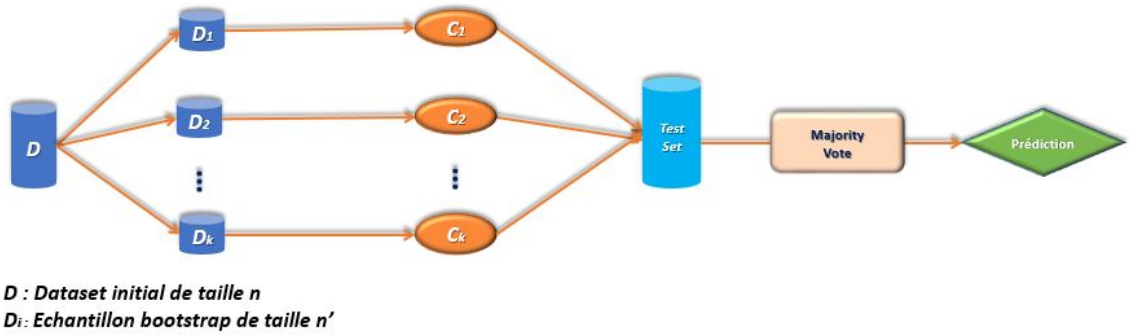

FIGURE 1.8 – *Aperçu de la méthode Bagging*

Ainsi, pour chaque tirage D_i, une classification C_i est obtenue. La classification finale est basée sur un vote majoritaire des classifications obtenues. Son avantage est qu'elle améliore la performance des classifieurs instables en calculant la moyenne de leurs réponses. Si les classifications C_i calculées pour chaque tirage D_i ont une variance importante, alors la classification finale aura une variance réduite. Un classifieur est dit instable si un petit changement dans les données d'apprentissage provoque un gros changement dans le comportement du classifieur. Le but du bagging est d'atténuer l'instabilité inhérente à certains classifieurs.

L'algorithme Bagging est décrit dans Algorithme 2.

Entrée: κ le nombre d'échantillons bootstrap

Sortie: Un classifieur agrégé, C^*

for *i=1 jusqu'à k* **do**

Créer un échantillon bootstrap D_i par tirage aléatoire de $n^{'}$ instances du dataset D

Entraîner un classifieur de base C_i sur D_i

end

Pour appliquer le classifieur agrégé C^* sur le dataset de Test, pour une instance x et sa vraie étiquette de classe y :

$$C^*(x) = argmax \sum_i^k \delta(C_i(x) = y)$$

$$\text{Avec } \delta(.) = \begin{cases} 1, & \text{si } C_i(x) = y \\ 0, & \text{sinon.} \end{cases}$$

Algorithm 2: Algorithme de Bagging

2.2.2 Boosting

Le Boosting est une autre méthode d'Ensemble très puissante introduite par Schapire [Schapire1990]. Elle consiste à combiner des classifieurs faibles, également appelés classifieurs de base, pour créer un classifieur plus fort, susceptible de produire de meilleurs résultats que ceux générés par un classifieur individuel. Contrairement au bagging, dans lequel chaque classifieur est exécuté en parallèle et les résultats sont combinés à la fin, le boosting consiste à former les classifieurs faibles de manière séquentielle, de sorte que chaque classifieur tente de corriger son prédécesseur en ajoutant plus de poids aux données qui ont été mal classées précédemment. Par conséquent, le prochain classifieur faible concentrera donc ses efforts sur les données les plus difficiles à prédire.

Les algorithmes de boosting existants diffèrent par leur méthodologie de pondération des erreurs pour l'obtention du classfieur suivant ou encore les techniques d'agrégation en elles-mêmes. Il existe de nombreux exemples d'algorithmes de Boosting tels que AdaBoost, GradientBoost, XGboost, etc. L'algorithme originel AdaBoost, pour Adaptative Boosting [Freund1999], est l'un des plus utilisés, souvent avec un algorithme d'arbre de décision comme classifieur de base. La figure 1.9 présente l'évolution simplifiée d'une classification avec la méthode boosting.

L'augmentation des poids des données mal classées est indiquée par l'augmentation de la taille de ces données. La classification finale est obtenue en agrégeant les trois classifieurs faibles :

1. Le premier classifieur classe correctement deux (+) et l'ensemble des (-). Les poids des mal classés vont alors augmenter pour le second classifieur.
2. Le second classifieur tient compte des nouveaux poids. Il classe correctement l'ensemble des (+), mais trois (-) sont mal classés. Les poids sont alors mis à jour pour le dernier classifieur.

3. Le troisième classifieur, en tenant compte des nouveaux poids, construit une nouvelle classification.
4. Au final, ils sont regroupés dans un classifieur final qui classe correctement toutes les données. Ce graphique 1.9 illustre bien le passage de classifieurs faibles à un bon classifieur.

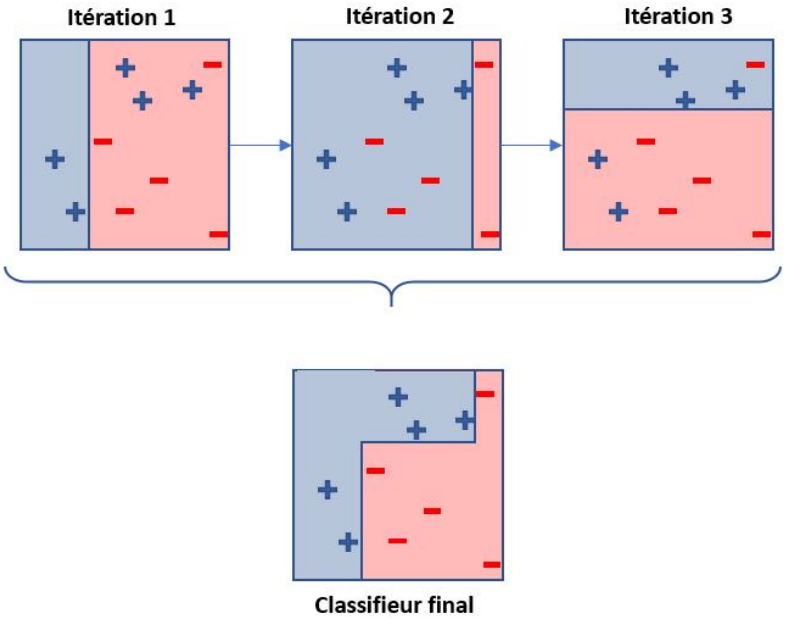

FIGURE 1.9 – *Exemple d'un boosting à 3 itérations*

En dépit de leurs avantages qui permettent d'augmenter considérablement la stabilité des modèles et l'amélioration des performances, les méthodes d'Ensemble existantes présentent plusieurs limites [Zhang2008] :

- Les résultats expérimentaux, obtenus après application des méthodes de Bagging et de Boosting, montrent que ces techniques peuvent réduire l'erreur de généralisation, et que le Boosting a un meilleur effet que le Bagging. Mais dans certains cas, le boosting peut causer un Overfitting [Quinlan2014] [Bauer1999].
- Pour les méthodes de Boosting, chaque classifieur de base est entraîné sur des données qui sont pondérées en fonction de la performance du classifieur précédent. Le classifieur de base suivant se concentre sur les échantillons actuels qui sont classifiés difficilement.
- Le Boosting permet à la fois de réduire le biais et la variance [Domingos2000], alors que le Bagging ne peut que réduire la variance. En effet, le Bagging ne contribue pas à la réduction du biais de manière ciblée, ce qui signifie que toute réduction du biais est uniquement le résultat du hasard [Breiman1996b].
- Le Boosting est sensible au bruit et aux valeurs extrêmes (Outliers) [Bauer1999]. Cette sensibilité au bruit est généralement attribuée à la fonction de perte exponentielle (Loss function) qui spécifie que si une observation n'est pas classée comme identique à son étiquette indiquée, le poids de l'observation augmentera de manière drastique. Par conséquent, lorsqu'une observation de la phase d'apprentissage est associée à une étiquette erronée, les méthodes de Boosting essaient tout de même de faire correspondre la prédiction à l'étiquette indiquée, ce qui entraîne une diminution des performances [Zhou2012].

3 Concepts de base

Cette section décrit les techniques de prétraitement qui ont été utilisé dans le cadre de ce travail, à savoir la méthode de clustering k-Means et les algorithmes génétiques.

3.1 Méthode de clustering k-Means

Le clustering est une méthode d'apprentissage non supervisé (unsupervised learning) utilisée pour le partitionnement d'un jeu de données non annotéées en un certain nombre de groupes. L'objectif de cette technique est de catégoriser les données d'un dataset de telle sorte que les données similaires soient regroupées dans un cluster et que les données dissemblables soient placées dans des clusters différents. La méthode de clustering la plus populaire est k-Means en raison de sa simplicité et de son efficacité [MacQueen1967].

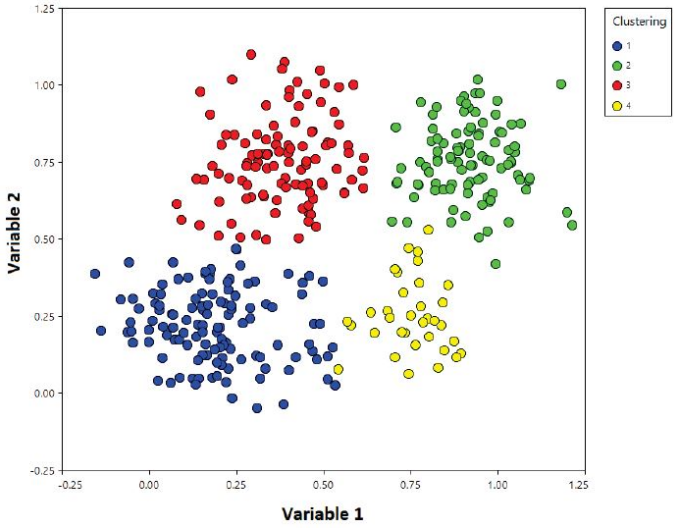

FIGURE 1.10 – *Exemple de résultats de clustering*

La méthode k-Means regroupe n objets en k groupes tout en conservant une forte similarité intra-groupe et une faible similarité inter-groupe des objets. L'algorithme k-Means positionne une donnée dans un groupe en cherchant le centre du cluster le plus proche à l'aide d'une mesure de distance. Cette mesure calcule la distance de similarité entre chaque donnée et les centroïdes. Un exemple de clustering en R^2 est fourni dans la figure 1.10, où quatre clusters sont formés à partir de deux variables désignées par Variable 1 et Variable 2. La figure montre comment les observations de données sont regroupées et distinguées entre 4 catégories. Parmi les nombreuses mesures de distance disponibles, la distance euclidienne est la plus utilisée pour calculer la distance entre les objets et les centroïdes.

k-Means est une méthode utilisée dans divers domaines afin de regrouper différents types de données. Parmi les applications dans lesquelles k-Means a été appliquée, on peut citer :

- La quantification des couleurs, où les pixels d'une image sont regroupés en clusters [Celebi2011] [Kanungo2002] [Juang2010].
- La segmentation des marchés [Kuo2002], en segments significatifs, comme la segmentation des habitudes des acquéreurs en fonction des groupes d'âge.
- L'analyse des données d'expression génétique [Tavazoie1999] [Yeung2003].
- Le groupement de documents [Haraty2015] [Steinbach2000], où les documents similaires sont regroupés dans un groupe tandis que les autres documents sont affectés à d'autres groupes.

L'algorithme k-Means est décrit dans Algorithme 3 :

Entrée: Soit un ensemble X de n points de données $X = \{x_1, x_2, x_3, \ldots, x_n\}$ dans un espace à d dimensions R^d et un nombre entier k.

1. Sélectionner aléatoirement l'emplacement de de k centroïdes $c_j([j = 1, \ldots, k])$.
2. Calculer la distance euclidienne entre toutes les données et chacun des centres des clusters.
3. Pour chaque donnée x_i, trouver le cluster C_j dont le centre c_j est le plus proche de x_i et affecter x_i à ce cluster, de telle sorte à minimiser la fonction d'erreur suivante :

$$\sum_{j=1}^{k} \sum_{x_i \in C_j} (\|x_i - c_j\|)^2$$

Où c_j est le centre du cluster C_j, $x_i \in C_j$ se réfère à tous les éléments appartenant au cluster C_j, n_j est le nombre d'éléments du cluster C_j et ||*|| représente la distance euclidienne.

4. Recalculer chacun des centres des clusters en tenant compte de la moyenne de toutes les données qui appartiennent à chaque cluster.

$$c_j = \frac{1}{n_j} \sum_{i=1}^{n_j} x_i$$

5. Recalculer la distance entre les nouveaux centres des clusters et chaque observation des instances frauduleuses.
6. Réaffecter l'appartenance de chaque donnée à un cluster selon le centre recalculé.
7. Si aucune donnée n'a été réaffectée ou si le nombre maximal d'itérations est atteint, alors arrêter. Sinon, revenir à l'étape 4.

Algorithm 3: Algorithme k-Means

3.2 Algorithmes génétiques

3.2.1 Déscription :

Les algorithmes génétiques (AGs) s'inspirent de la théorie de l'évolution développée par Charles Darwin (1859) sur la survie et la reproduction des espèces. Leur origine est attribuée à John Holland, qui a publié en 1975 un ouvrage exposant les racines scientifiques de cette technique [Holland1992]. Un AG est un algorithme de recherche métaheuristique utilisé depuis plus de 40 ans pour résoudre des problèmes complexes. Les AGs, qui imitent l'évolution naturelle des espèces, reposent sur les mécanismes de la sélection naturelle et de la génétique naturelle [Goldberg1994]. Par conséquent, les opérations et les concepts utilisés dans ces algorithmes sont basés principalement sur le vocabulaire de la génétique.

Ainsi, un ensemble d'individus est appelé population. Un individu est caractérisé par ses chromosomes et chaque chromosome est constitué d'un ensemble d'éléments appelés caractéristiques ou gènes. Dans un AG, la population évolue grâce à trois phases principales qui représentent la structure clé de l'algorithme : La sélection qui favorise le choix des bons individus de la population à évoluer (appelés parents). Le croisement (crossover) qui consiste en la génération, pour chaque couple sélectionné, d'enfants qui vont hériter des gènes (caractéristiques) de chacun de leurs parents. La mutation qui altère une partie du chromosome de l'individu (enfant) pour diversifier la population et permettre l'exploration d'autres parties de l'espace de recherche. Cette nouvelle population est par la suite évaluée et la procédure est répétée jusqu'à ce que le critère de terminaison soit atteint.

Dans un usage plus large du terme, un algorithme génétique est un modèle basé sur une population qui utilise des opérateurs de sélection et de recombinaison pour générer de nouvels individus dans un espace de recherche. La figure 1.11 présente une vue schématique des différentes étapes du processus.

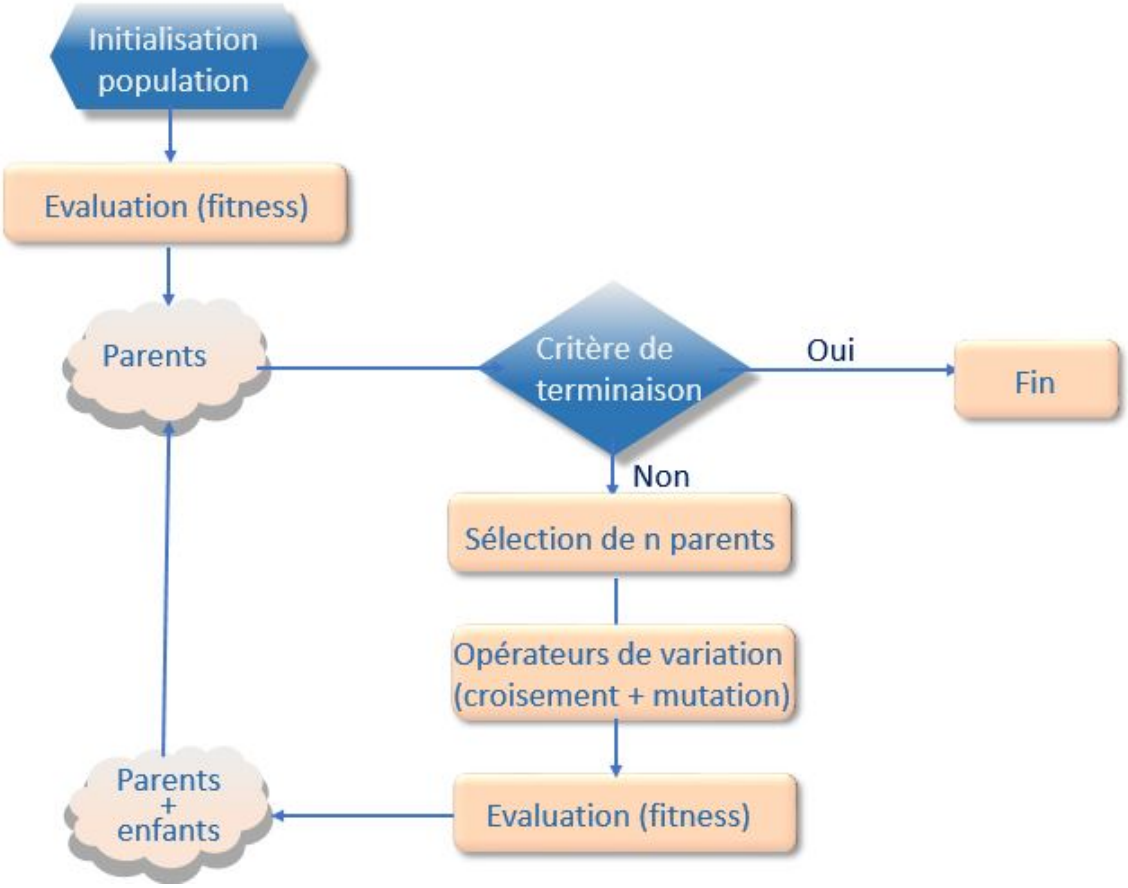

FIGURE 1.11 – *Organigramme d'un algorithme génétique*

Ainsi, un algorithme de programmation génétique se déroule comme suit :

1. Générer une population initiale formée d'un ensemble fini d'individus, dite génération initiale ;
2. Définir une fonction d'évaluation dite fitness permettant d'évaluer un individu et de le comparer aux autres ;
3. Choisir les individus par un mécanisme de sélection pour un éventuel couplage ;
4. Générer de nouvels individus à l'aide des opérateurs génétiques en utilisant :
    - L'opérateur de croisement : Manipule la structure des chromosomes des parents afin de produire des individus meilleurs ou différents. Cet opérateur est effectué selon une probabilité P_c .
    - L'opérateur de mutation : Sert à éviter d'établir des populations uniformes incapables d'évoluer. Il consiste à modifier les valeurs des gènes des chromosomes selon une probabilité de mutation P_m .
5. Etablir un compromis entre les individus produits (enfants) et les individus producteurs (les parents) afin de décider ceux éligibles à garder et ceux qui doivent disparaitre.

3.2.2 Opérateurs génétiques :

Comme décrit dans la section précédente, le processus évolutif des AGs repose sur trois opérateurs génétiques, à savoir la sélection, le croisement et la mutation.

Cette section fournit une brève description de ces opérateurs et de leurs différentes variantes.

- **Sélection :**
  La sélection permet d'identifier, dans une population, les meilleurs individus (appelés parents) susceptibles d'être croisés. Deux procédures de sélection fréquemment utilisées vont être présentées dans la sous-section suivante, à savoir : La sélection par tournoi et celle par roulette.

  Sélection proportionnelle (par roulette) : Il s'agit de représenter sur une roulette chacun des individus de la population par une section qui est proportionnelle à leur valeur de santé (fitness). Une fonction de fitness définit une mesure de la qualité d'un chromosome. L'AG utilisera cette mesure comme critère pour déterminer les chromosomes à faire évoluer. Ensuite, on lance P fois la roulette et on sélectionne chacun des gagnants. En principe, on lance la boule dans la roulette, et on choisit l'individu dans le secteur duquel la boule a fini sa course. Cette méthode de sélection favorise les meilleurs individus, mais les mauvais ont tout de même des chances d'être sélectionnés. Par contre, le coût d'exécution et la variance sont élevés. La perte de diversité est aussi possible car le nombre de copies obtenues des meilleurs individus (voir uniquement du meilleur) peut représenter l'ensemble de la prochaine population.

  Sélection par tournoi :

  La sélection par tournoi n'utilise aussi que des comparaisons entre les individus, et ne nécessite même pas de tri de la population. Elle possède un paramètre T, qui représente la taille du tournoi. Pour sélectionner un individu, on en tire T uniformément dans la population, et on sélectionne d'une manière déterministe le meilleur de ces T individus. Au cours d'une génération il y aura autant de tournois que d'individus à sélectionner. Cette technique est caractérisée par une pression de sélection en général plus forte que les méthodes proportionnelles (pour qu'un individu peu performant puisse être sélectionné, il faut que ses adversaires soient encore moins bon que lui). De plus, elle est la moins chère en termes de coût d'exécution ; elle est peu sensible aux erreurs, facilement paramétrable par la valeur de T, et ne conduit pas à une convergence prématurée.

- **Croisement :** Le croisement est l'opérateur qui simule le mécanisme d'accouplement dans le processus naturel de l'évolution d'une espèce. Pour appliquer le croisement, deux chromosomes de la population sont sélectionnés de manière aléatoire, ou suivant une certaine stratégie de sélection.

  Ces deux chromosomes sont considérés comme des chromosomes parents. L'accouplement consiste à croiser les deux chromosomes parents en mixant leurs gènes. Ce croisement produit deux chromosomes enfants qui se partagent les informations génétiques des parents. Le croisement permet donc de produire deux nouvels individus proches des individus parents.

  Il existe plusieurs stratégies pour définir l'opérateur de croisement [Hüe1997]. La stratégie la plus utilisée consiste à choisir aléatoirement un point de croisement (croisement 1X) où les deux parents sont coupés en deux parties, de part et d'autre du point de croisement, et les chromosomes enfants sont produits en sélectionnant de manière alternative une partie d'un des deux parents. Le croisement 2X consiste à choisir aléatoirement deux positions et à échanger les valeurs des deux chromosomes entre ces deux positions.

La figure 1.12 illustre ces deux exemples de croisement.

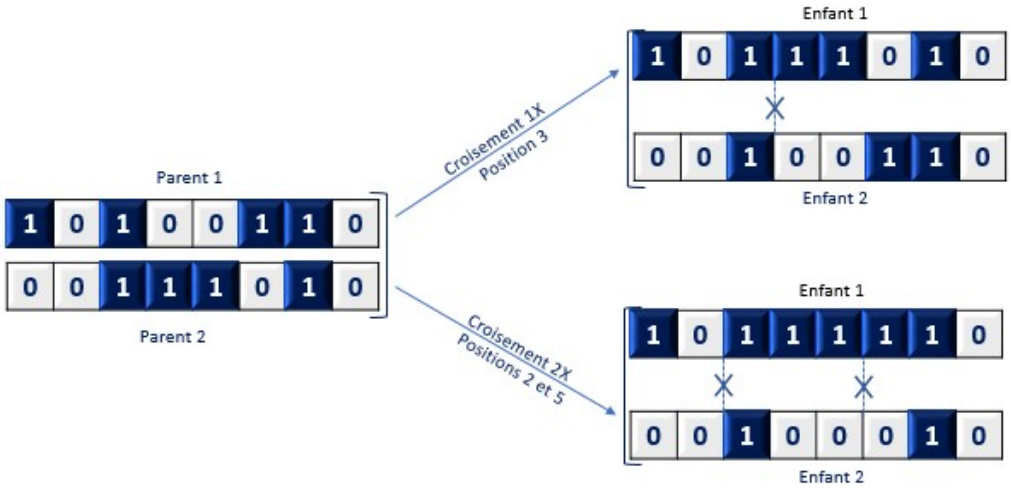

FIGURE 1.12 – *Opération de croisement 1X (en haut) et de croisement 2X (en bas)*

Il est à noter que d'autres stratégies plus avancées de croisement peuvent être utilisées telles que le croisement uniforme (Uniform Crossover) ou le croisement semi-uniforme (Half-Uniform Crossover) [Eshelman1991].

- **Mutation :** L'opérateur de mutation apporte aux algorithmes génétiques l'aléa nécessaire à une exploration efficace de l'ensemble de l'espace de recherche. Une mutation est donc l'altération d'un ou plusieurs gènes d'un individu choisi, ce qui nous garantit que l'AG serait susceptible d'atteindre la plupart des points du domaine réalisable et éviter la convergence rapide vers des optimums locaux.

  Cet opérateur de mutation est utilisé avec une probabilité (P_m) nommée probabilité de mutation. Dans les algorithmes génétiques à codage binaire, cette probabilité s'effectuerait sur le gène en échangeant sa valeur de 0 à 1 ou de 1 à 0, ce qui est connu sous le nom de mutation par retournement de bit (Bit-Flip), comme illustré au niveau de la figure 1.13. Il est à noter que la plupart des AGs appliquent une mutation avec une faible probabilité (généralement entre 0,001 et 0,01).

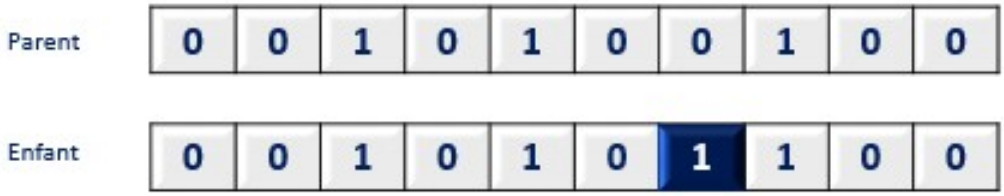

FIGURE 1.13 – *Exemple de mutation sur un gène.*

4 Approche proposée

Après avoir mené l'étude détaillée ci-dessus pour examiner le problème des datasets déséquilibrés et les différentes approches qui ont été proposées dans la littérature pour y remédier, nous avons constaté que nous ne pouvons toujours pas atteindre une sensibilité satisfaisante (c'est à dire, une classification correcte de la classe minoritaire) sans une baisse significative de l'exactitude (Accuracy). Pour cela, nous avons développé dans ce travail une nouvelle méthode hybride d'oversampling basée sur les machines learning pour résoudre le problème du déséquilibre des classes. Cette section décrit le dataset et l'approche proposée dans ce travail, ensuite sont détaillés les paramètres d'implémentation du modèle, les résultats expérimentaux et la discussion.

4.1 Déscription du dataset

Les datasets fournissent un outil d'apprentissage et de validation de la performance des modèles proposés et contribuent de manière décisive à l'avancement de la recherche scientifique. Une des contraintes de la détection de fraude par cartes bancaires est le fait que les données sont hautement confidentielles et ne sont pas accessibles au public [Dal Pozzolo2014a] [Lopez-Rojas2016]. Par conséquent, les chercheurs suggèrent d'utiliser des données synthétiques qui sont modélisées à partir d'un dataset réel de façon à ce que ces données comprennent des caractéristiques semblables. Pour ce travail, nous utilisons des données produites par BankSim, un outil de simulation spécialement conçu pour imiter les données de fraude [Vaughan2020]. Les données produites par BankSim sont disponibles en accès libre sur le site de Kaggle.

