الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي

UNIVERSITÉ BADJI MOKHTAR - ANNABA
BADJI MOKHTAR – ANNABA UNIVERSITY

جامعة باجي مختار – عنابـــــــــــة

Faculté : Science de l'Ingénieur
Département : Informatique
Domaine : Informatique.
Filière : Informatique
Spécialité : Système d'information décisionnel 'SID'

Mémoire

Présenté en vue de l'obtention du Diplôme de Master

Thème:

Classification des données de distribution déséquilibrée en utilisant les méthodes d'échantillonnage

Présenté par : *Chems El-Hak Mahouche*

Encadrant :Dr Nadjette Dendani *Grade MCB Université Badji Mokhtar Annaba*

Jury de Soutenance :

M[ed] Tarek Khadir	Professeur	Université Badji Mokhtar Annaba	Président
Nadjette Dendani	MCB	Université Badji Mokhtar Annaba	Encadrant
Nabiha Azizi	Professeur	Université Badji Mokhtar Annaba	Examinateur

Année Universitaire : 2020/2021

Remerciements

Je remercie :

Mon encadrant Dr. Dendani pour ses conseils et expérience.

Ma chère mère pour son soutien, patience, et encouragements.

Dédicaces

J'ai le grand honneur de dédier ce modeste travail :

A ma chère mère.

A mes amis et camarades de la promotion.

A tous ceux qui m'ont aidé de loin ou de près durant les moments difficiles.

Résumé

Ce mémoire présente la conception et réalisation d'une étude comparative de plusieurs méthodes d'échantillonnage dans le but de mettre en avant les différences entre ces dernières.

Les méthodes sélectionnées sont : IHT, OSS, SMOTE, LoRAS ainsi que leurs possibles hybridations.

Ces dernières sont appliquées de façon à présenter séparément les mesures de performances pour des algorithmes de sous-échantillonnage, sur-échantillonnage et la combinaison des deux.

Les résultats ont montré que la performance de chaque méthode ou hybridation dépend des caractéristiques des datasets et modèles de classification utilisés.

Mots clefs: IHT, OSS, SMOTE, LoRAS, méthodes d'échantillonnage, données déséquilibrées.

Abstract

This report details the conception and implementation of a comparative study of multiple resampling algorithms in order to bring forward their differences.

The selected algorithms are: IHT, OSS, SMOTE, LoRAS as well as their possible combinations.

They are applied in a way that would allow a grasp of their performance based on their category, whether it was under-sampling, over-sampling or both at the same time.

Results have shown that each algorithm or combination's performance depends on the used classification models and dataset features.

Key words: IHT, OSS, SMOTE, LoRAS, resampling methods, imbalanced data.

نبذة مختصرة

قدم هذا التقرير بالتفصيل مفهوم وتنفيذ دراسة مقارنة لخوارزميات إعادة أخذ العينات المتعددة من أجل إبراز الاختلافات بينهما.

كانت الخوارزميات المختارة هي: IHT، OSS، SMOTE، LoRAS بالإضافة إلى مجموعاتها الممكنة.
تم تطبيقها بطريقة تسمح بفهم أدائها بناءً على فئتها ، سواء كان ذلك بسبب نقص العينات ، أو الإفراط في أخذ العينات أو كليهما في نفس الوقت.
أظهرت النتائج أن أداء كل خوارزمية أو مجموعة يعتمد على نماذج التصنيف المستخدمة وميزات مجموعة البيانات.

الكلمات الدالة :

IHT, OSS, SMOTE, LoRAS, طرق إعادة التشكيل ,بيانات غير متوازنة.

Table des matières

Remerciements 2

Dédicaces 3

Table des matières 6

Introduction 8

1. Contexte du projet 8
2. Problématique 8
3. Motivations 8
4. Objectifs 9
5. Contenu du mémoire 9

Chapitre 1 : Classification des données déséquilibrées : Concepts de base 10

1. La classification 10
   - 1.1. Les réseaux de neurones artificiels (RNA) 10
   - 1.2. Les machines à vecteurs de support (SVM) 10
   - 1.3. Les K plus proches voisins (KNN) 11
2. Les données déséquilibrées 11
3. Principes de fonctionnement des méthodes d'échantillonnage 13
   - 3.1. Sous-échantillonnage 13
     - Condensed Nearest Neighbor (CNN) [WEB1] 13
     - Tomek Links (TL) [WEB1] 13
     - One Sided Selection (OSS) [WEB1] 14
     - Instance Hardness Threshold (IHT) [WEB2] 14
   - 3.2. Sur-échantillonnage 15
     - Synthetic Minority Oversampling Technique (SMOTE) 15
     - Localized Randomized Affine Shadowsampling (LoRAS) [WEB3] 15

Chapitre 2 : Etude bibliographique des travaux connexes 16

Chapitre 3 : Conception et réalisation 24

1. Datasets 26
   - 1.1. Pima Indians Diabetes Dataset (PIDD) 26

1.2. Parkinson 27
2. Méthodes d'échantillonnage sélectionnées 28
3. Mesures de performance 28
3.1. Accuracy 28
3.2. Precision 28
3.3. Recall 29
3.4. Specificity / True Negative Rate 29
3.5. False Positive Rate 29
3.6. F1 score 29
3.7. ROC – AUC 30
4. Résultats et interprétations 30
Conclusion et perspectives 46
Bibliographie 47
Webographie 49

Introduction

1. Contexte du projet

Depuis ses origines, l'intelligence artificielle avait généralement pour objectif de doter les machines de capacité à pouvoir effectuer des tâches réputées "intelligentes" tendant à rendre la machine capable d'acquérir de l'information, raisonner dans une situation statique ou dynamique, résoudre des problèmes combinatoires, faire un diagnostic, proposer une décision ou un plan d'action et expliquer. C'est dans ce cadre qu'apparaît l'apprentissage automatique qui constitue l'un des sous-domaines de l'intelligence artificielle les plus prometteurs de cette dernière décennie [ZH15].

2. Problématique

Comme dans tout domaine de recherche, plusieurs problèmes peuvent intervenir. Le plus connu est l'influence de la qualité de données sur le bon apprentissage comme le représentent clairement les données déséquilibrées [ZH15]. Ces dernières nuisent aux performances des modèles de classification durant la phase d'apprentissage grâce à une distribution de classes non équitable, engendrant ainsi un pouvoir discriminatif biaisé du modèle concerné.

3. Motivations

L'apprentissage automatique est un domaine où différentes méthodes peuvent être employées (parfois même combinées) pour garantir des résultats à la hauteur des attentes selon le contexte donné. Néanmoins, les distributions déséquilibrées des classes de données restent un problème de taille auquel il peut être difficile d'y remédier.

Le motif de ce projet est de permettre l'élaboration de meilleures approches de classification impliquant des données déséquilibrées dans des contextes similaires.

4. Objectifs

L'objectif de ce travail est de fournir une vue d'ensemble par le biais d'une étude comparative des méthodes utilisées pour remédier au problème des données déséquilibrées afin de répondre au motif précédemment évoqué.

5. Contenu du mémoire

Le présent rapport comporte 3 chapitres :

- Chapitre 1 : « Classification des données déséquilibrées : Concepts de base », donne des généralités sur la classification, les données déséquilibrées et détailles les méthodes relatives à ces dernières.
- Chapitre 2 : « Etude bibliographique des travaux connexes », liste des travaux relatifs dans le domaine d'échantillonnage.
- Chapitre 3 : « Conception et réalisation », liste les techniques utilisées dans ce projet puis présente l'étude expérimentale.

En dernier lieu, une conclusion générale qui résume les points essentiels de ce travail.

Chapitre 1 : Classification des données déséquilibrées : Concepts de base

1. La classification

Les méthodes de classification ont pour but d'identifier les classes auxquelles appartiennent des objets à partir de certains paramètres descriptifs. Elles s'appliquent à un grand nombre d'activités humaines et conviennent en particulier au problème de la prise de décision automatisée [HADK20A].

Cette partie présente une définition de trois modèles de classification (SVM, k-plus proche voisins et réseaux de neurones) incluant leur principe de fonctionnement. Ces méthodes ont montrés leurs efficacités dans des domaines d'applications très variés tels que le traitement d'image, la bio-informatique, la finance, la catégorisation de textes et le diagnostic médical [HCI11].