BankSim utilise une approche de simulation à agents multiples basée sur un échantillonnage de données de transactions réelles fournies par une banque espagnole. Les données bancaires réelles consistent en des milliers de transactions enregistrées entre novembre 2012 et avril 2013. BankSim imite ces données bancaires originales en utilisant plusieurs agents de trois catégories différentes : les commerçants, les clients et les fraudeurs. Ces agents interagissent pendant une séquence de jours simulés, résultant en un registre de transactions d'achat qui ressemblent étroitement aux données bancaires originales.

TABLEAU 1.1 – *Déscription des caractéristiques des transactions bancaires*

Nom	**Déscription**
Step	Le jour où la transaction a eu lieu de 1 à 180
Customer ID	Un numéro identifiant le compte client concerné par la transaction
Age Category	Une valeur catégorique classant le client dans un des 8 différents groupes d'âge.
Gender	Une variable catégorique indiquant le sexe du client
Zip Code of account	Le code postal associé au client
Merchant ID	Le numéro identifiant le commerçant impliqué dans la transaction
Zip Code of Merchant	Le code postal du commerçant
Category purchase	Une variable catégorielle indiquant le type de bien ou de service acheté
Amount of purchase	Le montant total de la transaction
Fraud status	Une variable binaire indiquant si la transaction est frauduleuse ou non

Le dataset utilisé dans le cadre de ce travail se compose de transactions correspondant à des achats par carte effectués pendant 180 jours simulés et compte 594 643 transactions différentes, parmi lesquelles 7 200 (≈1,2%) sont labelisées en tant que "fraudes", tandis que les 587 443 restantes sont labellisées en tant que "légitimes". Les données brutes contiennent des informations sur les transactions qui sont traitées. Le tableau 1.1 décrit les caractéristiques d'une transaction donnée.

4.2 Mise en oeuvre de l'approche

Dans ce travail, nous proposons une nouvelle approche d'oversampling en se basant sur la méthode de clustering k-Means et l'algorithme génétique pour résoudre la problématique des classes déséquilibrées.

- Tout d'abord, nous appliquons la méthode k-Means pour répartir, dans des clusters distincts, les observations de la classe minoritaire (Observations frauduleuses), en fonction de leurs similitudes. L'objectif de cette méthode de regroupement est d'avoir, au niveau de chaque cluster, des observations qui sont les plus similaires possible, ce qui garantira une meilleure couverture et représentation des nouvelles observations à générer par la suite.

- Ensuite, au moyen d'opérateurs génétiques de croisement et de mutation, nous générons au niveau de chaque cluster de nouvelles observations synthétiques appartenant à la classe minoritaire et qui imitent le plus fidèlement possible les observations initiales, puis nous les fusionnons avec le dataset initial pour obtenir un jeu d'apprentissage augmenté. La figure 1.14 illustre le schéma général de la méthode proposée.

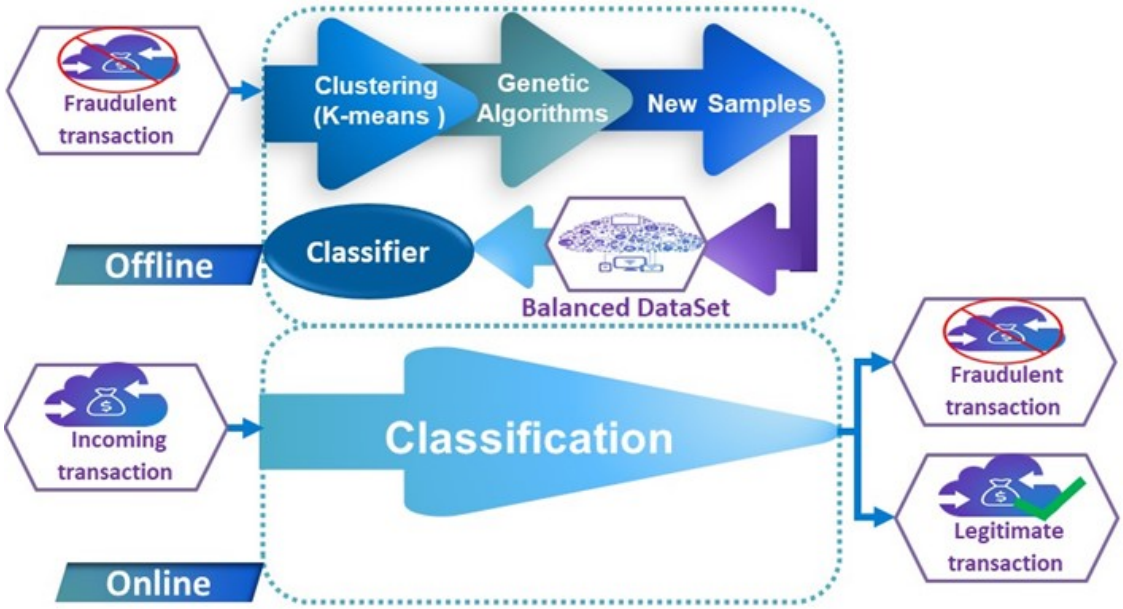

FIGURE 1.14 – *Architecture de notre approche*

4.2.1 Première étape : Le choix du nombre k de clusters

Etant donné que la méthode k-Means requiert en entrée de définir le nombre de clusters k, on utilise la méthode dite 'Elbow' pour déterminer le nombre optimal de clusters à utiliser.

Le principe de la méthode 'Elbow' est de lancer la méthode de clustering k-Means sur le dataset d'apprentissage pour différentes valeurs de k et de calculer la variance des différents clusters. La variance est la somme des distances entre chaque centroid d'un cluster et les différentes observations incluses dans le même cluster. Ainsi, on cherche à trouver un nombre de clusters de telle sorte que les clusters retenus minimisent la distance entre leurs centres (centroids) et les observations dans le même cluster. On parle de minimisation de la distance intra-classe.

En appliquant la méthode Elbow sur notre dataset, nous avons construit le graphique 1.15 qui illustre les différents nombres de clusters k en fonction de la variance. On remarque sur ce graphique, la forme d'un bras où le point le plus haut représente l'épaule et le point où k vaut 10 représente l'autre extrémité : La main. Le nombre optimal de clusters est le point représentant le coude.

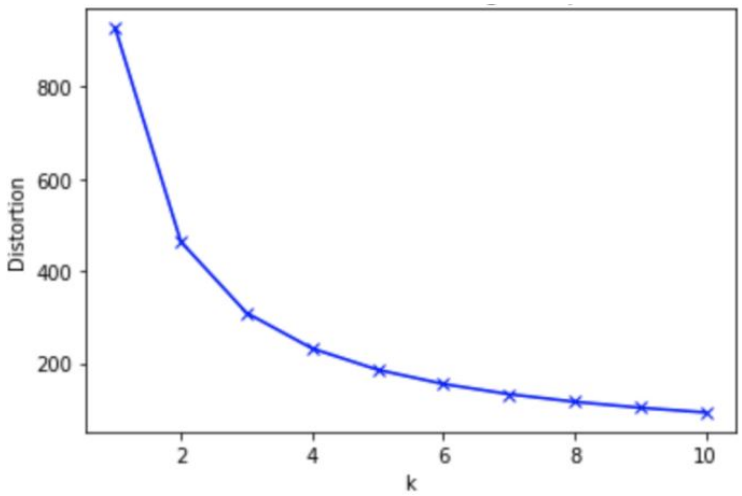

FIGURE 1.15 – *La méthode Elbow pour la définition du nombre k des clusters*

Ce point du coude correspond au nombre de clusters à partir duquel la variance ne se réduit plus significativement. En effet, la "chute" de la courbe de variance (distortion) entre 1 et 4 clusters est significativement plus grande que celle entre 6 clusters et 10 clusters. Le fait de chercher le point représentant le coude, a donné nom à cette méthode : La méthode **Elbow** (coude en anglais). On déduit que le coude est représenté par k=4 qui est le nombre optimal de clusters que nous fournirons en entrée de la méthode k-Means pour notre implémentation.

4.2.2 Deuxième étape : La méthode k-Means

Une fois le nombre k de clusters est determiné, la méthode k-Means est appliquée à notre dataset, afin de répartir la classe minoritaire des fraudes en quatre clusters distincts en se basant sur leurs similitudes. De nouvelles observations mimées peuvent être créées, grâce à des opérateurs génétiques au niveau de chaque cluster. Ainsi, chaque cluster peut obtenir une certaine proportion de nouvelles observations, ce qui garantit que les nouvelles observations du dataset de la classe minoritaire ont une meilleure couverture et représentation. La figure 1.16 présente les 4 custers formés après application de la méthode k-Means sur notre dataset.

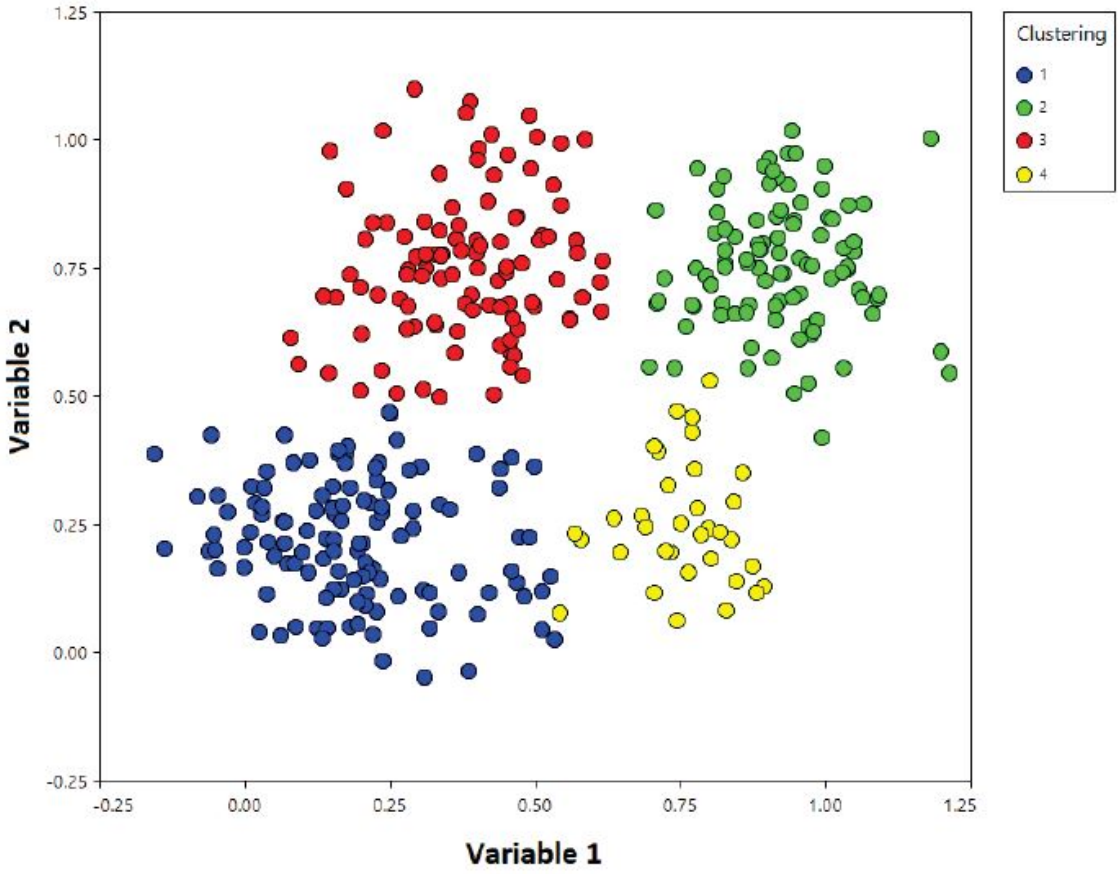

FIGURE 1.16 – *Répartition des données frauduleuses en utilisant la méthode k-Means*

4.2.3 Troisième étape : Les opérateurs génétiques

Par la suite, l'algorithme génétique considère les observations présentes au niveau de chaque cluster en tant que parents de la population actuelle et de nouvels individus synthétiques sont générés à partir de ces observations en utilisant les opérateurs génétiques de croisement et de mutation. Ces nouvels individus hériteront des caractéristiques de leurs parents, plutôt qu'une simple duplication. Les individus les moins adaptés de cette génération sont éliminés et les plus adaptés sont retenus comme parents pour la génération suivante. Cette procédure est renouvelée jusqu'à ce qu'un nombre de générations N_{gen} soit satisfait, et le dataset d'apprentissage augmenté est équilibré.

Dans ce travail, le dataset utilisé présente un déséquilibre de classe important, avec seulement 1,2% des transactions étiquetées comme frauduleuses (7200 fraudes), tandis que 98,8% des transactions sont étiquetées comme non-frauduleuses (587 443 transactions légitimes). Cela signifie que le ratio de déséquilibre de notre dataset est d'environ 1 :81, c'est-à-dire qu'il y a environ 81 transactions non-frauduleuses pour chaque transaction frauduleuse.

Notre approche a permis de trouver un ratio équilibré entre le nombre d'observations positives et le nombre d'observations négatives, en augmentant le nombre d'observations positives, jusqu'à atteindre un ratio équilibré de 1 :1 après un nombre de générations N_{gen} =100, permettant ainsi au modèle qui sera utilisé par la suite pour la classification d'être plus efficace lors de la détection des fraudes. La Figure 1.17 schématise le processus de génération des nouvelles observations moyennant les opérateurs génétiques.

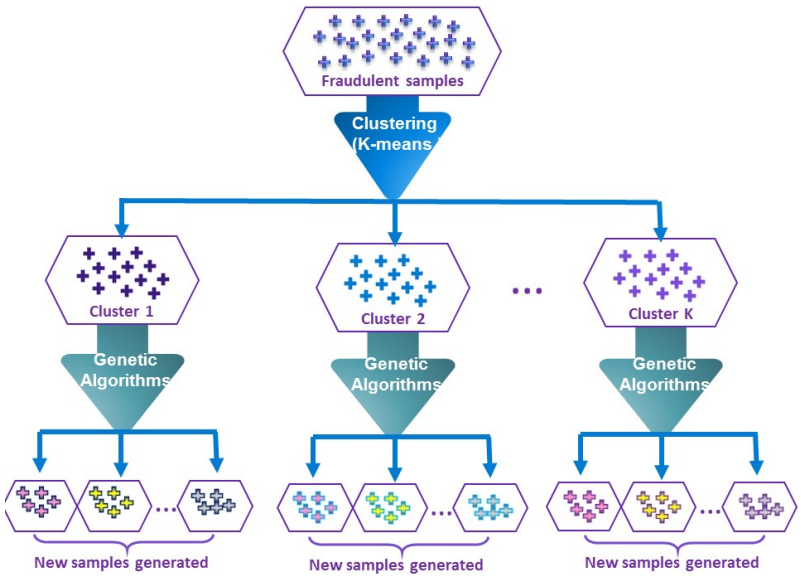

FIGURE 1.17 – *Procéssus de génération par l'AG des nouvelles observations frauduleuses*

Les étapes de notre algorithme sont décrites comme suit :

Entrée: k=4, le nombre de clusters à considérer, déterminé par la méthode Elbow

1. Avec k=4, les observations frauduleuses sont réparties en quatre clusters C_i en utilisant la méthode k-Means, avec 1 = 1 à 4.
2. La valeur de santé (fitness) de chaque observation est mesurée pour chaque population C_i.
3. On procède par la suite à un pairage aléatoire des individus m_i à l'intérieur de chaque cluster C_i , le nombre de paires est de $\frac{m_i}{2}$.
4. Chaque paire d'individus de la population C_i va subir un croisement selon une probabilité de croisement P_c donnée.
5. Après le croisement des parents, chaque individu (enfant) peut subir une mutation, selon une probabilité de mutation P_m donnée
6. Ce processus est répété le nombre de fois nécessaire pour atteindre la taille souhaitée de la population frauduleuse et obtenir ainsi un dataset d'apprentissage équilibré.

Algorithm 4: L'algorithme décrivant les étapes de notre approche

4.3 Algorithmes de classification

Dans le cadre de ce travail, nous divisons notre dataset en dataset de test et dataset d'apprentissage. Les données sont réparties de façon à ce que le ratio de répartition soit de 70 :30. Nous utilisons alors ce dataset pour les différentes étapes de notre expérimentation telles que le clustering, le sampling et la classification.

En vue d'évaluer notre approche, les trois méthodes comparées à notre dataset sont :

- Le classifieur Random Forest [Bolton2002], appliqué au dataset initial ;
- Le classifieur Random Forest, appliqué au dataset équilibré avec la méthode SMOTE ;
- La méthode d'Ensemble Adaboost.

4.4 Mesures de performance

En vue d'évaluer les différentes méthodes présentées dans cette thèse, la matrice de confusion du tableau 1.2 est calculée à partir du dataset de test en comparant la prédiction fournie par le modèle avec la valeur réelle. Pour un modèle parfait, toutes les observations positives seraient prédites comme positives et toutes les observations négatives seraient prédites comme négatives : FN et FP seraient alors égaux à 0.

TABLEAU 1.2 – *Matrice de confusion de classification.*

	positif réel y = 1	négatif réel y = 0
Prédiction positive c = 1	True positive (TP)	False positive (FP)
Prédiction négative c = 0	False negative (FN)	True positive (TN)

- Les vrais positifs (TP) représentent les cas classés comme positifs et qui le sont réellement.
- Les vrais négatifs (TN) sont les cas classés correctement comme négatifs.
- Les faux positifs (FP) sont des cas classés positifs alors qu'ils sont en réalité négatifs.
- Les faux négatifs (FN) sont des cas classés négatifs mais qui sont en réalité positifs.

Le taux d'exactitude (Accuracy) représente le pourcentage des observations correctement classées pour les deux classes :

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{1.1}$$

Selon la formule présentée ci-dessus, nous remarquons qu'en cas de déséquilibre, l'exactitude (Accuracy) est biaisée en faveur de la catégorie majoritaire, à savoir les TN.

La sensibilité (Sensitivity), aussi appelée taux de vrais positifs (TPR) et rappel (Recall), représente la proportion de valeurs positives classifiées correctement comme positives par rapport au nombre total de prédictions positives. Ce paramètre est d'une importance cruciale et sera considéré comme une mesure de performance avec l'exactitude (Accuracy). Il est défini comme suit :

$$Sensitivity = \frac{TP}{TP+FN} \tag{1.2}$$

Notons que, considérer la mesure de sensibilité (Sensitivity) seule est également trompeur, cela permet d'ignorer un grand nombre de faux positifs (FP). Notre objectif est de parvenir à trouver un équilibre entre ces deux mesures de performances, de telle sorte à obtenir un taux de détection de fraude élevé (Sensitivity), avec la plus grande exactitude (Accuracy) possible. Pour gérer cette problématique, nous considérons des mesures de "conciliation" telles que le score F_1 et la courbe ROC (Receiver Operating Characteristic).

La précision (Precision) est une mesure de performance qui représente le taux de prédictions positives correctes par rapport au nombre total de prédictions. Il est défini comme suit :

$$Precision = \frac{TP}{TP+FP} \tag{1.3}$$

L'équation 1.4 définit le score F_1, il s'agit de la moyenne de deux mesures, le rappel (Recall) et la précision (Precision). La valeur de sortie est comprise entre 0 et 1, où 0 désigne le scénario le plus défavorable et 1 le scénario le plus efficace.

$$F_1 score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{1.4}$$

Il est à noter que ces mesures sont interprétées différemment. En outre, chacune de ces mesures ne peut être utilisée seule pour confirmer la qualité concurrentielle des méthodes.

4.5 Résultats

Nous présentons ci-après les résultats comparatifs de notre modèle proposé avec les méthodes listées dans le paragraphe précédent. Les valeurs de l'exactitude (Accuracy), la sensibilité (Sensitivity) et le score F_1 sont indiquées dans le tableau 1.3.

TABLEAU 1.3 – *Résultats de performance*

Métriques d'évaluation	**Accuracy**	**Sensitivity**	**Score F_1**
Random Forest	94.7%	0.47	0.43
Random Forest + SMOTE	97.3%	0.67	0.70
Random Forest + k-Means + GA	**97.9%**	**0.81**	**0.80**
AdaBoost	97.2%	0.70	0.72

Comme nous pouvons l'observer, notre méthode d'Oversampling via k-Means et les opérateurs génétiques dépasse les autres méthodes en termes de sensibilité et de score F_1 . Cette méthode a obtenu le score F_1 le plus élevé, soit 0,80 et améliore considérablement la sensibilité.

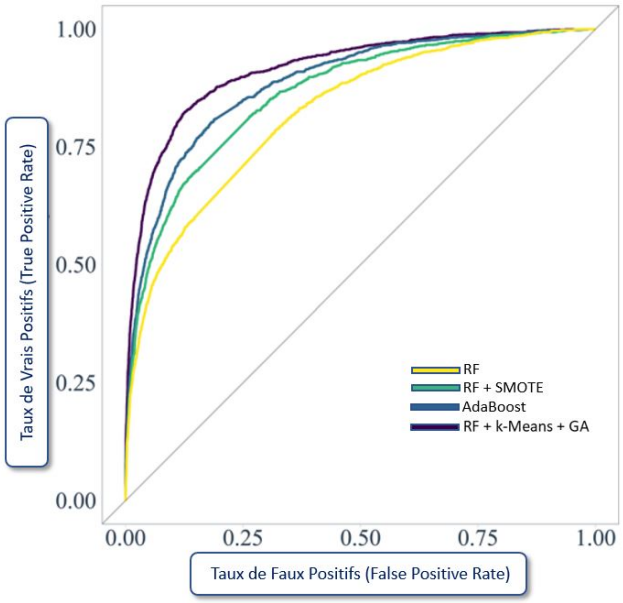

FIGURE 1.18 – *Courbes ROC pour les différentes méthodes*

Pour consolider nos résultats, la courbe ROC est également tracée au niveau de la Figure 1.18. Comme prévu, l'aire sous la courbe ROC est la plus élevée pour notre méthode proposée, ce qui prouve sa capacité à distinguer efficacement les cas de fraudes de ceux qui ne le sont pas. Sur la base de ces résultats, on peut déduire que notre méthode est la plus performante pour la classification du dataset des fraudes par cartes de crédit déséquilibré.

5 Conclusion

Dans ce chapitre, nous avons abordé le problème du déséquilibre des classes et nous avons proposé une nouvelle méthode d'Oversampling basée sur les machines learning. Nous avons fourni une description détaillée de notre méthode hybride, dans laquelle nous avons utilisé la méthode k-Means pour répartir, dans des groupes distincts, les données de la classe minoritaire initiale, puis nous avons appliqué, dans chaque groupe, les opérateurs génétiques afin de générer de nouvelles observations et construire ainsi un dataset équilibré.

Nous avons également comparé les performances de notre méthode, avec la méthode SMOTE et la méthode AdaBoost. La comparaison a été effectuée en appliquant ces méthodes à un dataset de fraudes par carte de crédit déséquilibré, en utilisant plusieurs mesures de performance, en se basant principalement sur la courbe ROC et le score F_1. Les résultats obtenus ont montré que notre méthode est plus performante que toutes les autres méthodes, ce qui résulte en un dispositif efficace de détection de fraude.

Bien que nous ayons présenté notre stratégie dans le contexte de la détection de fraude sur cartes bancaires, celle-ci est plutôt générale et peut être étendue à d'autres domaines d'application caractérisés par des taux de déséquilibre de classes importants.

Outre le problème du déséquilibre des classes, les systèmes de détection de fraude doivent également tenir compte du fait que les comportements d'achat et les stratégies de fraudes évoluent au fil du temps, rendant une fonction de décision apprise par un classifieur non pertinente si celui-ci n'est pas mis à jour, ce qui peut empêcher les systèmes de détection de fraude de conserver une bonne performance.

Dans le chapitre suivant, nous allons mettre en place une architecture de Deep Learning permettant de capturer l'historique des achats à partir de données séquentielles, qui se sont avérées très pertinentes pour la définition des comportements d'achat et des stratégies de fraudes.

2

Utilisation des données historiques pour la définition du contexte d'achat frauduleux

La détection de fraudes par carte de crédit présente plusieurs caractéristiques qui en font une tâche difficile pour le secteur des paiements électroniques. Tout d'abord, les attributs décrivant une transaction ignorent les informations séquentielles qui se sont avérées très pertinentes pour la détection de fraudes bancaires. De plus, les comportements d'achat et les stratégies de fraudes changent au fil du temps, rendant une fonction de décision apprise par un classifieur non pertinente si celui-ci n'est pas mis à jour, ce qui peut empêcher les systèmes de détection de fraudes de conserver une bonne performance. Dans ce chapitre, nous abordons ces défis complexes en utilisant des méthodes de machine learning visant à identifier les transactions frauduleuses. En particulier, nous exploitons les informations séquentielles au-delà des attributs de base d'une transaction et nous cherchons à détecter les fraudes qui sont difficiles à identifier en elles-mêmes, mais particulières en ce qui concerne la séquence dans laquelle elles apparaissent. Pour cela, nous utilisons un réseau de neurones récurrent (LSTM) pour modéliser les données séquentielles des transactions. Les résultats obtenus suggèrent que la modélisation basée sur des réseaux LSTM est une stratégie prometteuse pour caractériser des séquences de transactions et améliorer l'efficacité de la détection des fraudes. La méthode proposée peut être étendue à toute tâche supervisée comportant des datasets séquentiels.

1 Introduction

En raison du volume élevé des transactions par carte de crédit réalisées chaque jour et de la haute dimensionnalité des données collectées, les systèmes de détection de fraudes utilisés pour contrer les activités frauduleuses, doivent être capable, à l'aide des algorithmes de classification, de traiter efficacement une grande quantité de données et d'en extraire uniquement les informations qui sont utiles et pertinentes pour la détection de fraudes par cartes de crédit.

Dans le secteur de la détection de fraudes par carte de crédit, les systèmes traditionnels de détection de fraudes visent à identifier les transactions ayant une forte probabilité d'être frauduleuses, en se basant uniquement sur les informations relatives aux transactions individuelles telles que le montant, l'heure et le lieu de la transaction. Ces systèmes sont inadéquats, dans la mesure où ils ne tiennent pas compte du comportement d'achat du client et des informations séquentielles qui se sont avérées très utiles pour détecter des modèles de fraudes pertinents. [Quah2008]

En effet, une fraude n'est pas seulement une caractéristique de la transaction en elle-même, mais une caractéristique aussi bien de la transaction et du contexte particulier dans lequel elle s'est produite, c'est-à-dire le client et le marchand. Par conséquent, des comportements d'achat semblables peuvent à la fois représenter un comportement tout à fait légitime dans le contexte de certains clients ou des anomalies évidentes dans le contexte d'autres clients [Jurgovsky2018].