1.1. Les réseaux de neurones artificiels (RNA)

Par analogie avec le réseau de neurones du cerveau humain, un réseau de neurones artificiel est un ensemble de neurones virtuels liés par un réseau de même nature.

Ces neurones sont chacun une implémentation de concepts mathématiques : Une fonction algébrique bornée non linéaire est appliquée sur des données en entrée sous forme de variables pour produire une valeur dépendante de paramètres appelés « poids » ou « coefficients » qui servira à aiguiller les décisions d'autres neurones et au final, celle du réseau entier.

Le réseau peut ne pas fournir les résultats escomptés dès le départ, et pour remédier à cela, il suit un apprentissage par cycle appelé « époque » sur une base de données dédiée afin d'ajuster ses poids en fonction des résultats de chaque époque.

1.2. Les machines à vecteurs de support (SVM)

Une machine à vecteurs de support est un algorithme d'apprentissage automatique supervisé qui permet de résoudre des problèmes tant de classification que de régression ou de détection d'anomalie. Ils sont connus pour leurs solides garanties théoriques, leur grande flexibilité, ainsi que leur simplicité de mise en œuvre [HADK20B].

Les SVM reposent sur l'idée de trouver une droite ou un hyperplan (selon la dimension du problème), représentant une frontière dans le but de séparer les données en classes en utilisant une fonction de décision couramment notée *f(x)*, de telle façon à maximiser la distance entre cette dernière et les différents groupes de données. Cette distance est aussi

appelée « marge » et les SVM sont ainsi qualifiés de « séparateurs à vaste marge », les « vecteurs de support » étant les données les plus proches de la frontière [HADK20B].

1.3. Les K plus proches voisins (KNN)

C'est un algorithme d'apprentissage supervisé non paramétrique qui repose sur un principe de vote parmi K instances de classes données. La nouvelle instance à classifier se voit octroyer un vote de chacune des K instances minimisant au maximum la distance avec celle à classifier. Ce vote représente la classe à laquelle cette nouvelle instance est susceptible d'appartenir. Cette dernière est donc étiquetée par la classe ayant reçu le plus de votes.

Plusieurs types de distances peuvent être employés mais le plus commun est la distance euclidienne.

Le nombre d'instances votantes K est le facteur principal décidant de la performance de l'algorithme : une valeur plus importante signifie une classification plus précise au prix d'un temps de calcul plus important et vice-versa.

2. Les données déséquilibrées

L'apprentissage automatique permet d'affiner un modèle de classification grâce à des échantillons de données qui lui sont couramment fournis par une ou plusieurs bases de données.

En règle générale, tout modèle de classification tente à pencher vers les classes dont il a utilisé le plus d'instances durant son apprentissage lorsqu'il s'agit de classifier de nouvelles instances qui seraient encore méconnues. Cela signifie que le modèle en question disposerait d'une meilleure capacité de prédiction des classes dont il a appris le plus au point où il commet l'erreur d'attribuer ces classes à des données qui n'y appartiennent pas.

Cette situation est plus probable lorsqu'on utilise des sources de données non balancées, n'ayant pas un nombre égal ou similaire d'instances de chaque classe que l'on souhaite traiter. C'est le cas le plus courant dans la réalité et plus spécifiquement dans le domaine médical où les instances des cas les plus importants sont d'un nombre assez inférieur aux autres.

Afin de remédier au problème, plusieurs méthodes furent élaborées. Ces méthodes peuvent être réparties selon la hiérarchie suivante :

- Méthodes agissant sur le modèle de classification utilisé.
    - Séquentielles.
    - Parallèles.
- Méthodes agissant sur les données utilisées pour l'apprentissage.
    - Sur-échantillonnage.
    - Sous-échantillonnage.

Dans le domaine d'échantillonnage, et parmi les données d'apprentissage :

- Une **classe** est considérée comme « **majoritaire** » lorsqu'elle dispose d'un **plus grand nombre d'instances** par rapport aux autres.
- À l'inverse, elle est considérée comme « **minoritaire** » lorsqu'elle dispose **d'un nombre moins important d'instances** par rapport aux autres.

Les méthodes de **sur-échantillonnage** permettent **d'augmenter** le nombre d'instances des classes **minoritaires** afin qu'il soit égal à celui des **majoritaires**.