Pour construire un tel contexte qui définit le comportement d'achat des consommateurs, il est très important de résumer l'historique des habitudes de consommation des clients, afin de modéliser les corrélations séquentielles entre des transactions consécutives effectuées par carte de crédit. L'objectif étant de réduire le besoin de disposer de connaissances expertes pour la création manuelle des stratégies d'agrégation de transactions et de permettre à un classificateur de mieux détecter des transactions qui sont dissemblables parmi les achats effectués par un consommateur.

Dans ce chapitre, nous présentons une nouvelle approche qui vise à exploiter les informations séquentielles afin de construire automatiquement un tel contexte en utilisant les réseaux de neurones récurrents LSTM. L'approche suggérée permet d'analyser des événements complexes et de découvrir des menaces potentielles ou explicites d'activités frauduleuses en vue d'améliorer la précision de la détection de fraudes sur les nouvelles transactions entrantes. Les résultats obtenus montrent que notre approche est efficace, précise et améliore la performance du modèle de classification.

Ce chapitre est organisé comme suit. La section 2 résume quelques travaux connexes de la littérature. Dans la section 3, les concepts de base utilisés dans cette étude sont présentés. Nous abordons, dans la section 4, les détails de notre méthode proposée et discute les résultats obtenus. Enfin, la section 5 conclut le document et propose des idées pour de futures recherches.

2 Travaux connexes

Dans cette section, nous examinerons plusieurs travaux de recherche récemment menés dans ce secteur. L'étude réalisée par [Srivastava2008] a mis l'accent sur l'utilisation de la méthode Hidden Markov Model initialement entraîné par le comportement normal des titulaires de cartes de crédit. Les chercheurs ont exposé le processus stochastique du modèle et ont mené des recherches sur les tendances de consommation des clients, ce qui a permis d'améliorer la précision du modèle. Ils ont développé le système de telle sorte qu'il puisse être utilisé dans des situations en temps réel afin de vérifier la légitimité d'une transaction bancaire.

Dans [Malini2017], les chercheurs ont effectué une comparaison simple des modèles de clustering et de classification des données de cartes de crédit. Ils ont analysé les principaux modèles de détection des anomalies utilisés dans divers articles de recherche. Ils n'ont pas mené une analyse expérimentale sur un dataset de cartes de crédit et par conséquent ils n'ont pas publié de métriques pour évaluer la performance des modèles.

[Bolton2002] ont proposé deux techniques de clustering : La technique d'analyse peer group et la technique break-point en vue de détecter des comportements frauduleux. Ces deux méthodes ont permis de capturer des comportements suspects indiquant une transaction frauduleuse. [Weston2008] ont appliqué également l'analyse peer group sur des données réelles de transactions par carte de crédit pour identifier les transactions anormales et suspectes.

[RamaKalyani2012] et [Benchaji2019] ont appliqué une méthode de programmation génétique pour détecter les transactions frauduleuses par cartes de crédit. [Bentley2000] ont présenté un système de détection dit fuzzy Darwinian basé sur une programmation génétique permettant de produire des règles de fuzzy logic.

[Sahin2010] ont comparé dans leur étude, les arbres de décision (utilisant trois algorithmes : CART, C5.0 et CHAID) avec la méthode Support Vector Machine (avec des fonctions à kernel linéaire, sigmoïde, polynomial et radial) et ont conclu que les arbres de décision (en particulier l'algorithme CART) étaient plus performants que la méthode Support Vector Machine. [Ganji2012] ont suggéré de détecter les fraudes par carte de crédit à l'aide d'un algorithme de détection des anomalies au sein des flux de données, en se basant sur la méthode des k plus proches voisins inversés.

[Whitrow2009] ont analysé l'aggrégation des transactions et ont prouvé son efficacité. Ils ont conclu que la méthode Random Forest est plus performante que d'autres méthodes telles que Support Vector Machines, Logistic Regression et K-nearest neighbors. Une autre méthode utilisée dans les systèmes de détection des fraudes est le classifieur d'ensemble bagging, [Zareapoor2015] explorent cette méthode avec des résultats positifs. Ils utilisent des algorithmes d'arbre de décision pour créer le modèle. Ils ont également travaillé sur le traitement du déséquilibre des classes et ont testé les modèles avec différents nombres de cas de fraudes allant de 3% à 20%. Le modèle a obtenu les mêmes résultats dans les différents cas de test.

L'utilisation des techniques de Feature engineering en vue de détecter des fraudes était l'objectif principal de [Zhao2019]. Ils utilisent la propagation des labels combinée aux caractéristiques du réseau pour extraire les caractéristiques des fraudeurs. Ils prouvent que la logique du fraudeur est fondée sur un réseau. Sur la base de leur modèle, ils attribuent un score de fraude à chaque nœud. Ce score détermine la probabilité que le nœud en question commette une fraude. Le réglage des poids et la méthode d'initialisation ont été modifiés pour améliorer le modèle. Une fois ces informations réunies, ils mettent en œuvre des techniques comme le calcul de graphe et l'apprentissage automatique pour améliorer le score du modèle.

[Zheng2018] ont extrait les différents comportements des usagers à partir des transactions effectuées en ligne. Cette méthode tient également compte de la diversité des comportements en ligne des utilisateurs. Elle est ensuite utilisée pour dépasser les limites des modèles à chaîne de Markov. Les résultats obtenus avec le dataset prouvent également la même chose. Une amélioration majeure est possible par l'utilisation des algorithmes de clustering pour classer les différentes caractéristiques des transactions. Les auteurs peuvent également utiliser d'autres sources de données comme les commentaires des utilisateurs et leur historique des clics pour améliorer la précision.

Une étude détaillée de différents modèles d'apprentissage automatique a été réalisée par [Randhawa2018] pour détecter les fraudes. Les auteurs ont utilisé la méthode d'ensemble afin de combiner plusieurs modèles et de comparer les résultats. Les modèles qu'ils ont comparés sont les modèles bayésiens, les arbres de décision, les réseaux de neurones, la régression logistique et les machines à vecteurs de support. Cette étude est axée sur le modèle Adaboost. Le coefficient de corrélation de Matthews est la métrique utilisée. Ils obtiennent un score parfait de 1,0 en utilisant Adaboost et le vote majoritaire. Ils ont également ajouté diverses proportions de bruit aux données afin d'être sûrs des résultats.

[Jiang2018] utilisent les modèles de comportement des utilisateurs pour établir un profil de l'utilisateur. Étant donné que le système dispose d'une fonction de retour d'information, le classifieur adapte ses résultats en fonction du flux de transactions. Le profil change également de façon dynamique en fonction des nouveaux comportements de l'utilisateur. Dans l'expérience, ils ont amélioré la précision de deux modèles de base à plus de 80%.

Plusieurs autres recherches se sont concentrées sur l'application des réseaux de neurones pour détecter et prévenir les transactions frauduleuses [Ghosh1994] [Dorronsoro1997] [Zaslavsky2006]. Les auteurs [Syeda2002] ont proposé d'utiliser des réseaux de neurones fuzzy en parallèle sur des machines afin d'accélérer la génération des règles nécessaires à la détection de fraudes par cartes de crédit. [Maes2002] ont comparé les réseaux de neurones artificiels et les réseaux Bayesian belief sur des données réelles et ont montré que les réseaux Bayesian belief détectent 8% de plus de transactions frauduleuses que les réseaux de neurones artificiels.

Cependant, le principal inconvénient de ces approches traditionnelles est qu'elles ne parviennent pas à capturer les dépendances temporelles à long terme entre les transactions bancaires et nécessitent d'extraire manuellement des caractéristiques à partir de données transactionnelles de base. Ce processus est donc subjectif et risque d'entraîner une perte d'informations contenues dans ces données.

Dans ce travail, nous proposons d'utiliser les méthodes d'apprentissage profond basées sur les Réseaux de Neurones Récurrents (RNN) compte tenu de leur réputation comme l'un des algorithmes d'apprentissage les plus efficaces dans le traitement des séquences [Rumelhart1986] [Elman1990] [Graves2014]. En effet, RNN est une approche de machine learning évolutive capable d'analyser les comportements temporels dynamiques de différents comptes bancaires en modélisant la dépendance séquentielle entre les transactions consécutives [Graves2012].

Plus précisément, nous nous intéressons à l'une de ces variantes, les réseaux à mémoire à long terme (LSTM), qui a suscité un grand intérêt dans différents domaines en raison de sa capacité à apprendre les dépendances à long terme contenues dans les séquences. Cet avantage a conduit à un large succès dans des applications pratiques telles que la reconnaissance vocale et la traduction automatique [Graves2014] [Sutskever2014]. Dans le domaine financier, l'utilisation des LSTMs pour la détection de fraudes est plutôt limitée, nous experimentons donc dans ce travail, les avantages de l'utilisation des réseaux LSTMs pour la classification des séquences de transactions légitimes et frauduleuses.

3 Concepts de base

3.1 Les réseaux de Neurones

Les êtres humains ont l'incroyable capacité d'apprendre de nouvelles tâches simplement en considérant des exemples, sans être spécifiquement programmés pour cette tâche. Dans le cerveau humain, ce type d'apprentissage est dû à la communication entre des millions de neurones, des cellules cérébrales capables de communiquer instantanément entre elles grâce à des messagers chimiques appelés neurotransmetteurs. Ces neurotransmetteurs peuvent soit stimuler l'activité électrique de la cellule cérébrale, soit ralentir cette activité.

Un réseau de neurones artificiels (ANN) est un modèle inspiré du fonctionnement des neurones présents dans le cerveau humain, où chaque neurone est une fonction mathématique qui collecte et classifie les informations selon une architecture spécifique. Dans les réseaux de neurones artificiels, les neurones sont organisés en couches et dans chaque couche, à la place d'une impulsion électrique, une valeur numérique est transmise d'un neurone à l'autre.

Il existe en général trois types de couches dans un réseau de neurones : une couche d'entrée, où chaque neurone représente les variables prédictives plus une constante ; une ou plusieurs couches cachées ; et une couche de sortie, avec un ou plusieurs neurones représentant la ou les variables sortantes. Les cellules sont reliées par des poids changeants qui déterminent la robustesse de la connexion. Les données d'entrée sont fournies en entrée de la première couche, et les valeurs sont transmises depuis chaque neurone vers tous les autres neurones de la couche suivante jusqu'à ce que l'on obtienne le résultat escompté dans la couche finale. La figure 2.1 représente la structure d'un ANN en illustrant les différentes couches du réseau.

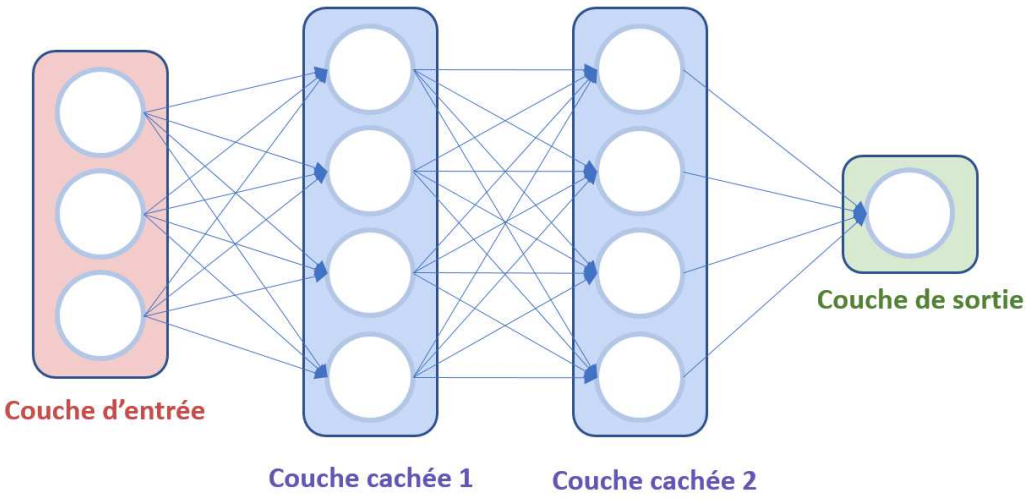

FIGURE 2.1 – *Architecture d'un réseau de neurones*

Le réseau de neurones apprend par la génération d'une prédiction pour chaque ensemble d'entrées et par la modification des poids dans chaque nœud jusqu'à ce que la prédiction exacte soit obtenue. Ce processus est répété sur une base itérative jusqu'à ce qu'un niveau acceptable de performance du modèle soit atteint. Chaque nœud du réseau est appelé perceptron et apprend en recevant des entrées et en affectant des poids à chacune de ces entrées, puis en appliquant une fonction d'activation et en transmettant la sortie au nœud suivant. La figure 2.2 représente le processus d'apprentissage pour chaque perceptron.

Initialement, tous les poids sont déterminés aléatoirement, et les prédictions de sortie ne sont donc pas précises. Le modèle apprend grâce au processus d'apprentissage où des observations pour lesquelles la sortie est connue sont présentées de façons répétitives au réseau, et la sortie prédite est comparée à la sortie connue. L'erreur calculée est ensuite renvoyée au réseau et les poids sont modifiés pour essayer de réduire l'erreur au maximum. Cette technique est connue sous le nom de rétro-propagation (back-propagation) et joue un rôle important dans l'amélioration des performances du modèle. Au fur et à mesure que l'entraînement se fait, le réseau devient de plus en plus précis dans la reproduction des sorties connues. Une fois entraîné, le réseau sera capable de prédire des cas futurs avec des sorties inconnues.

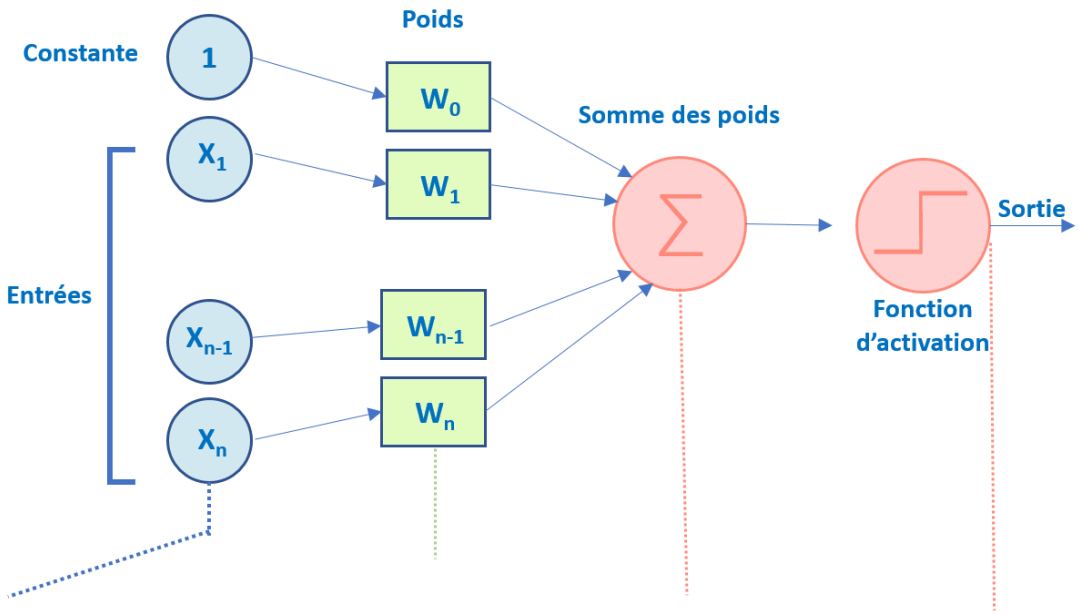

FIGURE 2.2 – *Forme mathématique d'un réseau de neurones*

Parmi les paramètres à considérer lors de la conception d'un réseau de neurones figurent le taux d'apprentissage (learning rate), le type de fonction d'activation et l'optimiseur. Le taux d'apprentissage est un paramètre qui permet de contrôler le taux de changement des poids du réseau. Pendant la rétropropagation, les poids sont modifiés de façon accélérée en fonction de l'erreur calculée. Lorsque le changement de poids est élevé, il peut entraîner des sauts plus importants et manquer les minima globaux. Il est donc essentiel de maintenir le bon taux d'apprentissage.

Les fonctions d'activation sont principalement utilisées pour introduire une certaine non-linéarité dans l'algorithme du réseau de neurones. Il existe une variété de fonctions d'activation utilisées telles que Sigmoïde, Tanh, ReLU, etc. ReLU est la fonction d'activation la plus utilisée et est utilisée lorsque les valeurs négatives doivent être interprétées comme des zéros. Les paramètres internes d'un modèle, tels que le poids et le biais, sont essentiels à la conception d'un bon modèle.

Les techniques d'optimisation jouent un rôle majeur dans le processus d'apprentissage du modèle en cherchant la solution optimale tout en minimisant les pertes. L'optimiseur Adaptive Moment Estimation (Adam) est une méthode qui calcule des taux d'apprentissage adaptatifs pour chaque paramètre. Dans la pratique, il a été constaté que l'optimiseur Adam fonctionne bien pour minimiser la fonction de perte et dépasse les autres techniques.

3.2 Les Réseaux de Neurones Récurrents RNN

Les réseaux de neurones récurrents sont une classe de réseaux de neurones artificiels créés dans les années 80 [Rumelhart1986] [Werbos1988] [Elman1990] dans le but de modéliser les informations de séries temporelles. La structure d'un RNN est similaire à celle d'un perceptron multicouches standard, avec en plus la capacité de connecter des unités cachées qui sont associées à des pas de temps discrets. Les pas de temps indexent les différents éléments d'une séquence d'entrées.

Par le biais des connexions entre ces pas de temps, le modèle peut retenir les informations sur les entrées précédentes, ce qui lui permet de découvrir des corrélations temporelles entre des événements qui peuvent être très éloignés les uns des autres dans la séquence d'entrées. Il s'agit d'une propriété cruciale pour l'apprentissage approprié des séries temporelles où l'occurrence d'un événement dans le passé fournit des informations sur les événements futurs (voir Figure 2.3).

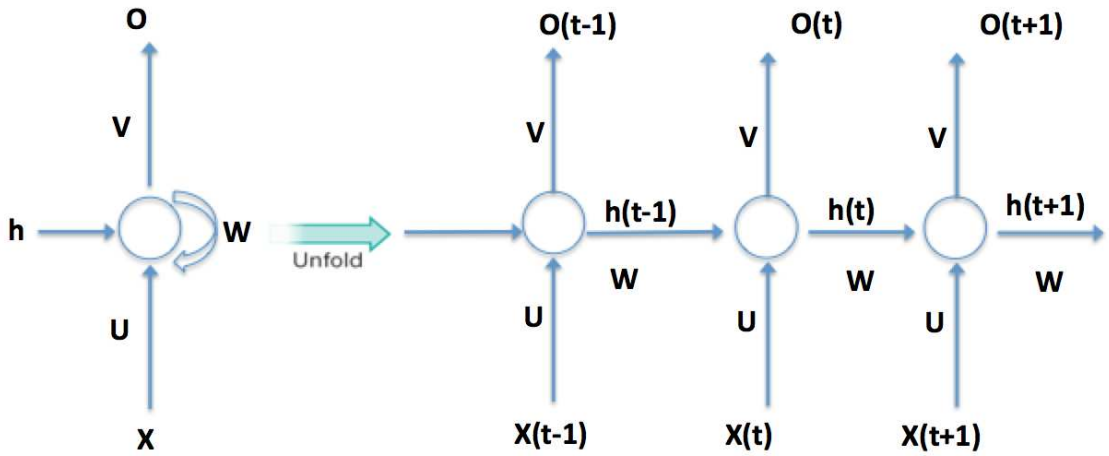

FIGURE 2.3 – *Structure d'un réseau de neurones récurrent déroulé dans le temps en créant une copie du modèle pour chaque pas de temps.*

Un réseau de neurones récurrent de type Vanilla est initialisé avec un vecteur d'état h_0, généralement constitué uniquement de zéros, puis reçoit une séquence $x_{1:T}$ de vecteurs d'entrée x_t, indexés par des valeurs entières positives t. Le RNN parcourt ensuite une séquence de vecteurs d'état h_t, déterminée par l'équation de récurrence suivante :

$$h_t = \sigma(W.h_{t-1} + U.x_t + b) \tag{2.1}$$

où les paramètres d'apprentissage du modèle sont la matrice de poids récurrente W, la matrice de poids d'entrée U et les biais b. Les hyperparamètres du modèle sont les dimensions des vecteurs et des matrices, et la fonction d'activation non linéaire. Le mapping depuis une séquence d'entrée vers une séquence d'état est typiquement appelé couche récurrente dans la terminologie des réseaux de neurones. Les RNN peuvent avoir plusieurs modèles de ce type superposés les uns au-dessus des autres. Pour l'apprentissage, nous spécifions maintenant en plus une fonction de coût et un algorithme d'apprentissage.

Une fonction de coût E_T mesure la performance du réseau pour une tâche déterminée après avoir examiné T vecteurs d'entrée et être passé à l'état h_T. La distribution entre les classes *fraude* et *non-fraude*, compte tenu de l'état h_T, est représentée par un modèle de sortie basé sur une

régression logistique (Logistic Regression) qui génère des prédictions $\widehat{o_T} = p(o_T|h_T)$. Nous interprétons le label réel $o_t \in 0,1$ d'une transaction par la probabilité d'appartenance de x_t à la classe 0 ou 1 et mesurons le coût engendré par les probabilités $\widehat{o_t}$ prédites par le modèle moyennant l'erreur d'entropie croisée (cross-entropy error), défini par l'équation 2.2 :

$$E_T = \iota(x_{1:T}, o_T) = -o_T log\widehat{o_T} - (1-o_T)log(1-\widehat{o_T}) \tag{2.2}$$

Les paramètres du modèle $\theta = (W,U,b)$ sont appris en minimisant le coût E_T avec une méthode d'optimisation basée sur le gradient. Une approche qui peut être utilisée pour calculer les gradients nécessaires est la Backpropagation Through Time (BPTT). Cette méthode consiste à dépiler un réseau récurrent dans le temps pour le représenter comme un réseau multicouches profond comportant autant de couches cachées qu'il y a de pas de temps (voir Figure 2.3). Ensuite, l'algorithme réputé de Backpropagation [Rumelhart1986] est appliqué au réseau déroulé.

Bien que le réseau de neurones récurrent soit un modèle simple et puissant, dans la pratique il présente des difficultés lors de la phase d'entraînement. En effet, pour entraîner un RNN, la rétropropagation à travers le temps est souvent utilisée (Back-Propagation Through Time), cependant celle-ci provoque des problèmes de *vanishing* et d'*explosion* du gradient [Bengio1994] étant donné que la BPTT consiste à représenter le réseau comme un réseau multicouches profond qui empile le RNN initial autant de fois qu'il y a de pas de temps (voir Figure 2.3). Le réseau artificiel sur lequel on applique la rétropropagation devient alors très profond.

La mise en place de cellules de mémoire permet de surmonter le problème de la disparition et de l'explosion du gradient. La structure de réseau qui comprend des cellules de mémoire est appelée réseau de mémoire à long terme (LSTM) [Bayer2015]. Les LSTM se sont révélés capables d'apprendre des dépendances à long terme dans des séquences avec un grand succès pour des tâches séquentielles telles que la reconnaissance vocale et la traduction automatique [Graves2014] [Sutskever2014] .

3.3 Les réseaux de neurones récurrents à mémoire court-terme et long terme (LSTM)

Les réseaux de neurones récurrents à mémoire court-terme et long terme (LSTM) sont un type particulier des réseaux de neurones récurrents (RNN) conçus pour mémoriser des informations sur de longues périodes de temps en vue de les réutiliser ultérieurement. Les réseaux de neurones récurrents se présentent sous la forme d'une chaîne de modèles de réseaux de neurones répétitifs. Dans le RNN Vanilla décrit ci-dessus, le modèle de récurrence est une couche sigmoïde qui reçoit en entrée le vecteur d'état précédent et le vecteur d'observation actuel, puis fournit en sortie le nouveau vecteur d'état.

Les réseaux LSTM présentent également cette structure en chaîne, mais au lieu d'une seule couche, leur modèle de récurrence est constitué de quatre couches de réseaux de neurones interactives qui utilisent la "mémoire" du LSTM. Les couches du modèle de récurrence peuvent être considérées comme trois types de portes logiques qui suppriment, écrivent et lisent des informations de/vers la mémoire (voir Figure 2.4).

La première couche sigmoïde du LSTM est nommée "porte d'oubli" puisqu'elle décide des informations à supprimer de la mémoire. Elle prend en entrée l'état précédent h_{t-1} ainsi que l'entrée x_t et délivre un nombre entre 0 et 1 pour chaque élément du vecteur mémoire c_{t-1} :

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \tag{2.3}$$

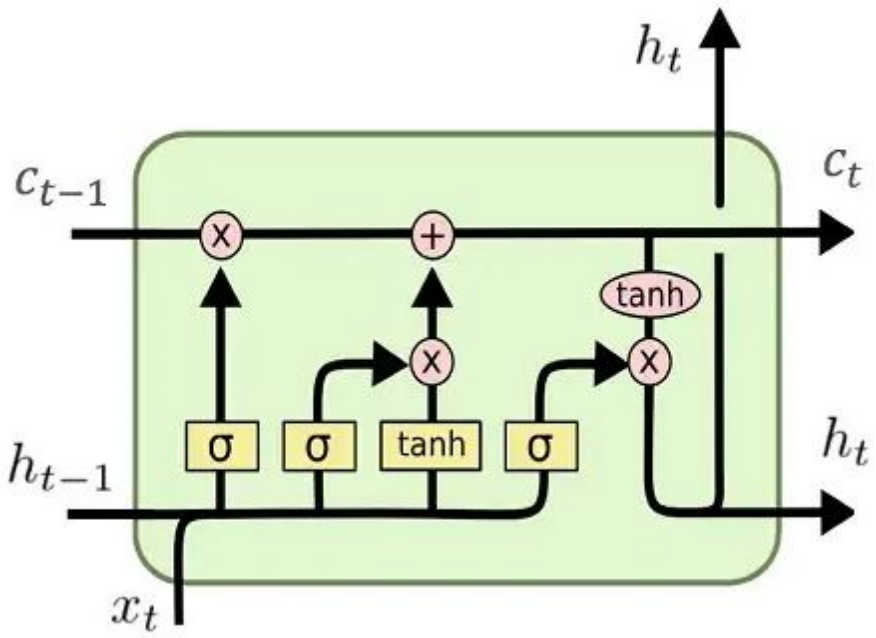

FIGURE 2.4 – *Illustration des calculs exécutés à l'intérieur d'une cellule LSTM. Les flèches noires représentent le flux de données. Les cercles rouges représentent les opérations appliquées sur les vecteurs et les cases jaunes représentent les couches du réseau de neurones et leurs fonctions d'activation, où* σ *désigne la fonction logistique et* tanh *la tangente hyperbolique.*

Une valeur stricte de 0 signifie que l'élément correspondant dans le vecteur mémoire est mis à zéro, tandis qu'une valeur stricte de 1 laisserait la mémoire inchangée. La deuxième porte est appelée "porte d'entrée" et se compose de deux couches de réseaux de neurones. La porte d'entrée décide quelle nouvelle information va être stockée et où elle va être stockée dans la mémoire. Ici, $\tilde{c_t}$ représente le nouveau vecteur mémoire et i_t détermine la proportion par laquelle les données présentes à l'intérieur de la mémoire seront remplacées par la nouvelle information $\tilde{c_t}$:

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \tag{2.4}$$

$$\tilde{c} = \tanh(W_c.[h_{t-1}, x_t] + b_c) \tag{2.5}$$

Une fois les sorties des portes calculées, nous pouvons utiliser ces sorties pour gérer la mémoire courante :

$$c_t = f_t.c_{t-1} + i_t.\tilde{c_t} \tag{2.6}$$

Enfin, le nouvel état h_t du LSTM est une fonction de la mémoire et, comme dans le RNN Vanilla, de l'état précédent h_{t-1} et de l'entrée x_t :

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o) \tag{2.7}$$

$$h_t = o_t . \tanh(c_t) \tag{2.8}$$

Où la porte d'oubli $f \in \mathbb{R}^d$, la porte d'entrée $i \in \mathbb{R}^d$, la porte de sortie $o \in \mathbb{R}^d$, la mémoire et sa mise à jour $c, \tilde{c} \in \mathbb{R}^d$ et l'état $h \in \mathbb{R}^d$. La variable d'entrée $x \in \mathbb{R}^m$ peut avoir un nombre de dimensions m différent. Les matrices de poids $W_i, W_c, W_o \in \mathbb{R}^{d \times (d+m)}$ avec les vecteurs de biais $b_f, b_i, b_c, b_o \in \mathbb{R}^d$ constituent les paramètres du LSTM. Les crochets [-, -] représentent la concaténation de deux vecteurs.