De l'autre côté, les méthodes de **sous-échantillonnage** permettent **de réduire** le nombre d'instances des classes **majoritaires** afin qu'il soit égal à celui des **minoritaires**.

Parmi ces méthodes, on compte :

Sous-échantillonnage	**Sur-échantillonnage**
• Condensed Nearest Neighbor (CNN) • Tomek Links (TL) • One Sided Selection (OSS) • Instance Hardness Threshold (IHT)	• Synthetic Minority Oversampling Technique (SMOTE) • Localized Randomized Affine Shadow sampling (LoRAS)

Ce projet se focalise sur la classification des données déséquilibrées en utilisant les méthodes d'échantillonnage et leurs hybridations possibles.

3. Principes de fonctionnement des méthodes d'échantillonnage

3.1. Sous-échantillonnage

- **Condensed Nearest Neighbor (CNN) [WEB1]**

Cet algorithme est basé sur le KNN. Son but est de sélectionner les instances à garder parmi les données d'apprentissage.

Il repose sur l'utilisation d'un ensemble qualifié d'entrepôt contenant au départ la totalité des instances de classe minoritaire et un nombre donné d'instances aléatoires de classe majoritaire.

À chaque itération, le KNN utilise les instances de l'entrepôt pour classifier l'ensemble des données restantes (en d'autres termes, de classe majoritaire). S'il n'en classe pas une correctement, celle-ci est rajoutée à l'entrepôt.

Au final l'entrepôt ne contiendra que les instances retenues de l'ensemble original.

Cette approche permet d'obtenir un ensemble de données d'apprentissage réduit n'impactant pas la performance du modèle de classification utilisé. Cet ensemble est appelé « ensemble consistent minimal ».

L'inconvénient de cet algorithme est que l'aspect aléatoire de l'initialisation peut résulter en un ensemble final redondant.

- **Tomek Links (TL) [WEB1]**

Contrairement au CNN, cette méthode vise à sélectionner les instances à supprimer parmi les données d'apprentissage.

Le but de cet algorithme est principalement de solidifier la frontière séparant les classes minoritaire et majoritaire en éliminant les instances ambigües pour permettre une discrimination plus stricte entre les deux.

En utilisant la distance euclidienne, l'algorithme cherche des paires d'instances étant plus proches voisins comportant une instance de chacune des deux classes.

Ces paires formeraient alors la frontière ambigüe, et leur partie majoritaire sera alors supprimée pour créer une frontière plus stricte tout en réduisant l'effectif de la classe majoritaire.

Cette méthode n'est en pratique efficace que lorsqu'elle est combinée avec d'autres, par exemple CNN.

- **One Sided Selection (OSS) [WEB1]**

Ceci est l'hybridation des méthodes CNN et TL.

Dans un premier temps, TL est appliqué pour réduire le nombre d'instances de classe majoritaire en éliminant celles formant la frontière ambigüe.

L'application de CNN permettrait alors d'obtenir un ensemble minimal consistent avec une frontière plus solide.

- **Instance Hardness Threshold (IHT) [WEB2]**

Cette méthode est plus ou moins similaire à TL pour ce qui est de réduire l'effectif de classe majoritaire en se focalisant sur les instances frontières entre classes. La différence réside dans la manière de sélectionner les instances à supprimer parmi ces dernières.

Dans un premier temps, N classificateurs sont sélectionnés, entrainés et testés sur le même ensemble de données non balancées.

Puis, chaque classificateur se voit assigner la probabilité qu'il classifie correctement chaque instance grâce aux mesures de performances et une fonction indicatrice.

L'indicateur IH (Instance Hardness) est donc défini par « 1 – cette probabilité » pour chaque instance. Cet indicateur exprime la probabilité de ne pas classifier correctement une instance donnée.

L'IHT (Instance Hardness Threshold) est donc le seuil auquel une instance est considérée comme « difficile à classifier ». C'est ce type d'instances qui sera supprimé par cet algorithme.

3.2. Sur-échantillonnage

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

Cette technique permet de créer des instances synthétiques (artificielles) de classe minoritaire afin d'égaliser en nombre ceux de classe majoritaire en utilisant KNN.

Au départ, une instance aléatoire de classe minoritaire est sélectionnée. L'algorithme cherche alors ses K plus proches voisins. Parmi ces derniers, un seul est retenu aléatoirement.

La ligne reliant l'instance minoritaire de départ et son plus proche voisin retenu est alors formée et un point aléatoire en est sélectionné comme nouvelle instance de classe minoritaire.

Cette démarche est donc répétée jusqu'à équilibre entre classes.

L'inconvénient de cet algorithme est qu'il tend à généraliser la classe minoritaire et de ce fait, à réduire la précision du modèle par rapport à la classe majoritaire.

- **Localized Randomized Affine Shadowsampling (LoRAS) [WEB3]**

Cet algorithme vise à créer des instances de classe minoritaire artificielles en générant une combinaison linéaire aléatoire à partir d'instances bruitées.

LoRAS génère un bruit dans de petits voisinages des instances de classe minoritaire pour créer des « échantillons d'ombre ». Il crée ensuite l'instance synthétique grâce à une combinaison linéaire de plusieurs de ces derniers.

Cette procédure est répétée pour toutes les instances de classe minoritaire.

Au final, les instances synthétiques sont rajoutées pour combler la différence d'effectif entre classes.

Ceci permet une meilleure exploration des possibilités de création des données synthétiques (plus de 2 instances incluses par combinaison linéaire) tout en réduisant leur redondance (le balayage de toutes les instances de classe minoritaire permet de contourner le risque de reprendre des instances déjà utilisées).

Chapitre 2 : Etude bibliographique des travaux connexes

Le tableau suivant présente quelques travaux d'échantillonnage de données déséquilibrées :

Auteurs	Titres	Descriptions
Lee et al. 2010 [LYCL10]	A hybrid algorithm applied to classify unbalanced data.	Ce document présente l'hybridation de 3 types d'algorithmes : sur-échantillonnage aléatoire, arbre de décision, optimisation par essaims particulaires (PSO). Premièrement, ils ont traité la base par l'utilisation de sur-échantillonnage aléatoire, ensuite PSO et arbre de décision pour sélectionner les caractéristiques intéressantes et classifier la base de données. Cette méthode a été comparée avec l'utilisation d'autres classificateurs qui sont KNN, SVM et arbre de décision. La meilleure précision obtenue est 99,5% réalisé par la méthode proposée sur la base de données zoo.
Gao et al .2011 [GHCH11]	A combined SMOTE and PSO based RBF classifier for two-class imbalanced problems.	Cet article combine des outils de résolution du déséquilibre comme SMOTE et un algorithme d'optimisation par essaims particulaires (pso) avvec un classificateur de type réseau de neurones dont la fonction d'activation est la fonction radiale (RBF). Les

		bases de données utilisées sont : Pima, haberman, ADI. Ils ont créé différents changement du taux de sur-échantillonnage sur les bases d'apprentissages (0%, 100%, 300%, 500%, 800%, 1000%, 1500%, et 2000%). La méthode proposée a été comparée avec le modèle Knn. Les résultats expérimentaux pour cette étude ont démontré que la technique proposée est meilleure grâce à son principe d'équilibrage et de minimisation d'erreurs.
Garcia et al. 2012 [GSM12]	On the effectiveness of preprocessing methods when dealing with different levels of class imbalance.	L'article présente une méthode qui permet de déterminer l'influence du degré de déséquilibre sur les techniques d'échantillonnage, utilisant différents algorithmes d'apprentissages sur 17 bases de données. Ils ont divisé ces dernières en bases avec un déséquilibre fort et faible. Les classificateurs sont appliqués sur les données prétraitées par les différentes stratégies d'échantillonnage. Les méthodes d'échantillonnage choisies sont : RUS, WE+MSS, SMOTE, et gg-SMOTE. Les modèles de classification sont : KNN, SVM, réseaux bayésiens,

		arbre de décision et RBF. Les résultats des bases de données avec un déséquilibre élevé montrent que le sur-échantillonnage et plus approprié parce qu'il rajoute des instances. Par contre, de l'application du sous-échantillonnage survient une perte d'information. Pour la deuxième catégorie, les deux techniques atteignent des résultats similaires mais toujours mieux que la base originale.
Rahman et al. 2013 [RD13]	Cluster Based UnderSampling for Unbalanced Cardiovascular Data.	Ce papier réalise une étude comparative entre SMOTE et une méthode de sous-échantillonnage basée sur le clustering. Les auteurs ont appliqué deux algorithmes d'apprentissage : arbre de décision et FURIA (Fuzzy Unordered Rule Induction Algorithm,) sur la base de données cardiovasculaire. SMOTE montre une bonne classification pour les deux modèles mais le sous-échantillonnage est meilleur en termes de temps d'exécution.
Cateni et al. 2014 [CCV14]	A method for resampling imbalanced data sets in binary classification tasks for real-world problems.	La méthode proposée est de combiner les deux types d'échantillonnage (sous et sur-échantillonnage) afin d'obtenir une base de données