Bien que le modèle LSTM contienne un mécanisme de récurrence plus complexe, ses paramètres peuvent être appris de la même manière que le modèle RNN classique, en utilisant la rétro-propagation dans le temps avec la descente de gradient stochastique comme procédure d'optimisation. Pour une description plus exhaustive du réseau LSTM, le lecteur intéressé est invité à consulter l'article original réalisé par [Hochreiter1997] .

4 Approche proposée

Dans cette section, nous décrivons le modèle, basé sur l'architecture LSTM, que nous proposons pour la détection de fraudes par cartes de crédit. Notre architecture est composée de deux étapes principales, à savoir (Figure 2.5) :

1. La préparation des données.
2. L'utilisation du classifieur séquentiel LSTM pour la détection des fraudes.

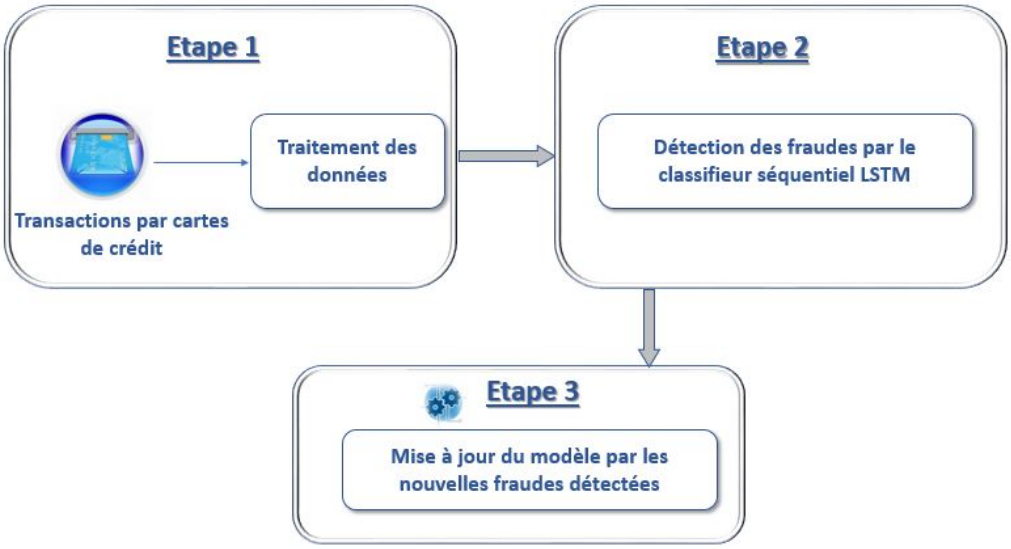

FIGURE 2.5 – *LSTM pour la détection des fraudes*

4.1 Préparation des données

Les valeurs des caractéristiques du dataset qui serviront comme entrées aux réseaux de neurones appartiennent à des échelles et des intervalles différents. Ces différences peuvent varier

considérablement et affecter les performances du classificateur. La normalisation des données s'effectue alors en ajustant minutieusement les caractéristiques d'entrée afin d'aligner la totalité de la distribution de probabilité des valeurs. Dans le cas contraire, l'algorithme sera biaisé en faveur des caractéristiques ayant des valeurs plus grandes [Han2006].

Pour le réseau de neurones récurrents LSTM, nous choisissons la technique de normalisation MinMax, ce qui signifie que chaque donnée d'entrée est normalisée dans l'intervalle de valeurs [0,1]. Cette technique réduit les effets du bruit et garantit que les réseaux de neurones actualisent efficacement leurs paramètres et accélèrent leur apprentissage [Basheer2000]. La formule correspondante est la suivante 2.9 :

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{2.9}$$

Où x_{new} est la valeur normalisée de x et x_{max} et x_{min} sont respectivement les valeurs maximale et minimale de x.

4.2 Le modèle LSTM pour la détection des fraudes sur cartes de crédit

Nous utilisons les réseaux avec mémoire longue LSTM pour modéliser la dépendance séquentielle entre les transactions consécutives de chaque détenteur de cartes de crédit. L'architecture à états cachés des LSTM permet d'établir des connexions entre les nœuds des réseaux de neurones à travers des étapes temporelles. Par conséquent, le modèle peut retenir les informations des entrées passées et identifier les associations temporelles entre des événements qui peuvent être dispersés dans la séquence d'entrée [Elman1990].

Afin de regrouper les observations du dataset, que nous avons décrit dans le chapitre précédent, et de les transformer en séquences appropriées pour la représentation et la classification du modèle, nous suivons les étapes suivantes :

1. Regrouper les transactions par compte et compter le nombre de transactions pour chaque compte.
2. Diviser les comptes en différents ensembles en fonction de leur nombre de transactions.
3. Classer les transactions par heure pour chaque compte dans chaque ensemble.

Notre modèle proposé est entraîné en se basant sur les données historiques des cartes de crédit qui contiennent les détails des achats du titulaire de la carte. À l'aide de ces données, le classifieur séquentiel LSTM compare les informations de la transaction entrante avec les informations précédemment stockées. Si les données correspondent au modèle, alors la carte est certainement utilisée par son propriétaire. S'il n'y a pas de correspondance, la probabilité de fraude est alors élevée.

4.3 Mise en oeuvre de l'approche

Nous construisons un réseau LSTM de détection de fraudes avec 9 neurones d'entrée, étant donné que chaque caractéristique d'entrée présente dans notre dataset sera représentée par son neurone d'entrée. La caractéristique "Fraud status" est utilisée comme neurone de sortie. Une couche cachée de 15 neurones a été utilisée pour analyser la structure des réseaux. Le tableau 2.1 présente les valeurs des paramètres utilisés dans le modèle LSTM proposé.

TABLEAU 2.1 – *LSTM : Paramètres d'entraînement.*

Paramètres	Valeurs LSTM
Nombre de caractéristiques	9
Taille de la mémoire	15
Nombre d'époch	100
Taux d'apprentissage	[0.1, 0.4]
Fonction de perte	Cross Entropy
Optimiseur	Adam Optimiseur

Ce modèle est basé sur le framework de deep learning Keras. Les étapes de mise en œuvre du modèle proposé sont détaillées dans l'algorithme 5 :

Entrée: Historique des transactions recueilli jusqu'au moment t : x_1,x_2,...,x_t

Sortie: Prédiction de fraude entrante

Début:

Importer les librairies Keras

Charger le dataset des transactions sur cartes de crédit

Normaliser le dataset en valeurs entre 0 et 1

Transformer le dataset en un tenseur tridimensionnel : échantillons, Nbre de pas de temps, Nbre de caractéristiques.

Définir les paramètres d'apprentissage : taille de la mémoire, taux d'apprentissage, taille du lot et epochs.

Définir la cellule LSTM.

Définir les variables tensorielles pour les vecteurs de poids et de biais.

Répartir le dataset en apprentissage, validation et test.

Calculer la sortie basée sur la fonction d'activation softmax.

Définir la fonction de perte d'entropie croisée et la fonction d'optimisation Adam.

Entraîner le réseau LSTM

Répéter :

Calculer l'erreur d'apprentissage.

Calculer l'erreur de validation.

Mettre à jour les poids et les biais en utilisant la propagation arrière.

Prédire pour le dataset de test en utilisant le LSTM entraîné.

Fin

Algorithm 5: Modèle de prédiction proposé

4.4 Résultats

La fonction de perte utilisée pour le réseau de détection des modèles est l'entropie croisée. La figure 2.6 montre le graphique de performance des sous-datasets d'entraînement et de validation. Dans notre cas, le réseau est bien entraîné puisque la fonction de perte diminue pour les données d'entraînement et de validation.

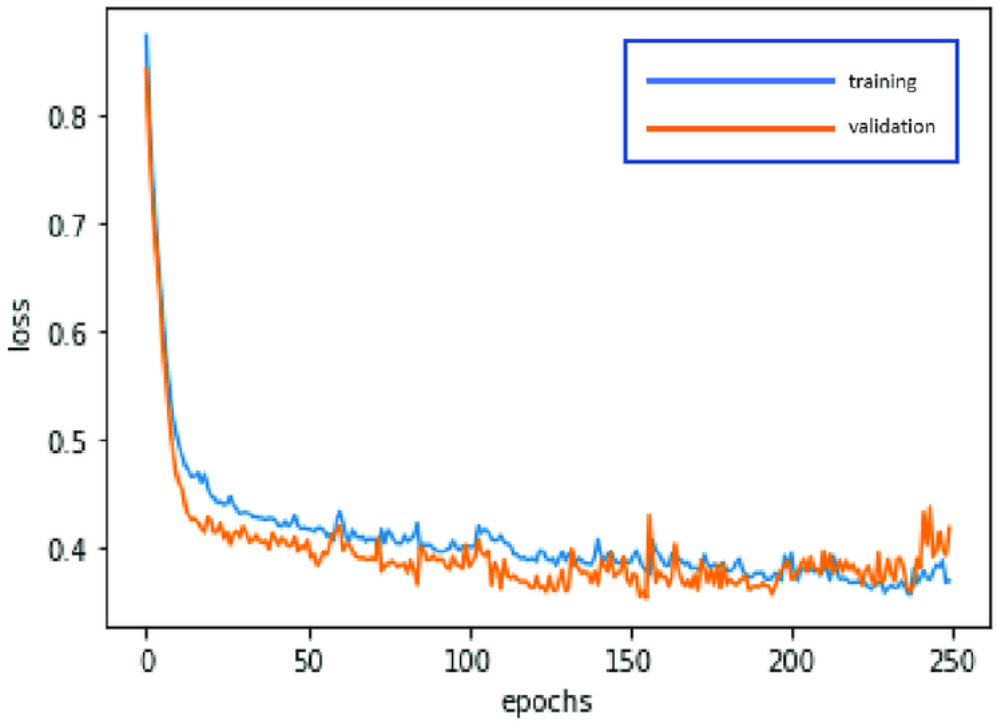

FIGURE 2.6 – *Fonction de perte LSTM*

Nous présentons ci-après les résultats comparatifs de notre modèle proposé avec le classifieur Random Forest [Bolton2002], appliqué au dataset initial. Les valeurs de l'exactitude (Accuracy), la sensibilité (Sensitivity) et le score F_1 sont indiquées dans le tableau 2.2.

TABLEAU 2.2 – *Résultats de performance*

Métriques d'évaluation	**Accuracy**	**Sensitivity**	**Score F_1**
Random Forest	94.7%	0.47	0.43
LSTM	**98.2%**	**0.85**	**0.82**

Comme nous pouvons l'observer, le modèle LSTM appliqué à notre dataset, dépasse la méthode Random Forest en termes de toutes les mesures de performance, montrant surtout qu'il est capable de fournir une haute sensibilité (sensitivity) lors de la détection d'instances frauduleuses qui sont d'un grand intérêt dans le domaine de détection de fraudes.

5 Conclusion

Dans ce chapitre, nous avons utilisé le classificateur séquentiel LSTM afin de capturer le comportement d'achat des titulaires de cartes de crédit à partir de données historiques. L'architecture récurrente des réseaux LSTM améliore les performances de prédiction des fraudes sur les nouvelles transactions entrantes et présente une alternative intéressante à l'agrégation manuelle des caractéristiques (Feature engineering) à partir des données.

Nos résultats montrent que le modèle LSTM produit les meilleures performances de prédiction dans la mesure où il améliore la performance de prédiction en termes de précision et de specificité par rapport à un classificateur de forêt aléatoire (Random Forest) lorsqu'il est entraîné sur les caractéristiques de base.

On déduit que l'architecture LSTM est un modèle adapté aux modèles de succession séquentielle de points de données où l'occurrence d'un événement peut dépendre, plus loin dans le temps, de la présence de plusieurs autres événements. Cependant, il existe encore de nombreux points à améliorer. D'une part, les réseaux LSTM doivent représenter l'ensemble de la séquence d'entrée sous la forme d'un vecteur unique, ce qui peut entraîner une perte d'informations puisque toutes les informations doivent être compressées en ce vecteur, ce qui constitue une tâche extrêmement complexe. D'autre part, les réseaux LSTM traitent les éléments de la séquence d'entrée de la même manière, il n'y a aucun moyen de donner plus d'importance à certains des éléments d'entrée par rapport à d'autres lors du traitement de la séquence.

Pour dépasser ces limites, nous proposons dans le chapitre suivant, d'utiliser les mécanismes d'attention afin de permettre au classifieur séquentiel d'extraire automatiquement les dépendances globales de la séquence de transactions et de se concentrer sur les éléments de données les plus pertinents pour la tâche de classification.

3

Extraction des informations pertinentes à la classification des fraudes bancaires grâce au Mécanisme d'Attention

L'objectif de ce chapitre est d'améliorer les performances du système de détection de fraudes en utilisant les mécanismes d'Attention, capables de se focaliser sur les informations issues des transactions bancaires les plus pertinentes pour la tâche de classification. Le modèle proposé, comparé aux études précédentes, tient compte de la nature séquentielle des données transactionnelles et permet au classificateur d'identifier les transactions les plus importantes dans la séquence d'entrée et qui prédisent avec une plus grande précision les transactions frauduleuses. Précisément, la robustesse de notre modèle est construite en combinant la force de trois sous-méthodes : UMAP (Uniform Manifold Approximation and Projection) pour sélectionner les caractéristiques prédictives les plus utiles, LSTM (Long Short-Term Memory) pour incorporer les séquences de transactions et les mécanismes d'Attention visant à améliorer la classification du modèle. Les performances de notre modèle présentent de bons résultats en termes d'efficience et d'efficacité.

1 Introduction

Dans le secteur des paiements, la détection de fraudes par carte de crédit vise à déterminer si une transaction est frauduleuse ou non sur la base de données historiques [ACFE2018] [Carcillo2018] [Chandola2010]. Cette décision est extrêmement difficile à atteindre pour les raisons suivantes :

1. Les fraudeurs inventent constamment de nouveaux schémas de fraudes, notamment ceux qu'ils utilisent pour s'adapter aux techniques de détection de fraudes.

2. Les modèles de machine learning qui ne sont jamais mis à jour sont inadéquats dans la mesure où ils ne tiennent pas compte des changements et des tendances relatives aux comportements d'achat des clients, par exemple pendant les périodes de vacances et les zones géographiques.

Dans de telles situations, les institutions financières sont appelées à mettre en place un système de détection de fraudes de plus en plus sophistiqué afin d'atténuer la menace actuelle de la fraude et de la détecter sans délai, dans le but de la prévenir avant qu'elle ne se produise, de protéger les intérêts des consommateurs et de réduire les lourdes pertes financières annuelles causées dans le monde entier [Popat2018] [Zafar2018] [Kültür2017] [Dhankhad2018] [Carcillo2021].

Notre contribution dans ce chapitre, est la proposition d'une nouvelle méthode de détection de fraudes dont les principales étapes sont les suivantes :

1. Optimiser le processus d'apprentissage des classifieurs en utilisant des algorithmes de sélection de caractéristiques et de réduction de dimensions tels que PCA (Principal Component Analysis), t-SNE (t-distributed Stochastic Neighbor Embedding) et UMAP (Uniform Manifold Approximation and Projection).

2. Résoudre le problème du déséquilibre des datasets en utilisant la technique d'oversampling des minorités synthétiques (SMOTE).

3. Concevoir un système de détection de fraudes capable de construire un contexte d'achat en utilisant les mécanismes d'Attention, de telle façon à permettre au classifieur séquentiel LSTM de prêter une attention sélective dans la séquence d'entrée et d'améliorer par la suite la décision finale lors de la détection de fraudes.

4. Tester les performances de notre modèle proposé sur deux datasets différents et comparer les résultats obtenus avec les travaux précédents.

Afin d'assurer la reproductibilité de ce travail, le code source et les résultats du modèle proposé peuvent être trouvés sur le lien : https://github.com/bibtissam/LSTM-Attention-FraudDetection.

Le reste du chapitre est organisé comme suit : la section 2 présente les travaux connexes décrivant les travaux antérieurs dans le domaine de la détection de fraudes par carte de crédit, la section 3 décrit le modèle que nous proposons, la section 4 décrit les datasets utilisés dans cette étude et présente les résultats obtenus. Enfin, le document est conclu dans la section 5 et suggère des idées pour les recherches futures.

2 Travaux connexes

Un large panel d'approches de machine learning basées sur l'apprentissage supervisé, l'apprentissage non supervisé, la détection d'anomalies et l'apprentissage d'Ensemble a été utilisé pour la détection des fraudes par carte de crédit [Abdallah2016]. En particulier, les techniques de classification supervisée se sont avérées extrêmement efficaces pour relever ce défi, où des datasets pré-classifiés qui contiennent des transactions historiques labellisées sont utilisés pour entraîner un classifieur capable de construire un modèle de détection en mesure de prédire si une nouvelle transaction est frauduleuse ou non.

Certains de ces algorithmes sont des machines à vecteurs de support (support vector machines) [Bhattacharyya2011] [Dhok2012], des modèles de Markov cachés (hidden Markov models) [Dhok2012] [Srivastava2008], des algorithmes de régression logistique (logistic regression algorithms) [Bhattacharyya2011] [Dal Pozzolo2014b], des arbres de décision (decision trees) [Phua2010] [Sahin2013], des forêts aléatoires (random forests) [Bhattacharyya2011] [Dal Pozzolo2014a] [Bahnsen2016] [Bahnsen2013] [Van Vlasselaer2015] et des k-voisins les plus proches (k-nearest neighbors) [Ganji2012] [Pun2012].

Les méthodes de classification non supervisées sont utilisées pour détecter le comportement inhabituel d'un système et pour identifier les transactions qui ne sont pas conformes au modèle comme des cas potentiels de fraudes [Liu2008] [Zhao2017]. Elles peuvent aider à détecter de nouveaux modèles de fraudes qui n'ont pas été découverts auparavant.

En revanche, la détection des fraudes par carte de crédit soulève de nombreux défis qui suscitent un vif intérêt de la part des chercheurs, et ce pour plusieurs raisons. L'une d'entre elles est le fait que les datasets sur la fraude par carte de crédit sont très déséquilibrés, étant donné que le nombre de transactions légitimes est beaucoup plus élevé que celui des transactions frauduleuses [Hlosta2013] [Benchaji2019].

D'un autre côté, ces classifieurs traditionnels visent à identifier les transactions ayant une forte probabilité d'être frauduleuses, en se basant uniquement sur les informations des transactions individuelles telles que le montant, l'heure et le lieu de la transaction [Donato1999] [Mahmoudi2015] [Minegishi2011], mais ignorent les informations séquentielles détaillées qui définissent le profil d'achat des consommateurs. De tels modèles sont inadéquats pour la détection de la fraude par carte de crédit, car ils ne tiennent pas compte du comportement d'achat, qui est un élément utile pour découvrir des modèles de fraude pertinents qui évoluent dans le temps en raison de la saisonnalité et des nouvelles stratégies d'attaque [Dal Pozzolo2017] [Quah2008].

Récemment, les méthodes d'apprentissage profond basées sur les réseaux de neurones récurrents (RNN) et plus particulièrement sa variante, les réseaux à mémoire à long terme (LSTM), ont été utilisées dans le domaine de la détection des fraudes, étant donné leur réputation comme l'un des algorithmes d'apprentissage les plus précis dans l'analyse des séquences [Rumelhart1986] [Elman1990] [Graves2014] [Benchaji2021a] [Jurgovsky2018].

Le RNN est une approche dynamique d'apprentissage automatique capable d'analyser les comportements temporels de divers comptes bancaires en modélisant la dépendance séquentielle entre les transactions consécutives des détenteurs de cartes de crédit. Cependant, ces modèles RNN doivent représenter l'ensemble de la séquence d'entrée sous la forme d'un vecteur unique, ce qui peut entraîner une perte d'informations puisque toutes les informations doivent être compressées dans ce vecteur. En outre, ils doivent décoder les informations transmises à partir de ce même vecteur uniquement, ce qui constitue une tâche extrêmement complexe.

Dans ce chapitre, nous exploitons les avantages apportées par les mécanismes d'Attention [Bahdanau2014] pour extraire de façon efficace les représentations dépendantes du contexte en se concentrant sur les éléments de données les plus pertinents pour la tâche de classification.

Ces mécanismes utilisés avec un grand succès pour définir le contexte dans les domaines de traduction automatique [Bahdanau2014] et de sous-titrage d'images [Xu2015], tiennent compte des dépendances entre les éléments d'une séquence et permettent de découvrir des corrélations temporelles entre des événements qui seraient très éloignés les uns des autres dans la séquence d'entrée, ce qui améliore l'efficacité de la tâche de classification et augmente la détection des transactions frauduleuses par rapport aux modèles traditionnels.

3 Concepts de base

3.1 Algorithmes de réduction de la dimensionnalité

La sélection et l'extraction des caractéristiques sont des étapes de prétraitement fondamentales dans les systèmes de détection de fraudes [Dal Pozzolo2014a] [Gore2016], afin de sélectionner le sous-groupe optimal de caractéristiques pertinentes tout en éliminant les caractéristiques qui sont redondantes, bruyantes et non pertinentes dans le dataset d'origine, et de réduire le coût de calcul sans effet négatif sur la précision de la classification.

3.1.1 Sélection des caractéristiques

La détection de fraudes par carte de crédit repose sur l'analyse du comportement d'achat des titulaires de cartes de crédit. Ce profil de consommation est analysé en utilisant une sélection optimale de variables qui caractérisent le comportement d'achat unique et discriminent les transactions qui sont différentes parmi les achats d'un client. En outre, comme les profils d'achat légitime ou frauduleux tendent à changer constamment, une sélection optimale de variables qui différencient fortement les deux profils est nécessaire pour parvenir à une classification efficace des transactions par carte de crédit [West2016] [Kamaruddin2016] [Hormozi2013].

Dans ce travail, la méthode bio-inspirée Swarm Intelligence [Brezočnik2018] sera utilisée pour améliorer la qualité des modèles de machine learning en identifiant les caractéristiques les plus importantes et en optimisant la performance globale du modèle. L'une des principales contributions de la méthode Swarm Intelligence est qu'elle peut améliorer l'efficacité de la sélection de caractéristiques en utilisant des techniques d'optimisation pour explorer efficacement l'espace de recherche des caractéristiques pertinentes. Elle peut également aider à éviter les problèmes courants de sur-apprentissage ou de sous-apprentissage en sélectionnant des caractéristiques plus discriminantes et en réduisant le bruit dans les données.

3.1.2 Extraction des caractéristiques

Pour réduire la dimension de nos datasets, nous avons utilisé trois algorithmes, à savoir Principal Component Analysis (PCA) [Jolliffe2016], t-distributed Stochastic Neighbor Embedding (t-SNE) [Linderman2019] [Van der Maaten2008], et Uniform Manifold Approximation and Projection (UMAP) [Becht2019] [McInnes2018]. Ces algorithmes sont considérés parmi les meilleurs algorithmes de réduction de dimensions, utilisés pour l'extraction de caractéristiques dans de nombreuses applications telles que la bio-informatique et la vision par ordinateur [Becht2019].

- **Principal Component Analysis (PCA)**
  PCA est une méthode de réduction de dimensions, visant à transformer l'ensemble initial de n variables en un nouveau sous-ensemble de m variables appelées composantes principales. Ces composantes sont des combinaisons linéaires des variables d'origine et sont dérivées par ordre décroissant d'importance, de sorte que la première composante principale représente le maximum de la variation.
  Etant donné un ensemble de n variables corrélées f_1,f_2,...,f_n, l'objectif de la méthode PCA consiste à remplacer ces n variables mesurées par m variables dérivées z_1,z_2,...,z_m, non corrélées et dont les variances décroissent de la première à la dernière, sans compromettre l'information contenue dans ces données. Cette transformation est effectuée en respectant les propriétés suivantes :

  1. z_1 a la variance maximale possible parmi toutes les variables linéaires possibles de f_1,f_2,...,f_n. L'équation correspondante est donnée par :

  $$z_1 = \alpha_1 f_1 + \alpha_2 f_2 + ... + \alpha_n f_n \tag{3.1}$$

  2. z_2 a une variance maximale possible parmi toutes les variables linéaires possibles de f_1,f_2,...,f_n, sous réserve que z_2 ne soit pas corrélée avec z_1.

  3. En général, z_k a le maximum de variance parmi toutes les variables linéaires possibles de f_1,f_2,...,f_n, sous réserve que z_k soit non corrélée avec z_1,z_2,...,z_{k-1}, pour $2 \leqslant k \leqslant n$.

  Bien que la méthode PCA soit capable de couvrir la variance maximale entre les variables, mais en tant qu'algorithme linéaire, elle peut être peu performante sur les variables ayant une relation non linéaire. C'est pourquoi, afin de projeter des données de grande dimension sur une base de dimension réduite et non linéaire, des algorithmes de réduction dimensionnelle non linéaires tels que t-SNE et UMAP sont utilisés.

- **t-distributed Stochastic Neighbor Embedding (t-SNE)**
  L'intégration de voisins stochastique distribuée en t (t-SNE) est un algorithme de machine learning qui est particulièrement utile pour réduire des données à haute dimension non linéaire dans un espace à deux ou trois dimensions. Il tente de positionner un point d'un espace de haute dimension dans un espace de faible dimension de manière à préserver l'identité du voisinage ; des points de données plus proches signifient une grande similarité.

  La méthode t-SNE comporte deux étapes principales. En premier lieu, elle trouve une distribution de probabilité sur les paires de données, de telle sorte qu'une paire de points de données similaires se voit attribuer une forte probabilité, tandis qu'une paire de points plus éloignés se voit attribuer une faible probabilité. Ensuite, elle définit une distribution de probabilité dans l'espace de dimension réduite qui est similaire à celle de l'espace de dimension initiale, et vise à minimiser la divergence de Kullback-Leibler (KL) entre les deux distributions [Linderman2019].

  Étant donné un dataset d'entrée à haute dimension x_1,x_2,...,x_n dans R^m, notre objectif est de trouver une représentation optimale à faible dimension y_1,y_2,...,y_n dans R^k, telle que $k \leqslant m$. La similarité du point de données x_j avec le point de données x_i est représentée par la probabilité conditionnelle p_{ji}. Pour les contreparties de faible dimension y_i et y_j des points de

données de haute dimension x_i et x_j, on calcule une probabilité corrélative similaire représentée par q_{ji}.

Une fois que p_{ji} et q_{ji} sont calculés, l'objectif de l'algorithme t-SNE est de minimiser l'inadéquation entre les représentations de haute et de faible dimension. La fonction de coût (équation (3.2)) qui minimise les divergences de Kullback-Leibler (KL) sur tous les points est donnée par :

$$KL(P|Q) = \sum_i \sum_j p_{ji} log \frac{p_{ji}}{q_{ji}} \tag{3.2}$$

Où P et Q représentent respectivement les distributions de probabilité pour p_{ji} et q_{ji}.

Bien que l'algorithme t-SNE soit une bonne technique pour visualiser des données dans un espace de faible dimension, il calcule des probabilités corrélées par paires pour chaque paire de données et implique des hyperparamètres qui ne sont pas toujours simples à régler, ce qui entraîne un coût de calcul élevé.

- **Uniform Manifold Approximation and Projection (UMAP) :**
  UMAP (Uniform Manifold Approximation and Projection) est une technique émergente de réduction de la dimensionnalité qui a été récemment publiée par McInnes et Healy [McInnes2018]. Elle est basée sur la théorie de la géométrie de Riemann et de la topologie algébrique qui utilise des approximations locales de manifolds et regroupe leurs représentations locales floues simplifiées pour construire une représentation topologique des données de haute dimension, puis un processus similaire est utilisé pour rechercher une projection de faible dimension des données qui présente la structure topologique floue équivalente la plus proche possible de l'espace initial.

  Contrairement à t-SNE qui utilise un modèle probabiliste, UMAP est un algorithme basé sur les graphes. La première phase de la méthode UMAP consiste à construire une représentation graphique pondérée à k voisins pour chaque point de données original à haute dimension, de telle sorte que l'entropie croisée entre le graphique pondéré et les données originales soit minimisée. Ensuite, les vecteurs propres à k dimensions du graphe UMAP sont utilisés pour représenter chacun des points de données originaux.

  UMAP considère les données d'entrée $X = \{x_1, x_2, ..., x_n\}$ dans R^m, avec une métrique (ou mesure de dissimilarité) d : $X \times X \rightarrow R^+$ et cherche une représentation optimale de dimension réduite $\{y_1, y_2, ..., y_n\}$ dans R^k, telle que $k < m$. Etant donné un hyperparamètre d'entrée k, pour chaque x_i, on calcule l'ensemble $\{x_{i1}, x_{i2}, ..., x_{ik}\}$ des k plus proches voisins de x_i sous la métrique d. Pour chaque x_i, nous définirons ρ_i et σ_i. Soit :

$$\rho_i = min\left\{d\left(x_i, x_{ij}\right) | 1 \leq j \leq k, d\left(x_i, x_{ij}\right) > 0\right\} \tag{3.3}$$

  Où σ_i est défini de telle sorte que :

$$\sum_{j=1}^{k} exp\left(\frac{-max(0, d\left(x_i, x_{ij}\right) - \rho_i)}{\sigma_i}\right) = log_2(k) \tag{3.4}$$

On choisit ρ_i pour s'assurer qu'au moins un point de données est connecté à x_i avec un poids d'arête de 1, ce qui équivaut à ce que l'ensemble flou simplifié résultant soit localement connecté à x_i.

Le paramètre σ_i est défini comme un paramètre d'échelle de longueur, définissant un graphe directionnel pondéré $\bar{G} = (V, E, \omega)$, où V est l'ensemble des sommets (dans ce cas, les données X), E est l'ensemble des arêtes dirigées E $=\{(x_i, x_{ij}) | 1 \leqslant j \leqslant k, 1 \leqslant i \leqslant n\}$, et ω est la fonction de pondération des arêtes définie en fixant :

$$\omega(x_i, x_{ij}) = exp\left(\frac{-max(0, d(x_i, x_{ij}) - \rho_i)}{\sigma_i}\right) \tag{3.5}$$

La méthode UMAP essaye de définir un graphe pondéré non orienté G à partir d'un graphe orienté $\bar{G}$ via la symétrisation. Soit A la matrice d'adjacence du graphe $\bar{G}$. Une matrice symétrique est obtenue par :

$$B = A + A^T - A \otimes A^T \tag{3.6}$$

Où T est la transposée et $\otimes$ désigne le produit de Hadamard (ou pointwise). Ensuite, le Laplacien pondéré non orienté G (le graphe UMAP) est défini par sa matrice d'adjacence B. L'objectif est de trouver les coordonnées optimales en faible dimension $\{y_i\}_{i=1}^n, y_i \in \mathbb{R}^k$, qui minimise l'entropie latérale croisée avec les données initiales à chaque point. L'évolution du graphe UMAP Laplacien G peut être considérée comme une approximation discrète de l'opérateur de Laplace-Beltrami sur un manifold défini par les données [Chen2021]. L'implémentation et de plus amples détails sur UMAP sont disponibles dans [McInnes2018].

Comparé à t-SNE, le modèle UMAP est mieux apte à préserver la structure des données locales et la structure des données globales, avec des performances d'exécution supérieures [McInnes2018].

3.2 Mécanismes d'Attention

Dans les travaux de recherche modernes sur le deep learning, tels que la vision par ordinateur et la traduction du langage [Bahdanau2014] [Chorowski2015], les mécanismes d'Attention sont devenus un moyen efficace pour atteindre des résultats optimums en mettant l'accent sur les informations importantes. Ces mécanismes visent à se concentrer uniquement sur les éléments d'information les plus pertinents, plutôt que sur l'ensemble des informations, ce qui est adéquat pour les traitements neuronaux qui suiveront [Hochreiter1997].

Pour mieux expliquer les mécanismes d'Attention, considérons une architecture RNN encodeur-décodeur : un encodeur lie la séquence de vecteurs en entrée X=(x_1,x_2,...,x_n) à un vecteur c_t. Cette approche est souvent exprimée dans une structure RNN sous la forme suivante :

$$S_t = f(x_t, S_{t-1}, c_t)) \tag{3.7}$$

Et :

$$c = q(S_1, , ..., S_n)) \tag{3.8}$$

où S_t est l'état caché, c_t est le vecteur de sortie du RNN qui est généré par les états cachés. Dans le modèle d'Attention, le vecteur de contexte c_t est fortement lié à la séquence des annotations (h_1,...,h_n) à laquelle un encodeur mappe la séquence d'entrée. L'annotation h_t contient des informations sur toute la séquence d'entrée, avec un focus particulier sur les parties situées aux alentours du t-ième élément de cette séquence d'entrée. Les détails sont présentés dans les paragraphes suivants.

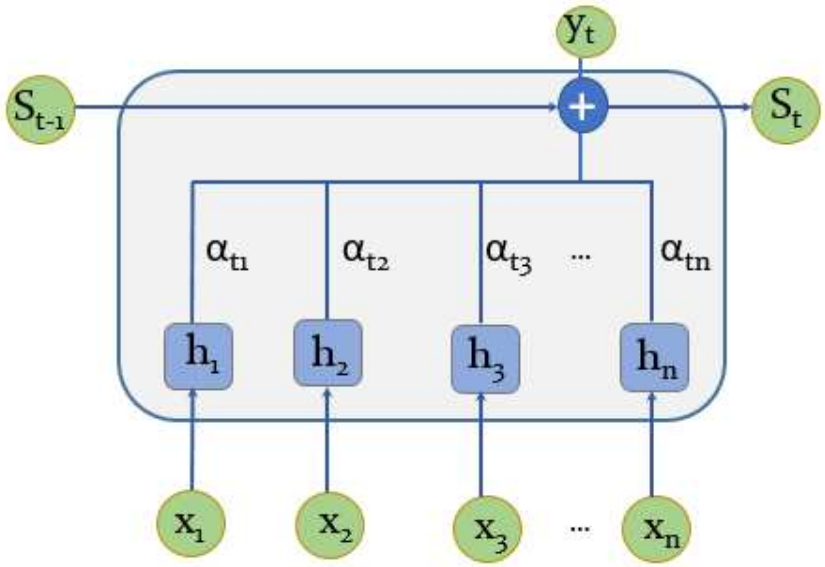

FIGURE 3.1 – *Mécanisme d'Attention*

La figure 3.1 illustre le mécanisme d'Attention dans un réseau de neurones. Une somme pondérée de ces annotations h_t constitue le vecteur de contexte c_t :

$$c_t = \sum_{j=1}^{n} \alpha_{tj} h_j \tag{3.9}$$

Où le poids α_{tj} de chaque annotation h_t est donné par :

$$\alpha_{tj} = \frac{exp(e_{tj})}{\sum_{k=1}^{n} exp(e_{tk})} \tag{3.10}$$

Dans lequel :

$$e_{tj} = a(S_{t-1}, h_j) \tag{3.11}$$

La fonction a(S_{t-1},h_j) est un modèle d'alignement qui décrit la capacité de correspondance entre les entrées autour de la position j et les sorties à la position t. L'état caché du RNN S_{t-1} et la j-ième annotation h_j de la séquence d'entrée sont utilisés pour calculer le score. Le mécanisme d'Attention permet à un réseau de neurones de se concentrer sur un sous-ensemble de ses entrées : il choisit toujours les entrées les plus pertinentes. Le mécanisme d'Attention de la figure 3.1 vise à sélectionner les entrées les plus importantes parmi les séquences d'entrée x_1,x_2,...,x_n en utilisant le poids α_{tj} .

4 Approche proposée

Comme indiqué ci-dessus, le modèle que nous proposons utilise en premier lieu des techniques de pré-traitement de données qui consistent à appliquer une sélection de caractéristiques et une réduction de la dimensionnalité des datasets de fraudes, dans le but de réduire le nombre de caractéristiques d'entrée avant de les introduire dans le modèle. Ensuite, le modèle à apprentissage séquentiel LSTM est utilisé comme classifieur responsable de l'identification dynamique de la dépendance séquentielle entre les transactions bancaires consécutives. Enfin, le mécanisme d'Attention est introduit pour mettre un focus particulier sur les informations importantes issues des couches cachées du réseau de neurones récurrent, ce qui permettra à notre modèle de découvrir des schémas de fraudes pertinents et de mieux détecter les transactions qui sont très différentes dans les achats d'un consommateur. L'architecture du système proposé est illustrée dans la figure 3.2.

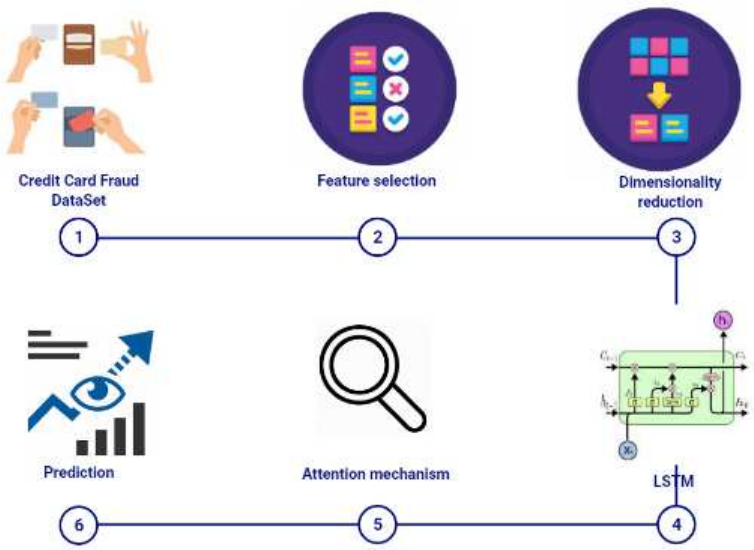

FIGURE 3.2 – *L'architecture du modèle proposé pour la détection des fraudes par carte de crédit*

Les étapes de notre modèle proposé pour la détection de fraudes par cartes de crédit sont détaillées ci-dessous. Nous décrirons les deux datasets que nous utilisons dans nos expériences, les résultats du pré-traitement des données et nous présenterons l'implémentation détaillée et les mesures de performance utilisées dans ce travail.

4.1 DataSets

Dans cette sous-section, nous décrivons deux datasets différents utilisés dans les expérimentations de notre approche proposée. Un bref résumé de ces deux datasets est présenté dans le Tableau 3.1.

TABLEAU 3.1 – *Déscription des datasets des cartes de crédit.*

Nom	Instances	Caractéristiques	Normal	Fraude
Dataset-1	284.807	31	284.315	492
Dataset-2	594.643	10	587.443	7200

4.1.1 DataSet -1

Le premier dataset, téléchargé sur www.kaggle.com, est constitué de transactions par carte de crédit effectuées par des détenteurs de cartes européens au cours de deux jours en septembre 2013. Il compte 492 fraudes sur 284 807 transactions. Il se compose de 31 caractéristiques, dont l'heure à laquelle une transaction a eu lieu, le montant des transactions, et 28 autres attributs labellisés de V1 à V28 et le l'attribut cible " Class" qui détermine si une transaction est frauduleuse ou non par une valeur binaire "1" et "0" respectivement.

4.1.2 DataSet -2

Le second dataset est constitué de 594 643 transactions effectuées pendant 180 jours simulés, parmi lesquelles 7200 (≈1,2%) sont considérées comme frauduleuses. Il s'agit d'un jeu de données synthétiques créé pour la détection de fraudes financières à l'aide du logiciel BankSim, qui est un outil de simulation spécialement conçu pour émuler des données de fraudes [Vaughan2020]. BankSim utilise une méthodologie de simulation multi-agents basée sur un ensemble de données de transactions réelles que propose une banque en Espagne. Les données bancaires initiales sont constituées de milliers d'enregistrements de données transactionnelles de novembre 2012 à avril 2013. BankSim utilise plusieurs agents de trois catégories différentes pour reproduire ces données bancaires initiales : commerçants, clients et fraudeurs. Ces agents communiquent entre eux au cours d'une séquence de jours simulés, ce qui donne lieu à un historique des transactions d'achat se rapprochant étroitement des données bancaires initiales. Tous les attributs sont présentés dans le tableau 3.2.

TABLEAU 3.2 – *Attributs du DataSet-2*

Nom	**Déscription**
Step	Le jour où la transaction a eu lieu de 1 à 180
Customer ID	Un numéro identifiant le compte client concerné par la transaction
Age Category	Une valeur catégorique classant le client dans un des 8 différents groupes d'âge.
Gender	Une variable catégorique indiquant le sexe du client
Zip Code of account	Le code postal associé au client
Merchant ID	Le numéro identifiant le commerçant impliqué dans la transaction
Zip Code of Merchant	Le code postal du commerçant
Category purchase	Une variable catégorielle indiquant le type de bien ou de service acheté
Amount of purchase	Le montant total de la transaction
Fraud status	Une variable binaire indiquant si la transaction est frauduleuse ou non

4.1.3 Traitement des données

Nous pouvons constater que les deux datasets sont fortement déséquilibrés vu que le nombre d'instances négatives (Class 0) est supérieur au nombre d'instances positives (Class 1). En effet, par exemple, dans le DataSet-1, les fraudes représentent généralement moins de 0,171% de l'ensemble des transactions, comme le montre la figure 3.3.

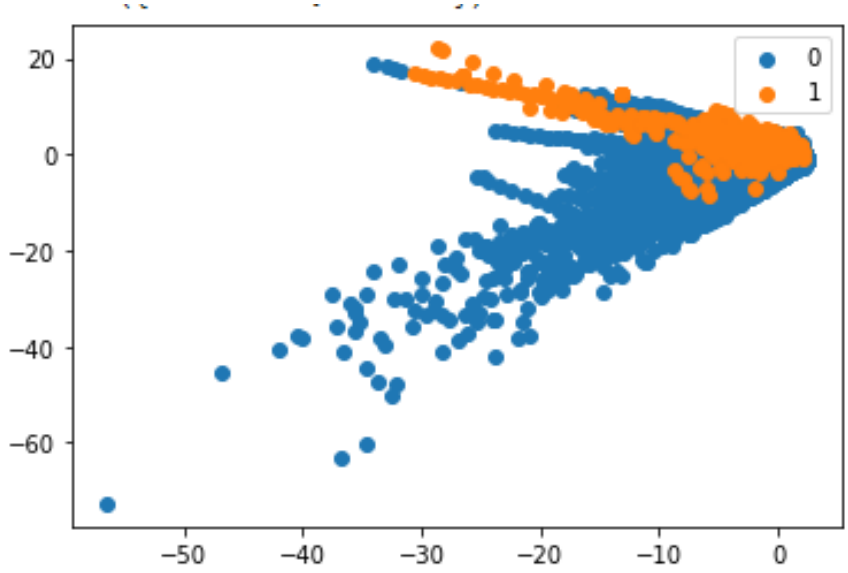

FIGURE 3.3 – *Graphe du dataset des cartes de crédit avant la transformation SMOTE.*

A ce titre, pour améliorer les performances de classification des instances frauduleuses, qui représentent la classe la plus intéressante, nous utilisons la technique de suréchantillonnage appelée Synthetic Minority Oversampling Technique (SMOTE) [Chawla2002] [Kumari2019], détaillée dans le chapitre 1, pour générer des instances d'entraînement synthétiques à partir de la classe minoritaire.

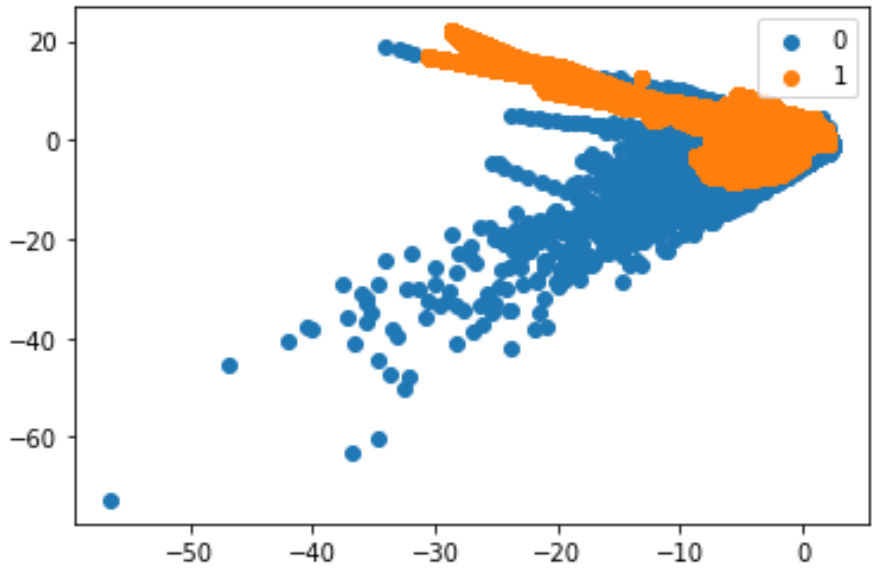

FIGURE 3.4 – *Graphe du dataset des cartes de crédit après la transformation SMOTE*

La technique SMOTE a permis de trouver un ratio équilibré entre le nombre d'observations positives et le nombre d'observations négatives, en augmentant le nombre d'observations positives, jusqu'à atteindre un ratio équilibré de 1 :1. La figure 3.4 présente le schéma du DataSet-1 transformé à l'aide de la méthode SMOTE.

4.2 La réduction de la dimensionnalité

4.2.1 Sélection des caractéristiques

Comme indiqué ci-dessus, nous utilisons la méthode Swarm Intelligence [Brezočnik2018] pour la sélection des caractéristiques comme première étape d'exploration de nos datasets. L'objectif est d'étudier l'influence de chaque caractéristique dans la prédiction de la classe cible et de sélectionner le sous-ensemble optimal de caractéristiques pertinentes en supprimant les attributs redondants et bruyants.

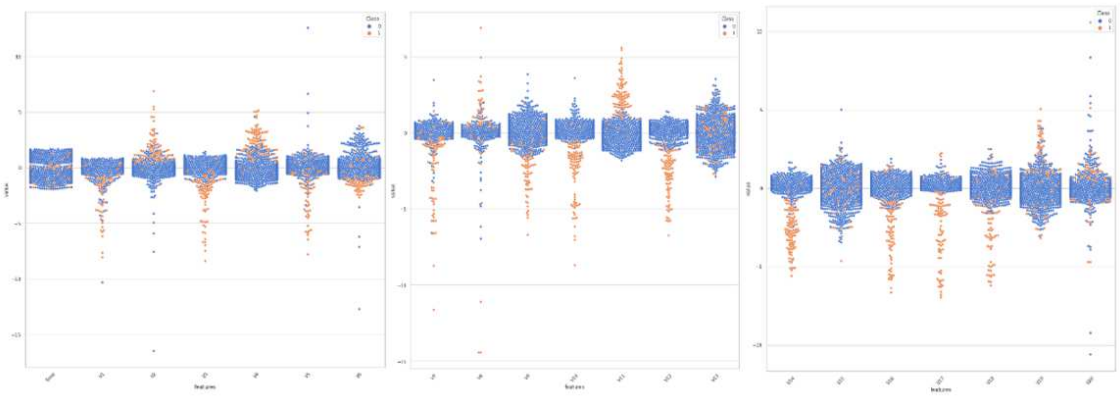

FIGURE 3.5 – *Graphes de l'algorithme Swarm Intelligence*

D'après les graphiques visuels de l'algorithme bio-inspiré Swarm Intelligence (Fig 3.5) appliqué au DataSet-1, nous pouvons observer que l'analyse comparative de ce dataset démontre que les caractéristiques labellisées Time, V5, V6, V7, V8, V9, V13, V15, V16, V18, V19, V20,V21, V22, V23, V24, V25, V26, V27, V28, Amount ne contribuent pas à la prédiction de la fraude.

A ce titre, nous décidons de les considérer comme des attributs non pertinents et de les supprimer du dataset initial. Le tableau 3.3 présente les caractéristiques conservées.

TABLEAU 3.3 – *Les caractéristiques restantes après application de l'algorithme Swarm*

V1	V2	V3	V4	V10	V11	V12	V14	V17

4.2.2 Extraction des caractéristiques

Nous avons appliqué les trois algorithmes de réduction de la dimensionnalité PCA, t-SNE et UMAP sur nos datasets afin d'obtenir les caractéristiques robustes et discriminantes des instances frauduleuses, dans le but de faciliter la détection des transactions illégitimes.

La figure 3.6 montre la performance de chaque algorithme de réduction appliqué sur le DataSet-1. Pour chaque cas, le dataset a été réduit dans un espace tridimensionnel en utilisant les paramètres par défaut, et les graphiques ont été coloriés en fonction du label de chaque observation présente dans le DataSet-1, à savoir la couleur violette est utilisée pour représenter les transactions légitimes et la couleur orange représente les transactions frauduleuses.

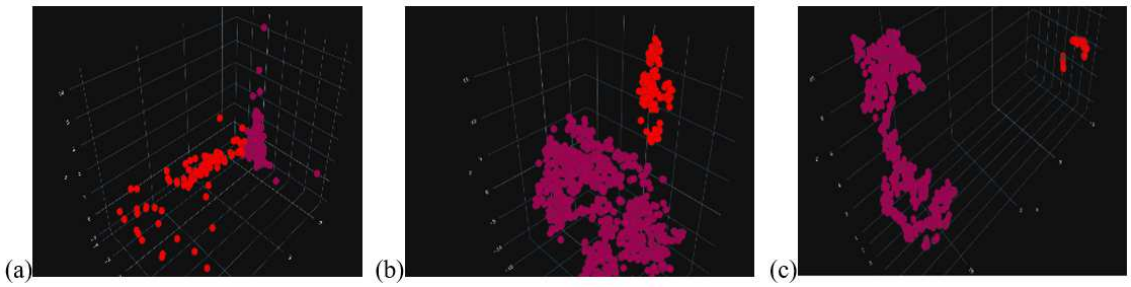

FIGURE 3.6 – *Performances des algorithmes de réduction sur notre dataset de fraudes. La dimension des caractéristiques est réduite à 3 par (a) PCA, (b) t-SNE et (c) UMAP.*

On peut voir que la méthode PCA ne présente pas une bonne discrimination, alors que les méthodes UMAP et t-SNE démontrent une très bonne discrimination. Cependant, en comparant t-SNE à UMAP, ce dernier est plus apte à préserver autant que possible la structure de données locale et globale, avec des performances d'exécution supérieures. Sur cette base, nous choisissons UMAP comme algorithme de réduction pour extraire les caractéristiques de référence qui seront utilisées pendant les phases d'apprentissage et de test.

4.3 Mise en oeuvre de l'approche

Dans ce travail, nous proposons de mettre en place un modèle capable de modéliser la dépendance séquentielle entre les transactions consécutives de chaque consommateur. Ce modèle aura la capacité de conserver les informations issues des entrées passées, et identifier les différentes associations temporelles entre les événements qui sont dispersés dans la séquence d'entrée.

Dans le travail précédent décrit au niveau du chapitre 2, nous avons utilisé les réseaux avec mémoire longue (LSTM) dont l'architecture à états cachés permet d'établir des connexions entre les nœuds du réseau de neurones à travers des intervalles de temps différents. Le LSTM est un modèle de traitement séquentiel où l'occurrence d'un événement peut dépendre, plus loin dans le temps, de la présence de plusieurs autres événements. Cependant, il existe encore de nombreux points à améliorer :

- Les réseaux LSTM doivent représenter l'ensemble de la séquence d'entrée $x_1, x_2, ..., x_n$ sous la forme d'un vecteur unique c, ce qui peut entraîner une perte d'informations puisque toutes les informations doivent être compressées dans ce vecteur c. En outre, ils doivent décoder les informations transmises à partir de ce même vecteur uniquement, ce qui constitue une tâche extrêmement complexe.
- Les réseaux LSTM traitent les éléments de la séquence d'entrée de la même manière, il n'y a aucun moyen de donner plus d'importance à certains des éléments d'entrée par rapport à d'autres lors du traitement de la séquence.