		équilibrée. Cette méthode est nommée "Similarity based Undersampling and Normal Distribution based Oversampling" (SUNDO). Les auteurs ont comparé SUNDO avec la classification de la base originale sans aucun changement, SMOTE, Under sampling based clustering (SBC) en utilisant les 4 modèles SVM, arbre de décision, réseaux de neurones et réseaux bayésien. L'application est réalisée sur trois bases de données : metal sheet quality assessment, BreastCancer, occlusion detection. Pour toutes les bases de données et tous les modèles et pour un même degré de déséquilibre, l'approche atteint des bonnes performances et même SMOTE a une haute précision.
Mazurowskia and al. 2008 [M08]	Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance.	Les auteurs ont proposé deux méthodes basées sur les réseaux de neurones, en d'autres termes la retro-propagation et l'optimisation par essaims particulaires pour vérifier l'utilité de l'échantillonnage. Selon eux, l'algorithme PSO était relativement sensible au problème de déséquilibre de

		données avec peu d'instances et beaucoup de caractéristiques.
Francisco Fernández-Navarro and al. 2011 [F11]	A dynamic oversampling procedure based on sensitivity for multi-class problems.	Les auteurs ont proposé une méthode d'échantillonnage dynamique, utilisant deux autres pour inspirer la classification de données déséquilibrées. Cette technique est intégrée dans un algorithme mimétique (MA) qui réduit les réseaux de neurones à fonctions de base radiale (RBFNNs). Les données d'apprentissages sont ré échantillonnées dans le but de traiter le déséquilibre de classes en deux étapes : La première étant l'utilisation de techniques de sur-échantillonnage pour équilibrer les groupes minoritaires. L'algorithme mimétique sur-échantillonne les données à plusieurs degrés et fournit un nouveau mode avec le plus petit niveau de sensitivité. La technologie de l'auteur a été pesée sur 13 datasets de référence en plus d'être compare à d'autres techniques basées sur les réseaux de neurones traitant des problèmes de déséquilibre de classes.
Jason Van Hulse and al.	Knowledge discovery from	L'auteur a proposé une nouvelle méthode pour

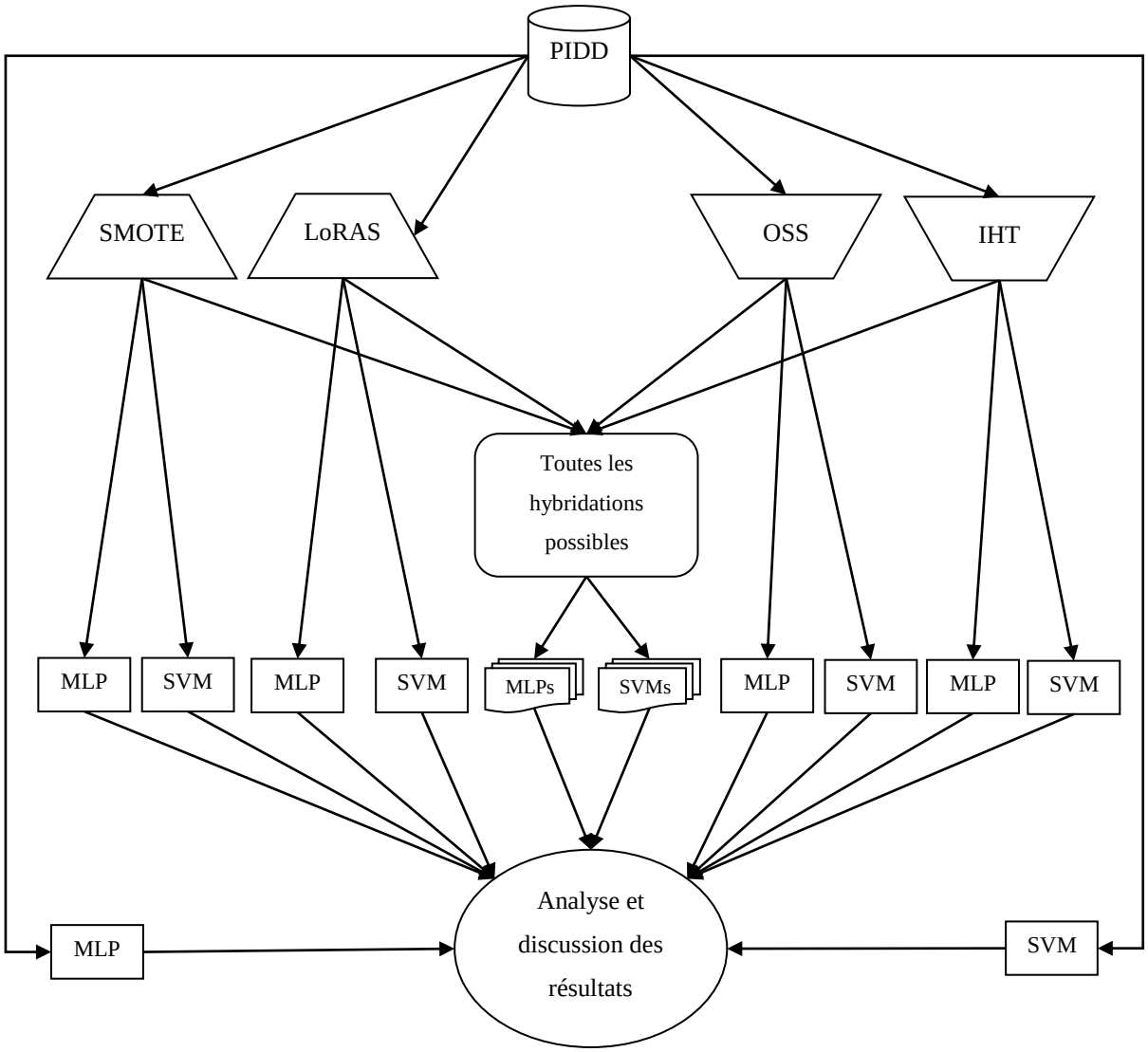
2009 [J09]	imbalanced and noisy data.	surmonter le problème de déséquilibre de classes. Il a utilisé 7 datasets déséquilibrés et a testé sur ces derniers une variété de méthodes d'échantillonnage. Comparé à plusieurs algorithmes d'apprentissage, SVM, KNN, random forest et naïve bayes sont les plus robustes. À partir des résultats expérimentaux, l'auteur a montré qu'en réduisant l'effectif de la classe majoritaire, le sous-échantillonnage aléatoire s'avère être efficace. À la différence des techniques de sur-échantillonnage complexes comme SMOTE et Borderline SMOTE, la méthode Wilson Editing offre de bonnes performances car elle cible les instances mal classifiées.
Hualong Yu and al. 2013 [H13]	ACOSampling: An ant colony optimization-based undersampling method for classifying imbalanced DNA microarray data.	Les auteurs ont proposé une méthode de sous-échantillonnage à heuristique basée sur l'idée d'optimiser des colonies de fourmis pour résoudre le problème de déséquilibre de classes. L'algorithme débute par une méthode de sélection de caractéristiques pour éliminer les gènes bruitées des données. Selon la fréquence