Pour dépasser les limites décrites ci-dessus, nous proposons d'exploiter les avantages apportés par les mécanismes d'Attention et les combiner avec le réseau LSTM afin de permettre au classifieur d'extraire de façon efficace les dépendances globales de la séquence d'entrée et de se concentrer sur les éléments de données les plus pertinents pour la tâche de classification.

Notre modèle est conçu sur la base du framework d'apprentissage profond Keras, qui est une bibliothèque de réseaux de neurones open source écrite en Python. Il est composé de 6 couches à savoir : Deux couches LSTM suivies d'un dropout au niveau de chaque couche, une couche

d'Attention ajoutée après la première couche LSTM comme le montre la Fig 3.7. La deuxième couche LSTM prend la sortie de la couche d'Attention comme entrée avec la fonction d'activation supposée être tanh.

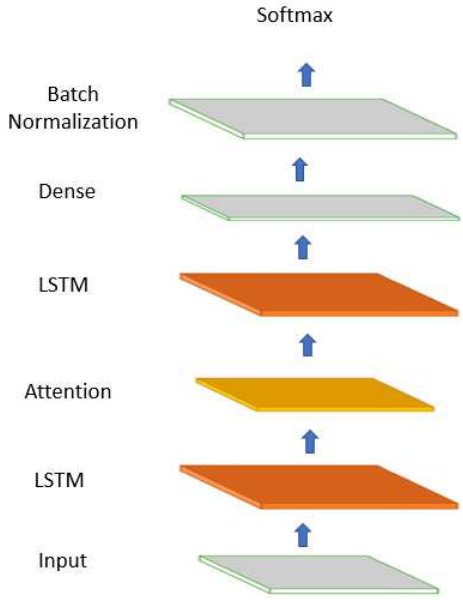

FIGURE 3.7 – *Architecture du modèle proposé avec la couche d'Attention.*

À la fin de la deuxième couche LSTM, nous ajoutons une couche dense pour obtenir deux sorties évaluées qui représentent les classes de prédiction (transaction normale et transaction frauduleuse). Enfin, la couche BatchNormalization est appliquée après la couche dense. La sortie de la couche BatchNormalization est passée dans une couche de classification softmax. À des fins de comparaison, nous présentons également le réseau LSTM sans la couche d'Attention dans la figure 3.8.

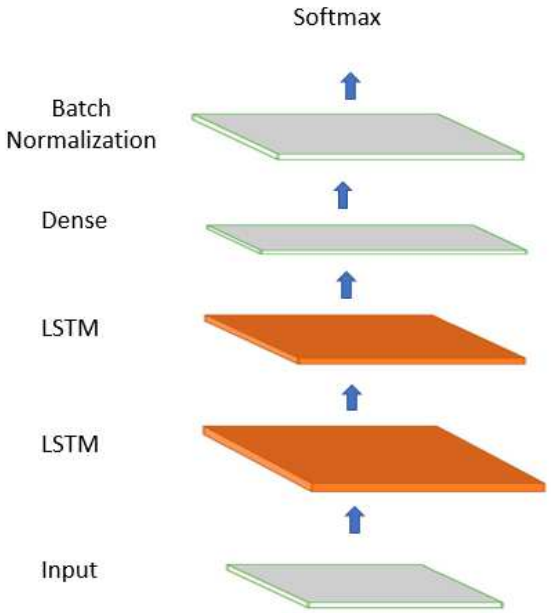

FIGURE 3.8 – *Architecture du modèle proposé sans la couche d'Attention.*

Le processus détaillé du modèle proposé est résumé comme suit : Algorithme 6

Entrée: Historique des transactions collecté jusqu'au moment n : x_1,x_2,...,x_n

Sortie: Prédiction de la fraude entrante

Début:

Répartir le dataset en apprentissage, validation et test.

Transformer le dataset en un tenseur tridimensionnel (N, L, F) où N est le nombre de séquences d'apprentissage, L est la longueur de la séquence et F est le nombre de caractéristiques de chaque séquence.

Définir les paramètres d'apprentissage (taille de la mémoire, taux d'apprentissage, taille du batch et epochs) et définir les variables tensorielles pour les vecteurs de poids et de biais.

Définir la fonction loss cross-entropy et ajouter la fonction d'optimisation Adam pour la minimiser.

Entraîner le réseau ainsi constitué avec les données des cartes de crédit.

Utiliser la sortie de la dernière couche comme prédiction de la prochaine étape temporelle.

Tant que *La convergence optimale n'est pas atteinte*

Calculer l'erreur d'apprentissage.

Calculer l'erreur de validation.

Mettre à jour les poids et les biais en utilisant la rétro-propagation.

Obtenir la prédiction en fournissant des données de test en entrée du réseau entraîné.

Évaluer la précision en comparant les prédictions obtenues avec les données réelles.

Fin

Algorithm 6: Algorithme du modèle de prédiction proposé

4.4 Mesures de performances

Pour évaluer les performances de notre système de détection de fraudes, nous utilisons la matrice de confusion décrite dans le Tableau 3.4.

TABLEAU 3.4 – *Matrice de confusion de classification.*

	positif réel y = 1	négatif réel y = 0
Prédiction positive c = 1	True positive (TP)	False positive (FP)
Prédiction négative c = 0	False negative (FN)	True positive (TN)

À partir de cette matrice, les mesures d'évaluation suivantes sont extraites, à savoir : Accuracy, Sensitivity (or Recall), Specificity et Precision. Ces métriques sont calculées comme suit :

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{3.12}$$

$$Sensitivity = \frac{TP}{TP+FN} \tag{3.13}$$

$$Specificity = \frac{TN}{FP+TN} \tag{3.14}$$

$$Precision = \frac{TP}{TP+FP} \tag{3.15}$$

Lors de l'évaluation des modèles de détection de fraudes, les institutions financières doivent faire face au taux de faux positifs et au taux de faux négatifs enregistrés. Les faux positifs (FP) sont des cas classés par le système de détection de fraudes (SDF) comme des transactions frauduleuses mais qui représentent en réalité des comportements normaux. Ces cas, bien qu'ils aient entraîné des erreurs lors de la classification, ne causent pas de dommages significatifs aux institutions financières.

Les faux négatifs (FN), quant à eux, sont des cas identifiés à tort par le SDF comme des transactions normales mais qui sont en réalité des transactions frauduleuses, ce qui entraîne des coûts importants pour les institutions financières ainsi qu'une diminution de la satisfaction des clients. Par conséquent, nous nous intéresserons davantage à la métrique dite sensitivity (Recall) qui donne une précision sur la classification des cas positifs (fraude), qui représente la métrique d'évaluation la plus appropriée dans ce domaine et que nous utilisons pour mesurer l'efficacité de notre modèle proposé.

4.5 Résultats

Cette étude se base sur des datasets, préalablement traités, de transactions effectuées par carte de crédit, caractérisés par des séquences de transactions temporellement ordonnées, qui permettent au modèle de classification proposé de prédire le label d'une transaction après avoir examiné plusieurs transactions qui la précèdent. Chaque dataset est divisé en trois ensembles. Le premier sous-ensemble de données de 70% est le dataset d'entraînement utilisé pour l'entraînement des modèles, le deuxième sous-ensemble de données de 15% est le dataset de validation utilisé pour valider les classificateurs afin d'éviter le sur-apprentissage et d'améliorer les performances du modèle et le dernier sous-ensemble de données de 15% est utilisé pour tester la généralisation du réseau. Les mêmes datasets d'entraînement et de test des données de cartes de crédit sont choisis pour comparer notre modèle proposé et le modèle LSTM de base.

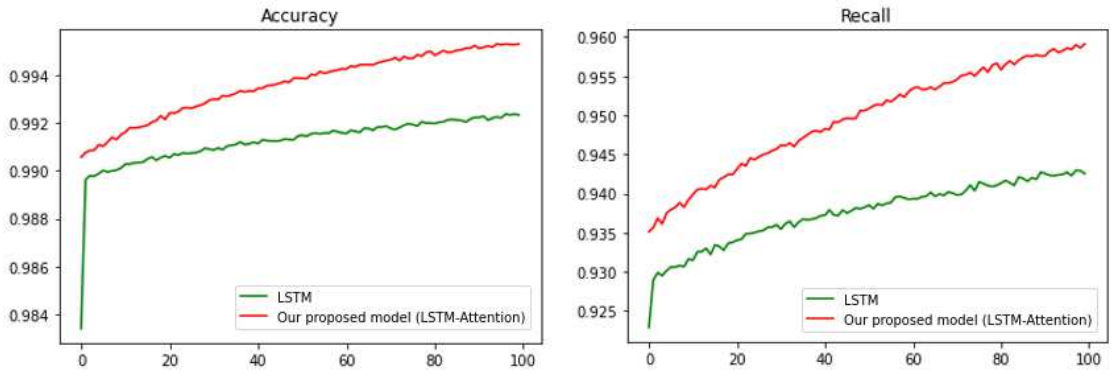

FIGURE 3.9 – *Les graphes d'Accuracy et Recall des modèles comparés*

Les graphiques illustrant les métriques accuracy et recall pour les deux modèles appliqués, par exemple sur notre dataset nommé DataSet-1, sont présentés dans la figure 3.9, à partir de laquelle nous observons que notre modèle (LSTM-Attention) a atteint les taux d'accuracy et de sensitivity (recall) les plus élevés. Cette amélioration significative est due au fait qu'en utilisant les mécanismes d'Attention, des modèles plus pertinents peuvent être extraits des séquences de transactions, ce qui permet au classifieur séquentiel de se concentrer automatiquement sur les éléments de données qui sont les plus importants pour la tâche de classification par une moyenne pondérée basée sur les données de chaque transaction contenue dans la séquence, ce qui entraîne une amélioration des performances de détection.

En outre, pour mettre en évidence les performances de notre modèle proposé, en termes de sensitivity, nous présentons une visualisation de la matrice de confusion réalisée sur chaque modèle, appliqué par exemple sur notre dataset DataSet-1 (Fig 3.10), et à partir de laquelle nous illustrons que notre modèle proposé a une bonne capacité à minimiser le nombre de transactions frauduleuses classées comme normales et à attraper les rares transactions frauduleuses, ce qui est d'une grande importance dans la vie réelle pour les fournisseurs de services financiers.

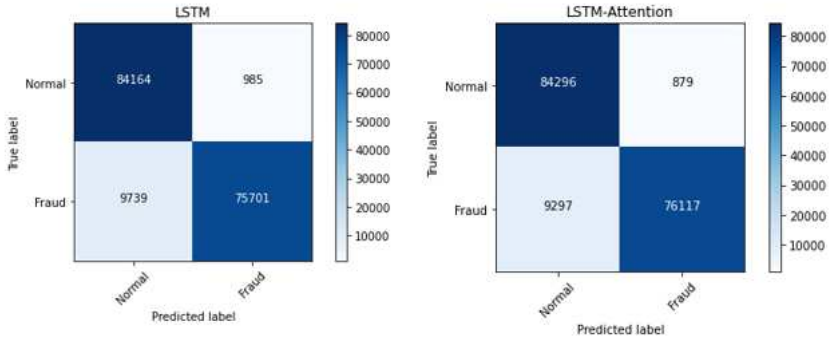

FIGURE 3.10 – *Matrices de confusion du modèle LSTM et de notre modèle proposé*

Aussi, pour valider la pertinence de nos résultats expérimentaux, nous avons comparé notre travail avec les modèles de détection de fraudes récemment utilisés dans la littérature et répertoriés dans le tableau 3.5. La principale raison du choix de ces modèles est qu'ils présentent des performances prometteuses et qu'ils utilisent le même jeu de données DataSet-1 décrit dans ce travail, ce qui rend la comparaison plus pratique et plus fiable.

TABLEAU 3.5 – *Les mesures de performances Accuracy, Recall et Precision.*

Algorithmes	**Accuracy**	**Precision**	**Recall**
GRU (2020) [Forough2021]	-	0.8626	0.7208
LSTM (2020) [Forough2021]	-	0.8575	0.7408
SVM (2021) [Asha2021]	0.9349	0.9743	0.8976
KNN (2021) [Asha2021]	0.9982	0.7142	0.0393
ANN (2021) [Asha2021]	0.9992	0.8115	0.7619
Modèle proposé (LSTM-Attention) appliqué sur Dataset-1	0.9672	**0.9885**	**0.9191**
Modèle proposé (LSTM-Attention) appliqué sur Dataset-2	0.9748	**0.9769**	**0.9422**

Le tableau 3.5 montre les valeurs de performance de chaque modèle utilisé, en termes d'accuracy, precision and sensitivity (recall). Cette dernière métrique est d'une grande importance dans le domaine de la détection de fraudes, puisque les institutions financières sont plus intéressées par la détection des cas de fraudes qui peuvent se produire, ce qui va permettre de protéger les intérêts des consommateurs et de réduire les lourdes pertes financières annuelles causées par la fraude.

Comme nous pouvons le déduire à partir de ces résultats expérimentaux, notre modèle proposé obtient de meilleurs résultats que les méthodes de classification comparées, à savoir GRU, LSTM, SVM, KNN et ANN, ce qui démontre son efficacité pour la tâche de détection de fraudes par carte de crédit.

5 Conclusion

Dans ce chapitre, nous avons cherché à améliorer l'efficacité du classifieur lors de la prédiction des transactions frauduleuses, en combinant la force de différentes techniques d'apprentissage automatique, à savoir : L'approche basée sur la méthode Swarm Intelligence pour sélectionner le sous-ensemble optimal de caractéristiques pertinentes, la méthode UMAP pour réduire la dimensionnalité du dataset, la technique SMOTE pour pallier au problème des données déséquilibrées, le modèle séquentiel LSTM pour modéliser les dépendances à long terme dans les séquences de transactions et les mécanismes d'Attention pour se concentrer spécifiquement sur les informations les plus pertinentes pour la tâche de classification. Ainsi, le modèle que nous proposons est capable de détecter des modèles utiles dans les comportements des consommateurs, ce qui permet de distinguer efficacement les transactions frauduleuses des transactions normales.

Pour valider notre approche, nous avons appliqué notre modèle sur deux datasets différents, et nous avons constaté qu'il était capable de fournir une haute sensibilité (sensitivity) lors de la détection d'instances frauduleuses qui sont d'un grand intérêt dans ce domaine. En outre, en termes de comparaison avec des travaux récents, notre modèle fournit une très bonne performance.

Comme travail futur, nous envisageons d'étudier un nouveau modèle de deep learning pour la détection de fraudes par carte de crédit basé sur l'approche Paragraph Vector-Distributed Memory approach (PV-DM). L'objectif étant d'explorer l'analyse contextuelle offerte par ce modèle et la comparer aux réseaux de neurones réucrrents fréquemment utilisés pour les traitements séquentiels.

4

Analyse contextuelle des fraudes par cartes de crédit basée sur le modèle PV-DM (Paragraph Vector-Distributed Memory)

Dans les travaux précédents, nous avons modélisé les comportements d'achat frauduleux en utilisant les réseaux récurrents LSTM et les mécanismes d'Attention. Les expériences ont montré que la caractérisation d'une transaction entrante, en tenant compte de l'historique des transactions précédentes, améliore les performances de prédiction de façon significative mais nécessite beaucoup de calculs pour fonctionner en raison de leur structure complexe avec des mémoires à court et à long terme pour stocker l'information à différents moments dans le temps, ce qui peut rendre leur utilisation coûteuse en termes de puissance de calcul. De plus, lorsque la séquence de données est trop longue, il peut y avoir une surcharge de mémoire qui empêche l'algorithme de stocker toutes les informations nécessaires et par conséquent entraîner des erreurs dans les prédictions. Nous proposons, dans ce chapitre, d'explorer un nouveau modèle de deep learning pour la définition du comportement d'achat frauduleux en se basant sur l'approche Paragraph Vector-Distributed Memory (PV-DM). Les valeurs expérimentales obtenues révèlent que le modèle PV-DM affiche de bonnes performances et est considéré plus robuste et plus simple que le modèle LSTM couramment utilisé pour le traitement séquentiel des données.

1 Introduction

La détection des fraudes opère dans un environnement dynamique dans la mesure où les utilisateurs changent constamment leurs comportements d'achat tant à l'échelle mondiale que locale et où les fraudeurs abandonnent leurs anciennes stratégies pour en adopter de nouvelles plus efficaces. [Baesens2015] [Akhilomen2013].

Dans les chapitres précédents, nous avons mis en évidence certaines limites inhérentes aux méthodes LSTM et nous avons montré que les réseaux de neurones récurrents utilisés pour modéliser les corrélations séquentielles ne sont pas suffisamment efficaces (Voir chapitre 3). En effet, même si ces approches donnent de bien meilleures performances en raison de leur capacité à gérer la dépendance à long terme, cette amélioration entraîne les inconvénients suivants :

- Les réseaux LSTM ont une structure de réseau plus complexe que les réseaux de neurones traditionnels, avec des mémoires à court et à long terme, des portes de mémoire et des portes d'oubli qui nécessitent des calculs supplémentaires pour mettre à jour les informations stockées dans les mémoires à chaque itération du réseau.
- Les réseaux LSTM doivent représenter l'ensemble de la séquence d'entrée $x_1, x_2, ..., x_n$ sous la forme d'un vecteur unique c, ce qui peut entraîner une perte d'informations puisque toutes les informations de la séquence doivent être compressées en ce vecteur. En outre, ils doivent prédire les instances frauduleuses à partir seulement de ce même vecteur unique c, ce qui constitue une tâche extrêmement complexe et peut causer une perte d'informations surtout lors du traitement de séquences de très grandes tailles.

Pour remédier à ces limitations, nous explorons dans ce chapitre une nouvelle méthode de détection de fraudes par cartes de crédit basée sur l'approche PV-DM. Cette nouvelle technique permet d'obtenir une représentation vectorielle des transactions moyennant le modèle PV-DM (Paragraph Vector-Distributed Memory) [Mikolov2013] en s'inspirant des travaux récents sur l'apprentissage de la représentation vectorielle des paragraphes. L'objectif est la génération d'une représentation numérique à partir des transactions et des séquences, créant ainsi des vecteurs compacts, où les indices contiennent implicitement des informations sur le contexte global de la séquence et les contextes locaux des transactions, ensuite ces vecteurs seront utilisés pour entraîner un modèle de détection de fraudes. Les principales démarches de notre approche sont :

1. Convertir les valeurs catégorielles présentes dans notre dataset en valeurs numériques en appliquant la méthode Label-Encoding comme technique de prétraitement des données.
2. Augmenter la taille de la classe minoritaire du dataset en utilisant la technique d'oversampling SMOTE afin de résoudre le problème du déséquilibre des classes.
3. Identifier le contexte associé au comportement d'achat frauduleux en utilisant le modèle PV-DM comme technique dynamique de modélisation des dépendances existantes entre les transactions bancaires.
4. Mener des expériences sur notre dataset à partir desquelles nous concluons que notre méthode est compétitive et alternative aux architectures LSTM existantes.

Le reste du chapitre est organisé comme suit : la section 2 présente les concepts de base décrivant les méthodes et techniques utilisées dans ce travail, la section 3 décrit en détail le modèle que nous proposons. Les résultats expérimentaux sont présentés et discutés dans la section 4 et enfin, la section 5 conclut le chapitre et suggère des idées pour les recherches futures.

2 Concepts de base

2.1 Encodage des caractéristiques (Feature Encoding)

La plupart des modèles d'apprentissage automatique requièrent une représentation numérique des caractéristiques (features) catégorielles. A cet effet, des opérations de prétraitement sont donc nécessaires pour les approches supervisées et non supervisées en vue de convertir ces variables en valeurs numériques. Pour faire, plusieurs stratégies de Feature Encoding sont présentées dans la littérature. Les deux méthodes couramment utilisées sont l'encodage des labels et l'encodage One-Hot.

Le choix de la technique de Feature Encoding dépendra des données et du modèle utilisé. Il est à noter que l'encodage des caractéristiques peut impacter les performances des modèles d'apprentissage automatique. Il est donc important de tester différentes techniques pour s'assurer de choisir celle qui donne les meilleurs résultats.

2.1.1 Encodage 1-of-K (aussi appelé Encodage One-Hot) :

Les valeurs K des attributs discrets et nominaux représentent des catégories distinctes et non reliées, tels que les différents pays ou les méthodes de paiement. Bien qu'il existe plusieurs façons d'encoder ces variables, le modèle d'encodage One-Hot est particulièrement pratique et couramment utilisé dans la littérature [Bishop2006]. Une variable catégorielle est mappée à un vecteur x à K dimensions, dans lequel un élément x_k est égal à 1 et tous les autres éléments sont égaux à 0. Par exemple, si nous avons une variable avec K = 4 valeurs et que nous voulons coder une observation où la variable prend la valeur x_2, nous représenterons l'observation par $x = (0, 1, 0, 0)$. Ce schéma d'encodage est similaire, mais ne doit pas être confondu, avec les variables dummy utilisées en statistique, où pour une variable catégorielle à K valeurs, nous définissons K-1 variables indicatives pour montrer la présence d'une des K-1 valeurs, la K-ième valeur étant implicitement déterminée lorsque tous les indicateurs sont égaux à zéro.

Cependant, il y a plusieurs inconvénients à utiliser l'encodage One-Hot et qui pourraient impacter les performances du modèle d'apprentissage automatique, à savoir :

- Augmentation de la dimensionnalité : l'encodage One-Hot crée une colonne pour chaque valeur possible de la variable catégorielle, ce qui peut entraîner un nombre important de colonnes pour une variable catégorielle ayant de nombreuses valeurs possibles. Cela peut entraîner des problèmes de mémoire et de temps de calcul pour les modèles d'apprentissage automatique.

- Valeurs manquantes : si certaines observations ne contiennent pas certaines valeurs de la variable catégorielle, l'encodage One-Hot crée quand même une colonne pour ces valeurs, ce qui peut entraîner des valeurs manquantes dans les données et entraîner des résultats imprécis ou des erreurs dans les prédictions

- Redondance des données : l'encodage One-Hot crée des colonnes qui sont souvent corrélées entre elles, car chaque colonne représente une valeur possible d'une variable catégorielle. Ces colonnes sont créées de manière indépendante les unes des autres. De ce fait, les informations contenues dans ces colonnes sont souvent redondantes entre elles, car elles indiquent toutes la même chose : la présence ou l'absence d'une valeur particulière pour une observation. Cette redondance de données peut causer des problèmes pour les modèles d'apprentissage automatique, car ils peuvent sur-estimer l'importance d'une variable par rapport à d'autres, ce qui peut entraîner des résultats imprécis.

- Perte d'informations : l'encodage One-Hot ne conserve pas d'informations sur l'ordre ou les relations entre les différentes valeurs de la variable catégorielle.

2.1.2 Encodage des labels (Label Encoding) :

Est une stratégie d'encodage utilisée pour convertir des variables catégorielles en variables numériques. Le Label Encoding consiste à transformer une caractéristique catégorielle avec n valeurs en une caractéristique numérique prenant n valeurs numériques distinctes. La valeur d'un élément nominal est encodée par sa fréquence relative d'apparition dans les classes d'un jeu d'apprentissage. Si la K-ième valeur d'un élément nominal apparaît n_p fois dans la classe positive et $n - n_p$ fois dans la classe négative, la valeur $x_k = \frac{n_p}{n-n_p}$ est attribuée à cet élément.

Plusieurs avantages à utiliser le Label Encoding par rapport au One-Hot Encoding peuvent être cités :

- Moins de dimensionnalité : le Label Encoding ne crée pas de nouvelles colonnes pour chaque valeur possible de la variable catégorielle, ce qui permet de réduire le nombre de colonnes pour une variable catégorielle ayant de nombreuses valeurs possibles. Cela peut réduire les problèmes de mémoire et de temps de calcul pour les modèles d'apprentissage automatique.

- Préservation de l'ordre : le Label Encoding préserve l'ordre des valeurs de la variable catégorielle, ce qui est important pour les modèles qui utilisent cette information lors des prédictions.

- Interprétabilité : le Label Encoding permet de conserver les valeurs originales de la variable catégorielle, ce qui facilite l'interprétation des résultats.

- Economie de mémoire : le Label Encoding utilise moins de mémoire par rapport au One-Hot Encoding, car il ne nécessite pas de stocker une colonne supplémentaire pour chaque valeur possible de la variable catégorielle.

Par conséquent, nous considérons dans ce travail, cette technique comme une méthode d'ingénierie des caractéristiques (Feature engineering) plutôt que la technique traditionnelle d'encodage One-Hot.

2.2 Paragraph Vector-Distributed Memory (PV-DM)

La méthode PV-DM est une technique d'apprentissage non supervisée qui s'inspire des approches basées sur les réseaux de neurones pour l'apprentissage de la représentation vectorielle des mots (Word embedding) [Mikolov2013] [Le2014]. L'idée principale de PV-DM par rapport à Word2vec [Mikolov2013] est que chaque paragraphe P est représenté en plus par un vecteur unique qui contribue à la prédiction du mot suivant d'une phrase. Cette méthode permet d'améliorer la qualité des représentations des phrases et des mots, ce qui améliore les performances des modèles de traitement de la langue naturelle.

Plus précisément, chaque paragraphe est mappé à un vecteur unique représenté par une colonne de la matrice D, et chaque mot est mappé à un vecteur unique représenté par une colonne de la matrice W (Figure 4.1).

Ensuite, le vecteur de paragraphe et les vecteurs de mots sont moyennés et concaténés par un classificateur tel que Softmax pour prédire le mot suivant en fonction du contexte. Nous obtenons :

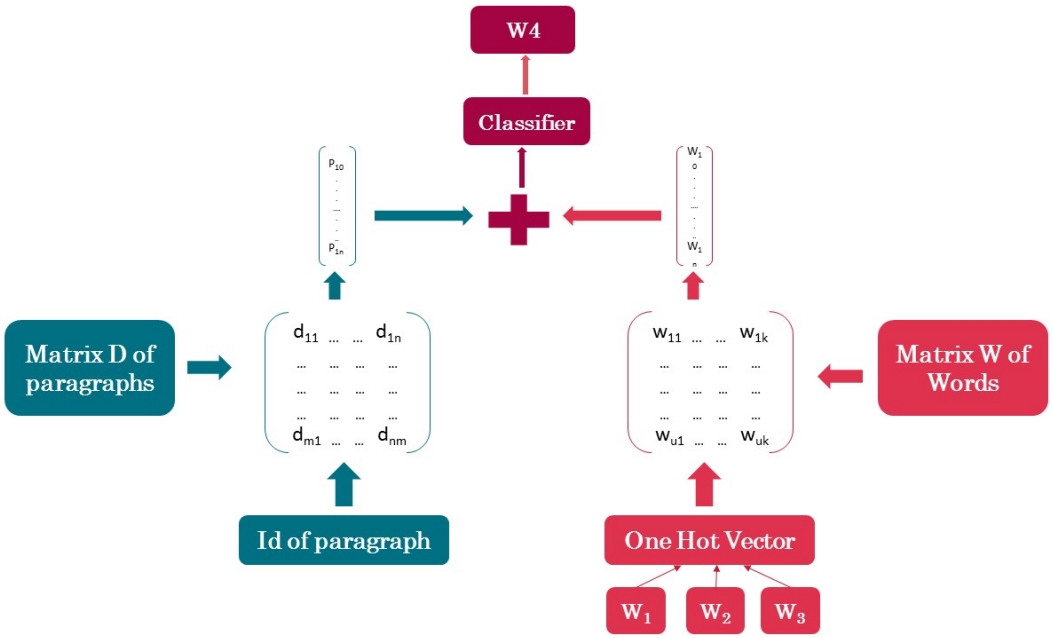

FIGURE 4.1 – *Un framework pour l'apprentissage des vecteurs de paragraphes*

$$p(w_t|w_{t-k},...,w_{t+k};d,W,D) = \frac{e^{y_{wt}}}{\sum_i e^{y_i}} \tag{4.1}$$

$$y = b + Uh(w_t|w_{t-k},...,w_{t+k};d,W,D) \tag{4.2}$$

Où U et b sont les paramètres du classificateur, h est construit à partir de W et D, et d est le vecteur du paragraphe dont sont issus les mots w_{t-k},... ,w_{t+k}.

On peut citer plusieurs avantages à utiliser la méthode PVDM pour la représentation vectorielle :

- Prise en compte des contextes : PVDM prend en compte les contextes dans lesquels se trouvent les mots, ce qui permet de capturer les relations sémantiques entre les mots dans un paragraphe ou une phrase.

- Capacité à traiter des phrases ou des paragraphes complets : PVDM peut traiter des phrases ou des paragraphes complets, ce qui permet de conserver la structure syntaxique des phrases et d'obtenir des informations sémantiques plus riches.

- Meilleure performance pour les tâches de traitement de la langue naturelle : PVDM est particulièrement utile pour les tâches de traitement du langage naturel telles que la classification de textes et la génération de textes.

En résumé, PVDM offre une meilleure compréhension des contextes et des relations sémantiques des mots, il est adapté pour le traitement des phrases et des paragraphes complets et améliore les performances pour les tâches de traitement de la langue naturelle.

3 Approche proposée

3.1 DataSet

Le dataset, qui a été utilisé pour valider l'efficacité de notre approche, a été généré à l'aide du logiciel BankSim, un simulateur de données bancaires élaboré pour une banque en Espagne [Vaughan2020]. Ce dataset est constitué de 594 643 transactions effectuées pendant 180 jours simulés, parmi lesquelles 7200 (≈1,2%) sont considérées comme frauduleuses. Les attributs sont présentés dans le tableau 4.1. Une description détaillée de ce dataset a été fournie dans le chapitre précédent.

TABLEAU 4.1 – *Attributs du dataset*

Nom	**Déscription**
Step	Le jour où la transaction a eu lieu de 1 à 180
Customer ID	Un numéro identifiant le compte client concerné par la transaction
Age Category	Une valeur catégorique classant le client dans un des 8 différents groupes d'âge.
Gender	Une variable catégorique indiquant le sexe du client
Zip Code of account	Le code postal associé au client
Merchant ID	Le numéro identifiant le commerçant impliqué dans la transaction
Zip Code of Merchant	Le code postal du commerçant
Category purchase	Une variable catégorielle indiquant le type de bien ou de service acheté
Amount of purchase	Le montant total de la transaction
Fraud status	Une variable binaire indiquant si la transaction est frauduleuse ou non

3.2 Préparation des données

Durant la phase de pré-traitement des données, nous allons d'abord convertir les données catégorielles en des valeurs numériques adaptées aux algorithmes de machine learning. Pour cela, nous appliquons le Label encoding sur les caractéristiques nommées : **'customer', 'age', 'gender', 'merchant', 'category'** présentes dans notre DataSet.

D'autre part, nous avons expliqué dans les chapitres précédents que notre dataset est fortement déséquilibré puisque le nombre d'instances négatives est supérieur au nombre d'instances positives. En effet, les fraudes représentent (≈1,2%) de l'ensemble des transactions (Figure 4.2). Ce problème affecte la généralisation du modèle et réduit l'efficacité du classificateur à prédire les classes minoritaires, ce qui conduit le modèle à échouer dans la tâche de classification.

Par conséquent, pour améliorer les performances de classification des instances minoritaires de fraude, nous avons utilisé la technique d'oversampling SMOTE [Chawla2002] pour générer des instances d'entraînement synthétiques à partir de la classe minoritaire (Voir chapitre 2).

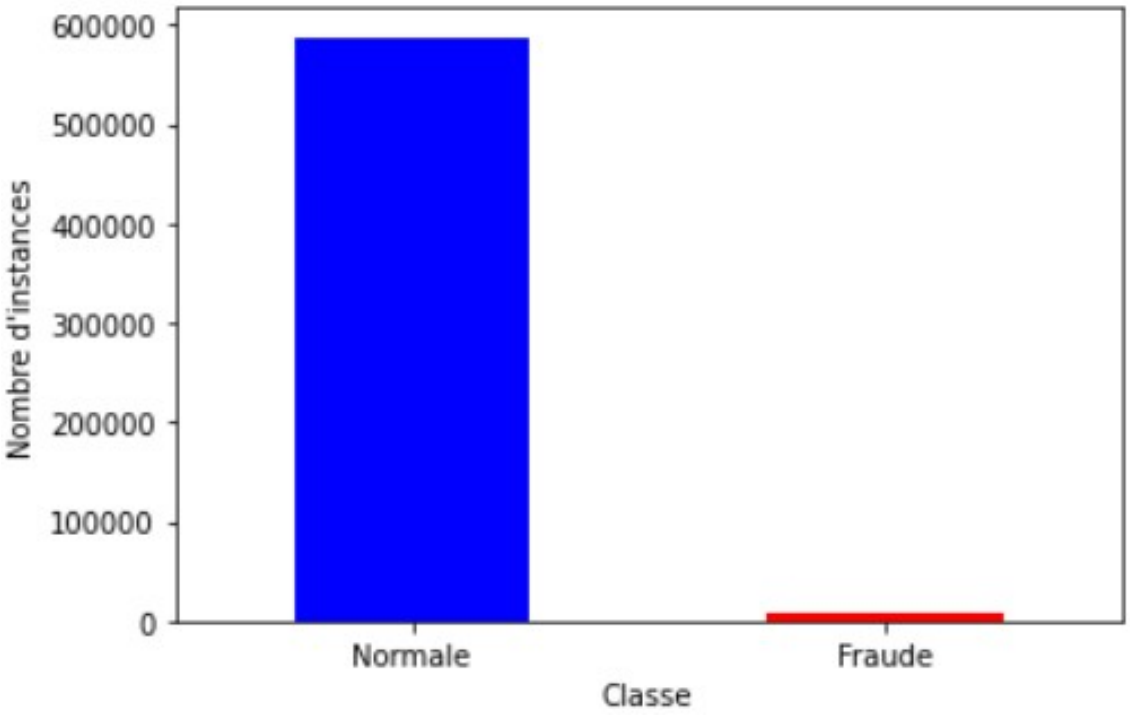

FIGURE 4.2 – *Distribution extrêmement déséquilibrée des classes*

3.3 Mise en oeuvre de l'approche

Une fois la phase de pré-traitement terminée, nous appliquons le modèle PV-DM pour obtenir des vecteurs numériques qui sont ensuite utilisés comme entrées pour les classifieurs de machine learning. L'approche proposée vise à capturer le comportement d'achat historique des détenteurs de cartes de crédit en construisant une représentation compacte du contexte global de la transaction tout en prenant en compte l'ordre et la relation entre les transactions. La Figure 4.3 illustre les phases d'apprentissage et de test relatives à notre approche.

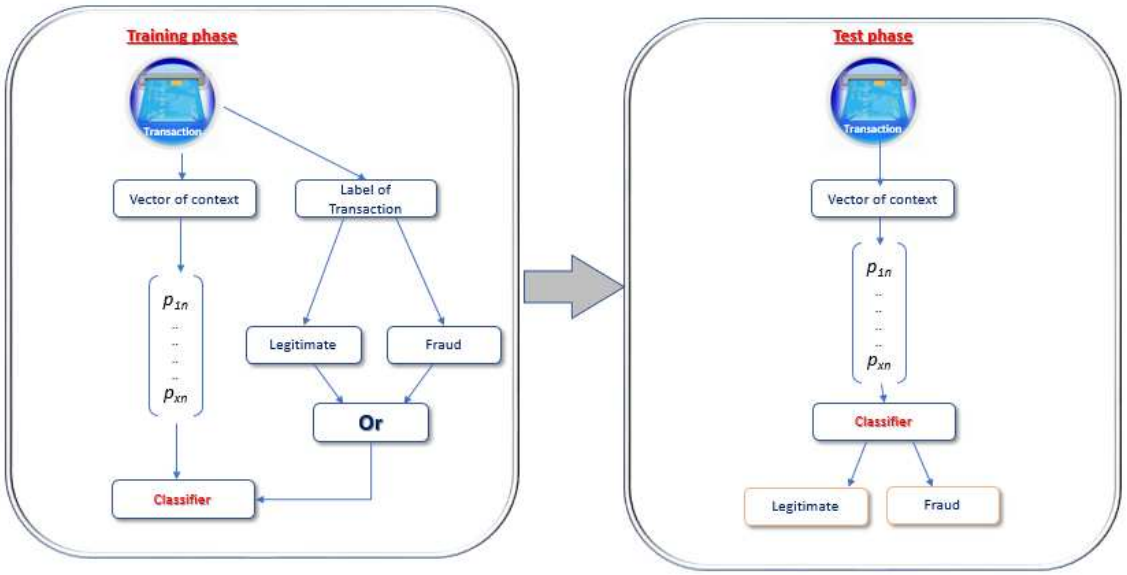

FIGURE 4.3 – *Phases d'entraînement et de test*

Par conséquent, pour entraîner un classifieur, tel que Logistic Regression, les vecteurs de contexte construits par le modèle PV-DM et le label correct de chaque transaction représenteront l'entrée du classifieur, qui générera un modèle pour chaque représentation. Les paramètres du modèle PV-DM sont énumérés dans le tableau suivant 4.2.

TABLEAU 4.2 – *Tableau des paramètres du modèle PV-DM.*

Paramètres du modèle PVDM	**Valeurs**
Nombre de caractéristiques	Toutes,7
min_count	5
window	8
vector_size	100
sample	1e-4
negative	5
workers	4
alpha	0.025
min_alpha	0.025
epochs	100

Au terme de la phase d'apprentissage de chaque classifieur utilisé dans ce travail, notre système de classification binaire devrait être capable de juger si une entrée donnée est un enregistrement normal ou non, en fonction des valeurs de confiance obtenues par chaque représentation.

3.4 Résultats

Notre étude est basée sur le dataset de fraudes par cartes de crédit précité. Ce dataset est divisé en trois ensembles. Le premier sous-ensemble de 70% des données est l'ensemble d'apprentissage utilisé pour entraîner les modèles, le deuxième sous-ensemble de 15% des données est l'ensemble de validation utilisé pour valider la classification et éviter le sur-apprentissage et le dernier sous-ensemble de 15% des données est utilisé pour tester la généralisation du réseau. Les modèles de classification SVM, X_GBoost et Random Forest sont utilisés pour évaluer notre modèle proposé.

TABLEAU 4.3 – *Performances des modèles de classification combinés avec le modèle PV-DM*

Classifieur	**Accuracy**	**Precision**	**Score F_1**	**Sensitivity**
SVM	97.38	96.55	97.40	98.26
X_GBoost	97.59	97.14	97.60	98.05
Random Forest	97.53	96.93	97.55	98.18
LSTM-Attention	97.48	97.69	96.44	94.22

Les résultats de performances des modèles comparés et appliqués à notre dataset sont présentés dans le tableau 4.3, à partir desquelles nous voyons que notre modèle proposé a atteint les taux de précision et de sensibilité (rappel) les plus élevés. Cette amélioration significative est due au fait qu'en utilisant le modèle PV-DM, l'historique du comportement d'achat des détenteurs de cartes de crédit est capturé en construisant une représentation compacte du contexte global de la transaction tout en prenant en compte l'ordre et la relation entre les transactions, ce qui améliore les performances de détection.

Comme nous pouvons le constater à partir de ces résultats expérimentaux, notre approche proposée obtient de meilleurs résultats avec les modèles de classification SVM, X_GBoost et Random Forest, ce qui démontre l'efficacité de notre méthode pour la tâche de détection de fraudes par cartes de crédit.

4 Conclusion

Dans ce chapitre, nous avons cherché à améliorer l'efficacité de la prédiction des transactions frauduleuses, en utilisant le modèle PV-DM (Distributed Memory Model of Paragraph Vectors). Ainsi, le modèle que nous proposons est capable de capturer des modèles de comportements pertinents, ce qui permet de distinguer efficacement les transactions frauduleuses des transactions normales.

Pour valider nos résultats, nous avons appliqué notre méthode à deux différents datasets afin de démontrer sa capacité à fournir une haute sensibilité(Sensitivity) lors de la détection précisément d'instances frauduleuses.

Comme travail futur, nous envisageons d'étudier un nouveau modèle nommé Transformer, basé uniquement sur les mécanismes d'Attention, qui pourrait à termes, prendre une place de plus en plus importante pour le traitement séquentiel en remplacement des réseaux de neurones récurrents LSTM.

Conclusion générale

Conclusion

La fraude par cartes de crédit est depuis longtemps un problème critique qui a de lourdes conséquences sur le secteur financier. Au fil des années, la fraude a été au centre des travaux de recherches scientifiques dans ce secteur, qui a investi des montants importants et a fait appel à des experts tels que les data scientists ou les ingénieurs financiers, en vue de concevoir et déployer des dispositifs de détection de fraudes qui soient plus performants que les techniques traditionnelles.

Pourtant, de nombreuses enquêtes menées par de grands cabinets d'études tels que Forrester et Gartner ont indiqué que la tendance à la hausse des cas de fraudes se poursuit. Il est donc évident que les méthodes de détection de fraudes doivent être renforcées.

En outre, l'avènement de techniques d'Intelligence Artificielle extrêmement puissantes a donné lieu à une multitude d'opportunités pour les industries afin de faire évoluer leurs méthodes de détection de fraudes. Grâce à ces techniques, les entreprises ont pu développer des solutions innovantes qui exploitent la puissance des données et détectent les fraudes avec une grande précision et une grande rapidité, chose que les approches classiques basées sur les connaissances des experts sont incapables de garantir.

Bien que la détection de fraudes axée sur les données offre de nombreuses opportunités, certains défis complexes nécessitent encore des analyses approfondies. Dans cette thèse, nous avons d'abord introduit de manière exhaustive les défis que revêt la détection de fraudes et les difficultés inhérentes à ce domaine, principalement :

1. Le déséquilibre des DataSets
2. L'utilisation des données historiques pour la déinition du contexte d'achat
3. L'éxtraction des données pertinentes pour la classification des fraudes bancaires
4. L'évaluation des performances

Au regard de ces défis, nous avons analysé les travaux connexes et donné un aperçu sur les diverses approches qui ont été proposées dans la littérature.

La première contribution de cette thèse est une approche originale qui vise à résoudre le problème du déséquilibre fort des classes. En utilisant la méthode de clustering k-Means et les opérateurs génétiques, nous avons conçu une méthode d'oversampling afin d'augmenter la taille de la classe minoritaire. La méthode k-Means a été utilisée d'abord pour répartir, dans des clusters distincts, les observations de la classe minoritaire en fonction de leurs similitudes. Ensuite, au moyen d'opérateurs génétiques de croisement et de mutation, nous avons généré au niveau de chaque cluster de nouvelles observations synthétiques appartenant à la classe minoritaire et qui imitent le plus fidèlement possible les observations initiales. Nous avons fusionné ces nouvelles observations avec le dataset initial pour obtenir un jeu d'apprentissage augmenté. Les résultats de nos expérimentations ont prouvé qu'un classifieur entraîné sur ce dataset équilibré est plus performant que

le même classifieur entraîné sur des données initiales non équilibrées, surtout en ce qui concerne la sensibilité (sensitivity), ce qui résulte en un dispositif efficace de détection de fraudes.

Notre deuxième contribution consiste à modéliser le contexte qui définit le comportement d'achat légitime ou frauduleux des consommateurs, à partir de données historiques. En se basant sur l'architecture récurrente des réseaux LSTM, nous avons exploité les informations séquentielles afin de construire automatiquement ce contexte. Le modèle proposé présente une alternative intéressante à l'agrégation manuelle des caractéristiques (Feature engineering) puisqu'il est capable d'intégrer des informations temporelles importantes qui permettent d'analyser des événements complexes et de découvrir des menaces potentielles ou explicites d'activités frauduleuses. Les résultats obtenus montrent que la capture séquentielle des fraudes ajoute une précision significative par rapport à un classificateur sans mémoire.

L'architecture LSTM est adaptée aux modèles de succession séquentielle de points de données où l'occurrence d'un événement peut dépendre, plus loin dans le temps, de la présence de plusieurs autres événements. Cependant, nous avons démontré que la représentation de l'ensemble de la séquence d'entrée sous la forme d'un vecteur unique peut entraîner une perte d'informations puisque toutes les informations de la séquence doivent être compressées en ce vecteur, ce qui constitue une tâche extrêmement complexe surtout lors du traitement de séquences de très grandes tailles.

Pour dépasser cette limite, nous proposons dans notre troisième contribution, d'améliorer les performances du système de détection de fraudes en utilisant les mécanismes d'Attention, capable de se focaliser sur les informations les plus pertinentes pour la tâche de classification. Le modèle proposé, comparé aux études précédentes, tient compte de la nature séquentielle des données transactionnelles et permet au classificateur d'identifier les transactions les plus importantes dans la séquence d'entrée et qui prédisent avec une plus grande précision les transactions frauduleuses. Les performances de notre modèle présentent de bons résultats en termes d'efficience et d'efficacité.

Notre quatrième contribution consiste à explorer un nouveau modèle de deep learning pour la définition du comportement d'achat frauduleux en se basant sur l'approche Paragraph Vector-Distributed Memory (PV-DM). L'objectif de ce modèle est la génération d'une représentation numérique à partir des transactions et des séquences, créant ainsi des vecteurs compacts, où les indices contiennent implicitement des informations sur le contexte global de la séquence et les contextes locaux des transactions. Les valeurs expérimentales obtenues révèlent que le modèle PV-DM affiche de bonnes performances et est considéré plus robuste et plus simple que le modèle LSTM couramment utilisé pour le traitement séquentiel des données.

Les travaux réalisés au cours de cette thèse constituent un bon point de départ vers la prédiction efficace des fraudes par cartes de crédit. Ces travaux ont permis de surmonter certaines limites abordées dans l'état de l'art. Dans cette section, nous allons discuter de quelques améliorations possibles qui peuvent être apportées à un système de détection de fraudes dans le futur :

1. **Généralisation du modèle**
   Nous avons validé la conception de notre système de détection de fraudes en utilisant les données disponibles sur Kaggle et celles fournies par BankSim. A terme, il s'agira de mener des expérimentations sur une base de données à grande échelle issues de transactions bancaires réelles afin de consolider les résultats de nos algorithmes et de notre approche. L'exploration des résultats obtenus à partir de ce dataset pourrait constituer une étude intéressante sur la progression de la fraude et la qualité des mesures de performance utilisées.

2. **Utilisation de nouveaux modèles de prédiction**
   Les modèles LSTM, PV-DM et les mécanismes d'Attention ont été testés pour identifier les transactions frauduleuses émises illégitimement au nom du titulaire légitime de la carte. Dans l'étape suivante, nous envisageons de mettre en œuvre d'autres algorithmes de prédiction capables de traiter et de modéliser des données temporelles. Le choix du modèle à retenir repose sur plusieurs critères tels que les données utilisées pour construire et entraîner ces modèles et le choix des résultats à atteindre. A titre d'exemple, nous avons utilisé les réseaux LSTMs pour modéliser la séquence temporelle des transactions par cartes de crédit. Cependant, les informations séquentielles peuvent également être modélisées par d'autres modèles tels que les GRUs ou les CNNs, etc. Nous envisageons donc d'utiliser d'autres réseaux de neurones et de proposer d'autres variantes pour notre architecture. En outre, le paramétrage des algorithmes d'apprentissage étant crucial, il conviendra de faire appel à de nouvelles techniques pour optimiser le choix de ces paramètres.

3. **Validation du résultat par un expert financier**
   Le système de détection de fraudes fournit les décisions (prédictions) qui devraient être validées par un expert financier. Les informations fournies lui permettront de confirmer ou d'infirmer le résultat de la décision. De ce fait, l'association entre la performance du modèle et le savoir-faire des experts métiers permettra d'obtenir des résultats d'analyse qui seront pertinents. Il sera par ailleurs intéressant de valider notre approche en incluant une analyse fonctionnelle afin de vérifier par exemple la cohérence des règles.

4. **Interprétation des réusltats obtenus par les modèles de Deep Learning**
   L'interprétation des résultats obtenus par les modèles de Deep Learning permettra de comprendre la logique du modèle et, dans de nombreux cas, elle va améliorer la capacité du modèle à généraliser les connaissances acquises. De plus, ces connaissances peuvent se présenter sous la forme de règles très spécifiques qui fournissent une prédiction simple à interpréter, et que les experts financiers peuvent intégrer dans un système de détection de fraudes éxistant.

Bibliographie

[Abdallah2016] Aisha Abdallah, Mohd Aizaini Maarof et Anazida Zainal. *Fraud detection system : A survey*. Journal of Network and Computer Applications, vol. 68, pages 90–113, 2016.

[ACFE2018] ACFE. *Report to the Nations 2018 Global Study on Occupational Fraud and Abuse*. 2018.

[Akhilomen2013] John Akhilomen. *Data Mining Application for Cyber Credit-Card Fraud Detection System*. volume Vol. 7987, pages 218–228, 07 2013.

[Aleskerov1997] Emin Aleskerov, Bernd Freisleben et Bharat Rao. *Cardwatch : A neural network based database mining system for credit card fraud detection*. In Proceedings of the IEEE/IAFE 1997 computational intelligence for financial engineering (CIFEr), pages 220–226. IEEE, 1997.

[Altman1992] Naomi S Altman. *An introduction to kernel and nearest-neighbor nonparametric regression*. The American Statistician, vol. 46, no. 3, pages 175–185, 1992.

[Asha2021] RB Asha et Suresh Kumar KR. *Credit card fraud detection using artificial neural network*. Global Transitions Proceedings, vol. 2, no. 1, pages 35–41, 2021.

[Baesens2015] Bart Baesens, Véronique Van Vlasselaer et Wouter Verbeke. Fraud analytics using descriptive, predictive, and social network techniques : A guide to data science for fraud detection. 08 2015.

[Bahdanau2014] Dzmitry Bahdanau, Kyunghyun Cho et Yoshua Bengio. *Neural machine translation by jointly learning to align and translate*. arXiv preprint arXiv :1409.0473, 2014.

[Bahnsen2013] Alejandro Correa Bahnsen, Aleksandar Stojanovic, Djamila Aouada et Björn Ottersten. *Cost sensitive credit card fraud detection using Bayes minimum risk*. In 2013 12th international conference on machine learning and applications, volume 1, pages 333–338. IEEE, 2013.

[Bahnsen2015] Alejandro Correa Bahnsen, Djamila Aouada, Aleksandar Stojanovic et Björn Ottersten. *Detecting credit card fraud using periodic features*. In 2015 IEEE 14th international conference on machine learning and applications (ICMLA), pages 208–213. IEEE, 2015.

[Bahnsen2016] Alejandro Correa Bahnsen, Djamila Aouada, Aleksandar Stojanovic et Björn Ottersten. *Feature engineering strategies for credit card fraud detection*. Expert Systems with Applications, vol. 51, pages 134–142, 2016.

[Basheer2000] Imad A Basheer et Maha Hajmeer. *Artificial neural networks : fundamentals, computing, design, and application*. Journal of microbiological methods, vol. 43, no. 1, pages 3–31, 2000.

[Batista2000] Gustavo EAPA Batista, Andre CPLF Carvalho et Maria Carolina Monard. *Applying one-sided selection to unbalanced datasets*. In Mexican International Conference on Artificial Intelligence, pages 315–325. Springer, 2000.

[Batista2003] Gustavo EAPA Batista, Ana LC Bazzan, Maria Carolina Monard*et al.* *Balancing Training Data for Automated Annotation of Keywords : a Case Study.* In WOB, pages 10–18, 2003.

[Batista2004] Gustavo EAPA Batista, Ronaldo C Prati et Maria Carolina Monard. *A study of the behavior of several methods for balancing machine learning training data*. ACM SIGKDD explorations newsletter, vol. 6, no. 1, pages 20–29, 2004.

[Bauer1999] Eric Bauer et Ron Kohavi. *An empirical comparison of voting classification algorithms : Bagging, boosting, and variants*. Machine learning, vol. 36, no. 1, pages 105–139, 1999.

[Bayer2015] Justin Simon Bayer. *Learning sequence representations*. Thèse de Doctorat, Technische Universität München, 2015.

[Becht2019] Etienne Becht, Leland McInnes, John Healy, Charles-Antoine Dutertre, Immanuel WH Kwok, Lai Guan Ng, Florent Ginhoux et Evan W Newell. *Dimensionality reduction for visualizing single-cell data using UMAP*. Nature biotechnology, vol. 37, no. 1, pages 38–44, 2019.

[Benchaji2019] Ibtissam Benchaji, Samira Douzi et Bouabid El Ouahidi. *Novel learning strategy based on genetic programming for credit card fraud detection in Big Data*. In Proceedings of international conference Big Data analytics, data mining and computational intelligence, pages 3–10, 2019.

[Benchaji2021a] Ibtissam Benchaji, Samira Douzi et Bouabid El Ouahidi. *Credit card fraud detection model based on LSTM recurrent neural networks*. Journal of Advances in Information Technology Vol, vol. 12, no. 2, 2021.

[Benchaji2021b] Ibtissam Benchaji, Samira Douzi, Bouabid El Ouahidi et Jaafar Jaafari. *Enhanced credit card fraud detection based on attention mechanism and LSTM deep model*. Journal of Big Data, vol. 8, pages 1–21, 2021.

[Bengio1994] Yoshua Bengio, Patrice Simard et Paolo Frasconi. *Learning long-term dependencies with gradient descent is difficult*. IEEE transactions on neural networks, vol. 5, no. 2, pages 157–166, 1994.

[Bentley2000] Peter J Bentley, Jung-Won Kim, Gil-Ho Jung et Jong-Uk Choi. *Fuzzy darwinian detection of credit card fraud*. In Proceedings of the Korea Information Processing Society Conference, pages 277–280. Korea Information Processing Society, 2000.

[Bhattacharyya2011] Siddhartha Bhattacharyya, Sanjeev Jha, Kurian Tharakunnel et J Christopher Westland. *Data mining for credit card fraud : A comparative study*. Decision support systems, vol. 50, no. 3, pages 602–613, 2011.

[Bhowan2012] Urvesh Bhowan, Mark Johnston, Mengjie Zhang et Xin Yao. *Evolving diverse ensembles using genetic programming for classification with unbalanced data*. IEEE Transactions on Evolutionary Computation, vol. 17, no. 3, pages 368–386, 2012.

[Bishop2006] Christopher M Bishop et Nasser M Nasrabadi. Pattern recognition and machine learning, volume 4. Springer, 2006.

[Bolton2002] Richard J Bolton et David J Hand. *Statistical fraud detection : A review*. Statistical science, vol. 17, no. 3, pages 235–255, 2002.

[Breiman1996a] Leo Breiman. *Bagging predictors*. Machine learning, vol. 24, no. 2, pages 123–140, 1996.

[Breiman1996b] Leo Breiman. *Bias, variance, and arcing classifiers*. Rapport technique, Tech. Rep. 460, Statistics Department, University of California, Berkeley ..., 1996.

[Brezočnik2018] Lucija Brezočnik, Iztok Fister Jr et Vili Podgorelec. *Swarm intelligence algorithms for feature selection : a review*. Applied Sciences, vol. 8, no. 9, page 1521, 2018.

[Carcillo2018] Fabrizio Carcillo, Yann-Aël Le Borgne, Olivier Caelen et Gianluca Bontempi. *Streaming active learning strategies for real-life credit card fraud detection : assessment and visualization*. International Journal of Data Science and Analytics, vol. 5, no. 4, pages 285–300, 2018.

[Carcillo2021] Fabrizio Carcillo, Yann-Aël Le Borgne, Olivier Caelen, Yacine Kessaci, Frédéric Oblé et Gianluca Bontempi. *Combining unsupervised and supervised learning in credit card fraud detection*. Information sciences, vol. 557, pages 317–331, 2021.

[Celebi2011] M Emre Celebi. *Improving the performance of k-means for color quantization*. Image and Vision Computing, vol. 29, no. 4, pages 260–271, 2011.

[Chandola2010] Varun Chandola, Arindam Banerjee et Vipin Kumar. *Anomaly detection for discrete sequences : A survey*. IEEE transactions on knowledge and data engineering, vol. 24, no. 5, pages 823–839, 2010.

[Chawla2002] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall et W Philip Kegelmeyer. *SMOTE : synthetic minority over-sampling technique*. Journal of artificial intelligence research, vol. 16, pages 321–357, 2002.

[Chawla2004] Nitesh V Chawla, Nathalie Japkowicz et Aleksander Kotcz. *Special issue on learning from imbalanced data sets*. ACM SIGKDD explorations newsletter, vol. 6, no. 1, pages 1–6, 2004.

[Chen2021] Jiahui Chen, Rundong Zhao, Yiying Tong et Guo-Wei Wei. *Evolutionary de rham-hodge method*. Discrete and continuous dynamical systems. Series B, vol. 26, no. 7, page 3785, 2021.

[Chorowski2015] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho et Yoshua Bengio. *Attention-based models for speech recognition*. Advances in neural information processing systems, vol. 28, 2015.

[Cieslak2012] David A Cieslak, T Ryan Hoens, Nitesh V Chawla et W Philip Kegelmeyer. *Hellinger distance decision trees are robust and skew-insensitive*. Data Mining and Knowledge Discovery, vol. 24, no. 1, pages 136–158, 2012.

[Dal Pozzolo2014a] Andrea Dal Pozzolo, Olivier Caelen, Yann-Ael Le Borgne, Serge Waterschoot et Gianluca Bontempi. *Learned lessons in credit card fraud detection from a practitioner perspective*. Expert systems with applications, vol. 41, no. 10, pages 4915–4928, 2014.

[Dal Pozzolo2014b] Andrea Dal Pozzolo, Reid Johnson, Olivier Caelen, Serge Waterschoot, Nitesh V Chawla et Gianluca Bontempi. *Using HDDT to avoid instances propagation in unbalanced and evolving data streams*. In 2014 International Joint Conference on Neural Networks (IJCNN), pages 588–594. IEEE, 2014.

[Dal Pozzolo2015] Andrea Dal Pozzolo, Olivier Caelen, Reid A Johnson et Gianluca Bontempi. *Calibrating probability with undersampling for unbalanced classification*. In 2015 IEEE symposium series on computational intelligence, pages 159–166. IEEE, 2015.

[Dal Pozzolo2017] Andrea Dal Pozzolo, Giacomo Boracchi, Olivier Caelen, Cesare Alippi et Gianluca Bontempi. *Credit card fraud detection : a realistic modeling and a novel learning strategy*. IEEE transactions on neural networks and learning systems, vol. 29, no. 8, pages 3784–3797, 2017.

[Delamaire2009] Linda Delamaire, Hussein Abdou et John Pointon. *Credit card fraud and detection techniques : a review*. Banks and Bank systems, vol. 4, no. 2, pages 57–68, 2009.

[Dhankhad2018] Sahil Dhankhad, Emad Mohammed et Behrouz Far. *Supervised machine learning algorithms for credit card fraudulent transaction detection : a comparative study*. In 2018 IEEE international conference on information reuse and integration (IRI), pages 122–125. IEEE, 2018.

[Dhok2012] Shailesh S Dhok et GR Bamnote. *Credit card fraud detection using hidden Markov model*. International Journal of Soft Computing and Engineering (IJSCE), vol. 2, no. 1, pages 231–237, 2012.

[Domingos2000] Pedro Domingos. *A unified bias-variance decomposition for zero-one and squared loss*. AAAI/IAAI, vol. 2000, pages 564–569, 2000.

[Donato1999] June M Donato, Jack C Schryver, Gregory C Hinkel, Richard L Schmoyer Jr, Michael R Leuze et Nancy W Grandy. *Mining multi-dimensional data for decision support*. Future generation computer systems, vol. 15, no. 3, pages 433–441, 1999.

[Dorronsoro1997] Jose R Dorronsoro, Francisco Ginel, C Sgnchez et Carlos S Cruz. *Neural fraud detection in credit card operations*. IEEE transactions on neural networks, vol. 8, no. 4, pages 827–834, 1997.

[Drummond2003] Chris Drummond, Robert C Holte*et al. C4. 5, class imbalance, and cost sensitivity : why under-sampling beats over-sampling*. In Workshop on learning from imbalanced datasets II, volume 11, pages 1–8. Citeseer, 2003.

[Elman1990] Jeffrey L Elman. *Finding structure in time*. Cognitive science, vol. 14, no. 2, pages 179–211, 1990.

[Eshelman1991] Larry J Eshelman. *The CHC adaptive search algorithm : How to have safe search when engaging in nontraditional genetic recombination*. In Foundations of genetic algorithms, volume 1, pages 265–283. Elsevier, 1991.

[Estabrooks2004] Andrew Estabrooks, Taeho Jo et Nathalie Japkowicz. *A multiple resampling method for learning from imbalanced data sets*. Computational intelligence, vol. 20, no. 1, pages 18–36, 2004.

[Fernández2018] Alberto Fernández, Salvador Garcia, Francisco Herrera et Nitesh V Chawla. *SMOTE for learning from imbalanced data : progress and challenges, marking the 15-year anniversary*. Journal of artificial intelligence research, vol. 61, pages 863–905, 2018.

[Forough2021] Javad Forough et Saeedeh Momtazi. *Ensemble of deep sequential models for credit card fraud detection*. Applied Soft Computing, vol. 99, page 106883, 2021.

[Freund1999] Yoav Freund, Robert Schapire et Naoki Abe. *A short introduction to boosting*. Journal-Japanese Society For Artificial Intelligence, vol. 14, no. 771-780, page 1612, 1999.

[Friedman2002] Jerome H Friedman. *Stochastic gradient boosting*. Computational statistics & data analysis, vol. 38, no. 4, pages 367–378, 2002.

[Ganji2012] Venkata Ratnam Ganji et S Naga Prasad Mannem. *Credit card fraud detection using anti-k nearest neighbor algorithm*. International Journal on Computer Science and Engineering, vol. 4, no. 6, pages 1035–1039, 2012.

[Ghattas2000] Badih Ghattas. *Agrégation d'arbres de classification*. Revue de statistique appliquee, vol. 48, no. 2, pages 85–98, 2000.

[Ghosh1994] Sushmito Ghosh et Douglas L Reilly. *Credit card fraud detection with a neural-network*. In System Sciences, 1994. Proceedings of the Twenty-Seventh Hawaii International Conference on, volume 3, pages 621–630. IEEE, 1994.

[Goldberg1994] David E Goldberg. *Genetic and evolutionary algorithms come of age*. Communications of the ACM, vol. 37, no. 3, pages 113–120, 1994.

[Gore2016] Shounak Gore et Venu Govindaraju. *Feature selection using cooperative game theory and relief algorithm*. In Knowledge, information and creativity support systems : recent trends, advances and solutions, pages 401–412. Springer, 2016.

[Goughlin2018] Tom Goughlin. *175 zettabytes by 2025*. 2018.

[Graves2012] Alex Graves. *Supervised sequence labelling*. In Supervised sequence labelling with recurrent neural networks, pages 5–13. Springer, 2012.

[Graves2014] Alex Graves et Navdeep Jaitly. *Towards end-to-end speech recognition with recurrent neural networks*. In International conference on machine learning, pages 1764–1772. PMLR, 2014.

[Han2005] Hui Han, Wen-Yuan Wang et Bing-Huan Mao. *Borderline-SMOTE : a new over-sampling method in imbalanced data sets learning*. In International conference on intelligent computing, pages 878–887. Springer, 2005.

[Han2006] Jiawei Han et Micheline Kamber. *Data mining : concepts and techniques, 2nd*. University of Illinois at Urbana Champaign : Morgan Kaufmann, 2006.

[Haraty2015] Ramzi A Haraty, Mohamad Dimishkieh et Mehedi Masud. *An enhanced k-means clustering algorithm for pattern discovery in healthcare data*. International Journal of distributed sensor networks, vol. 11, no. 6, page 615740, 2015.

[He2008] Haibo He, Yang Bai, Edwardo A Garcia et Shutao Li. *ADASYN : Adaptive synthetic sampling approach for imbalanced learning*. In 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence), pages 1322–1328. IEEE, 2008.

[He2009] Haibo He et Edwardo A Garcia. *Learning from imbalanced data*. IEEE Transactions on knowledge and data engineering, vol. 21, no. 9, pages 1263–1284, 2009.

[Hicks2019] David Hicks, Judd Caplain, Natalie Faulkner, Enric Olcina, Thomas Stanton et Lem Chin Kok. *Global banking fraud survey the multi-faceted threat of fraud : Are banks up to the challenge ?* KPMG, 2019.

[Hlosta2013] Martin Hlosta, Rostislav Stríz, Jan Kupcík, Jaroslav Zendulka et Tomás Hruska. *Constrained classification of large imbalanced data by logistic regression and genetic algorithm*. International Journal of Machine Learning and Computing, vol. 3, no. 2, page 214, 2013.

[Hochreiter1997] Sepp Hochreiter et Jürgen Schmidhuber. *Long short-term memory*. Neural computation, vol. 9, no. 8, pages 1735–1780, 1997.

[Holland1992] John H Holland. Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence. MIT press, 1992.

[Hormozi2013] Hadi Hormozi, Mohammad Kazem Akbari, Elham Hormozi et Morteza Sargolzaei Javan. *Credit cards fraud detection by negative selection algorithm on hadoop (To reduce the training time)*. In The 5th Conference on Information and Knowledge Technology, pages 40–43. IEEE, 2013.

[Hüe1997] Xavier Hüe. *Genetic algorithms for optimization*. Rapport technique, Citeseer, 1997.

[Jacobson2013] Ralph Jacobson. *2.5 quintillion bytes of data created every day. How does CPG & Retail manage it ?* 2013.

[Japkowicz2002] Nathalie Japkowicz et Shaju Stephen. *The class imbalance problem : A systematic study*. Intelligent data analysis, vol. 6, no. 5, pages 429–449, 2002.

[Jiang2018] Changjun Jiang, Jiahui Song, Guanjun Liu, Lutao Zheng et Wenjing Luan. *Credit card fraud detection : A novel approach using aggregation strategy and feedback mechanism*. IEEE Internet of Things Journal, vol. 5, no. 5, pages 3637–3647, 2018.

[Jolliffe2016] Ian T Jolliffe et Jorge Cadima. *Principal component analysis : a review and recent developments*. Philosophical Transactions of the Royal Society A : Mathematical, Physical and Engineering Sciences, vol. 374, no. 2065, page 20150202, 2016.

[Juang2010] Li-Hong Juang et Ming-Ni Wu. *MRI brain lesion image detection based on color-converted K-means clustering segmentation*. Measurement, vol. 43, no. 7, pages 941–949, 2010.

[Jurgovsky2018] Johannes Jurgovsky, Michael Granitzer, Konstantin Ziegler, Sylvie Calabretto, Pierre-Edouard Portier, Liyun He-Guelton et Olivier Caelen. *Sequence classification for credit-card fraud detection*. Expert Systems with Applications, vol. 100, pages 234–245, 2018.

[Kamaruddin2016] Sk Kamaruddin et Vadlamani Ravi. *Credit card fraud detection using big data analytics : use of PSOAANN based one-class classification*. In Proceedings of the international conference on informatics and analytics, pages 1–8, 2016.

[Kanungo2002] Tapas Kanungo, David M Mount, Nathan S Netanyahu, Christine D Piatko, Ruth Silverman et Angela Y Wu. *An efficient k-means clustering algorithm : Analysis and implementation*. IEEE transactions on pattern analysis and machine intelligence, vol. 24, no. 7, pages 881–892, 2002.

[Krivko2010] Maria Krivko. *A hybrid model for plastic card fraud detection systems*. Expert Systems with Applications, vol. 37, no. 8, pages 6070–6076, 2010.

[Kültür2017] Yiğit Kültür et Mehmet Ufuk Çağlayan. *Hybrid approaches for detecting credit card fraud*. Expert Systems, vol. 34, no. 2, page e12191, 2017.

[Kumari2019] Priyanka Kumari et Smita Prava Mishra. *Analysis of credit card fraud detection using fusion classifiers*. In Computational Intelligence in Data Mining, pages 111–122. Springer, 2019.

[Kuo2002] RJ Kuo, LM Ho et Clark M Hu. *Integration of self-organizing feature map and K-means algorithm for market segmentation*. Computers & Operations Research, vol. 29, no. 11, pages 1475–1493, 2002.

[Laleh2009] Naeimeh Laleh et Mohammad Abdollahi Azgomi. *A taxonomy of frauds and fraud detection techniques*. In International Conference on Information Systems, Technology and Management, pages 256–267. Springer, 2009.

[Le2014] Quoc Le et Tomas Mikolov. *Distributed representations of sentences and documents*. In International conference on machine learning, pages 1188–1196. PMLR, 2014.

[Linderman2019] George C Linderman, Manas Rachh, Jeremy G Hoskins, Stefan Steinerberger et Yuval Kluger. *Fast interpolation-based t-SNE for improved visualization of single-cell RNA-seq data*. Nature methods, vol. 16, no. 3, pages 243–245, 2019.

[Liu2008] Fei Tony Liu, Kai Ming Ting et Zhi-Hua Zhou. *Isolation forest*. In 2008 eighth ieee international conference on data mining, pages 413–422. IEEE, 2008.

[Lopez-Rojas2016] Edgar Alonso Lopez-Rojas et Stefan Axelsson. *A review of computer simulation for fraud detection research in financial datasets*. In 2016 Future technologies conference (FTC), pages 932–935. IEEE, 2016.

[MacQueen1967] J MacQueen. *Classification and analysis of multivariate observations*. In 5th Berkeley Symp. Math. Statist. Probability, pages 281–297, 1967.

[Maes2002] Sam Maes, Karl Tuyls, Bram Vanschoenwinkel et Bernard Manderick. *Credit card fraud detection using Bayesian and neural networks*. In Proceedings of the 1st international naiso congress on neuro fuzzy technologies, volume 261, page 270, 2002.

[Mahmoudi2015] Nader Mahmoudi et Ekrem Duman. *Detecting credit card fraud by modified Fisher discriminant analysis*. Expert Systems with Applications, vol. 42, no. 5, pages 2510–2516, 2015.

[Malini2017] N Malini et M Pushpa. *Analysis on credit card fraud identification techniques based on KNN and outlier detection*. In 2017 third international conference on advances in electrical, electronics, information, communication and bio-informatics (AEEICB), pages 255–258. IEEE, 2017.

[McInnes2018] Leland McInnes, John Healy et James Melville. *Umap : Uniform manifold approximation and projection for dimension reduction*. arXiv preprint arXiv :1802.03426, 2018.

[Mikolov2013] Tomas Mikolov, Kai Chen, Greg Corrado et Jeffrey Dean. *Efficient estimation of word representations in vector space*. arXiv preprint arXiv :1301.3781, 2013.

[Minegishi2011] Tatsuya Minegishi et Ayahiko Niimi. *Proposal of credit card fraudulent use detection by online-type decision tree construction and verification of generality*. International Journal for Information Security Research (IJISR), vol. 1, no. 4, pages 229–235, 2011.

[More2016] Ajinkya More. *Survey of resampling techniques for improving classification performance in unbalanced datasets*. arXiv preprint arXiv :1608.06048, 2016.

[Muhlbaier2008] Michael D Muhlbaier, Apostolos Topalis et Robi Polikar. *Combining Ensemble of Classifiers With Dynamically Weighted Consult-and-Vote for Efficient Incremental Learning of New Classes*. IEEE transactions on neural networks, vol. 20, no. 1, pages 152–168, 2008.

[Phua2004] Clifton Phua, Damminda Alahakoon et Vincent Lee. *Minority report in fraud detection : classification of skewed data*. Acm sigkdd explorations newsletter, vol. 6, no. 1, pages 50–59, 2004.

[Phua2005] Clifton Phua, Ross Gayler, Vincent Lee et Kate Smith. *On the approximate communal fraud scoring of credit applications*. Proceedings of Credit Scoring and Credit Control, pages 1–10, 2005.

[Phua2010] Clifton Phua, Vincent Lee, Kate Smith et Ross Gayler. *A comprehensive survey of data mining-based fraud detection research*. arXiv preprint arXiv :1009.6119, 2010.

[Popat2018] Rimpal R Popat et Jayesh Chaudhary. *A survey on credit card fraud detection using machine learning*. In 2018 2nd international conference on trends in electronics and informatics (ICOEI), pages 1120–1125. IEEE, 2018.

[Pun2012] Joseph Pun et Yuri Lawryshyn. *Improving credit card fraud detection using a meta-classification strategy*. International Journal of Computer Applications, vol. 56, no. 10, 2012.

[Quah2008] Jon TS Quah et M Sriganesh. *Real-time credit card fraud detection using computational intelligence*. Expert systems with applications, vol. 35, no. 4, pages 1721–1732, 2008.

[Quinlan2014] J Ross Quinlan. C4. 5 : programs for machine learning. Elsevier, 2014.

[RamaKalyani2012] K RamaKalyani et D UmaDevi. *Fraud detection of credit card payment system by genetic algorithm*. International Journal of Scientific & Engineering Research, vol. 3, no. 7, pages 1–6, 2012.

[Randhawa2018] Kuldeep Randhawa, Chu Kiong Loo, Manjeevan Seera, Chee Peng Lim et Asoke K Nandi. *Credit card fraud detection using AdaBoost and majority voting*. IEEE access, vol. 6, pages 14277–14284, 2018.

[Report2019] Nilson Report. *Card fraud losses reach $28.65 billion*. 2019.

[Robinson2016] Ben Robinson et Joel Winteregg. A-z of banking fraud 2016. Temenos and NetGuardians, 2016.

[Rumelhart1986] David E Rumelhart, Geoffrey E Hinton et Ronald J Williams. *Learning representations by back-propagating errors*. nature, vol. 323, no. 6088, pages 533–536, 1986.

[Sahin2010] Yusuf Sahin et Ekrem Duman. *Detecting credit card fraud by decision trees and support vector machines*. In World Congress on Engineering 2012. July 4-6, 2012. London, UK., volume 2188, pages 442–447. International Association of Engineers, 2010.

[Sahin2013] Yusuf Sahin, Serol Bulkan et Ekrem Duman. *A cost-sensitive decision tree approach for fraud detection*. Expert Systems with Applications, vol. 40, no. 15, pages 5916–5923, 2013.

[Sánchez2009] Daniel Sánchez, MA Vila, L Cerda et José-Maria Serrano. *Association rules applied to credit card fraud detection*. Expert systems with applications, vol. 36, no. 2, pages 3630–3640, 2009.

[Santoso2017] B Santoso, H Wijayanto, KA Notodiputro et B Sartono. *Synthetic over sampling methods for handling class imbalanced problems : A review*. In IOP conference series : earth and environmental science, volume 58, page 012031. IOP Publishing, 2017.

[Schapire1990] Robert E Schapire. *The strength of weak learnability*. Machine learning, vol. 5, no. 2, pages 197–227, 1990.

[Srivastava2008] Abhinav Srivastava, Amlan Kundu, Shamik Sural et Arun Majumdar. *Credit card fraud detection using hidden Markov model*. IEEE Transactions on dependable and secure computing, vol. 5, no. 1, pages 37–48, 2008.

[Steinbach2000] Michael Steinbach, George Karypis et Vipin Kumar. *A comparison of document clustering techniques*. 2000.

[Sutskever2014] Ilya Sutskever, Oriol Vinyals et Quoc V Le. *Sequence to sequence learning with neural networks*. Advances in neural information processing systems, vol. 27, 2014.

[Syeda2002] Mubeena Syeda, Yan-Qing Zhang et Yi Pan. *Parallel granular neural networks for fast credit card fraud detection*. In 2002 IEEE World Congress on Computational Intelligence. 2002 IEEE International Conference on Fuzzy Systems. FUZZ-IEEE'02. Proceedings (Cat. No. 02CH37291), volume 1, pages 572–577. IEEE, 2002.

[Tavazoie1999] Saeed Tavazoie, Jason D Hughes, Michael J Campbell, Raymond J Cho et George M Church. *Systematic determination of genetic network architecture*. Nature genetics, vol. 22, no. 3, pages 281–285, 1999.

[Tomek1976] Ivan Tomek. *A generalization of the k-NN rule*. IEEE Transactions on Systems, Man, and Cybernetics, no. 2, pages 121–126, 1976.

[Valentini2002] Giorgio Valentini et Francesco Masulli. *Ensembles of learning machines*. In Italian workshop on neural nets, pages 3–20. Springer, 2002.

[Van der Maaten2008] Laurens Van der Maaten et Geoffrey Hinton. *Visualizing data using t-SNE.* Journal of machine learning research, vol. 9, no. 11, 2008.

[Van Vlasselaer2015] Véronique Van Vlasselaer, Cristián Bravo, Olivier Caelen, Tina Eliassi-Rad, Leman Akoglu, Monique Snoeck et Bart Baesens. *APATE : A novel approach for automated credit card transaction fraud detection using network-based extensions*. Decision Support Systems, vol. 75, pages 38–48, 2015.

[Vaughan2020] Gregory Vaughan. *Efficient big data model selection with applications to fraud detection*. International Journal of Forecasting, vol. 36, no. 3, pages 1116–1127, 2020.

[Vona2017] Leonard W Vona. Fraud data analytics methodology : The fraud scenario approach to uncovering fraud in core business systems. John Wiley & Sons, 2017.

[Wasikowski2009] Mike Wasikowski et Xue-wen Chen. *Combating the small sample class imbalance problem using feature selection*. IEEE Transactions on knowledge and data engineering, vol. 22, no. 10, pages 1388–1400, 2009.

[Werbos1988] Paul J Werbos. *Generalization of backpropagation with application to a recurrent gas market model*. Neural networks, vol. 1, no. 4, pages 339–356, 1988.

[West2016] Jarrod West et Maumita Bhattacharya. *Intelligent financial fraud detection : a comprehensive review*. Computers & security, vol. 57, pages 47–66, 2016.

[Weston2008] David J Weston, David J Hand, Niall M Adams, Christopher Whitrow et Piotr Juszczak. *Plastic card fraud detection using peer group analysis*. Advances in Data Analysis and Classification, vol. 2, no. 1, pages 45–62, 2008.

[Whitrow2009] Christopher Whitrow, David J Hand, Piotr Juszczak, David Weston et Niall M Adams. *Transaction aggregation as a strategy for credit card fraud detection*. Data mining and knowledge discovery, vol. 18, no. 1, pages 30–55, 2009.

[Xu2015] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel et Yoshua Bengio. *Show, attend and tell : Neural image caption generation with visual attention*. In International conference on machine learning, pages 2048–2057. PMLR, 2015.

[Yeung2003] Ka Yee Yeung, Mario Medvedovic et Roger E Bumgarner. *Clustering gene-expression data with repeated measurements*. Genome biology, vol. 4, no. 5, pages 1–17, 2003.

[Zafar2018] Aliza Zafar et Mehreen Sirshar. *A survey on application of Data Mining techniques ; it's proficiency in fraud detection of credit card*. Res Rev J Eng Technol, vol. 7, pages 15–23, 2018.

[Zareapoor2015] Masoumeh Zareapoor, Pourya Shamsolmoali*et al. Application of credit card fraud detection : Based on bagging ensemble classifier*. Procedia computer science, vol. 48, no. 2015, pages 679–685, 2015.

[Zaslavsky2006] Vladimir Zaslavsky et Anna Strizhak. *Credit card fraud detection using self-organizing maps*. Information and Security, vol. 18, page 48, 2006.

[Zhang2008] Yi Zhang et W Nick Street. *Bagging with adaptive costs*. IEEE transactions on knowledge and data engineering, vol. 20, no. 5, pages 577–588, 2008.

[Zhao2017] Xujun Zhao, Jifu Zhang et Xiao Qin. *LOMA : A local outlier mining algorithm based on attribute relevance analysis*. Expert Systems with Applications, vol. 84, pages 272–280, 2017.

[Zhao2019] Pengya Zhao, Xiangling Fu, Weiqiang Wu, Da Li et Jing Li. *Network-based feature extraction method for fraud detection via label propagation*. In 2019 IEEE International Conference on Big Data and Smart Computing (BigComp), pages 1–6. IEEE, 2019.

[Zheng2018] Lutao Zheng, Guanjun Liu, Chungang Yan et Changjun Jiang. *Transaction fraud detection based on total order relation and behavior diversity*. IEEE Transactions on Computational Social Systems, vol. 5, no. 3, pages 796–806, 2018.

[Zhou2012] Zhi-Hua Zhou. Ensemble methods : foundations and algorithms. CRC press, 2012.