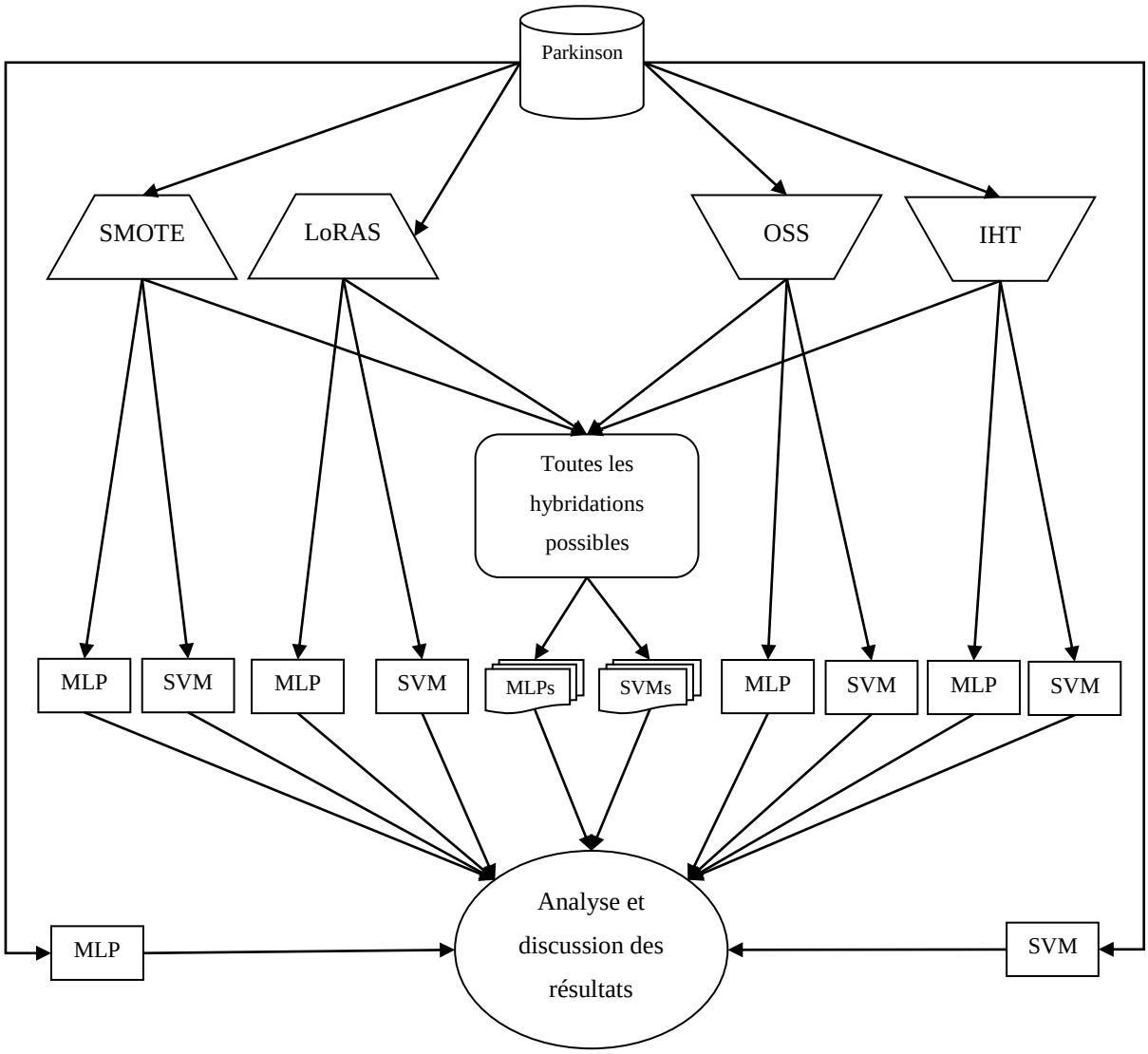
		choisie, il est censé fournir un grand nombre d'échantillons majoritaires. La méthode offre le meilleur ensemble d'équilibres majoritaires. Contrairement à la méthode d'échantillonnage simple, l'inconvénient principal de cette méthode est qu'elle soit plus gourmande en temps.
Show-Jane Yen and Yue-Shi Lee. 2009 [S09]	Cluster-based undersampling approaches for imbalanced data distributions.	Les auteurs ont proposé une technique de sous-échantillonnage basée sur les clusters afin de surmonter le problème de perte de données due à la réduction des instances majoritaires. Cette technique divise la classe majoritaire en K clusters pour en sélectionner les instances avec les bons échantillons de classe minoritaire, puis en produit K datasets combinés. Pour finir, tous les datasets sont classifiés et cela donne la meilleure précision sur les datasets déséquilibrés.
Yang Yong and al. 2012 [Y12]	The Research of Imbalanced Data Set of Sample Sampling Method Based on K-Means Cluster and Genetic Algorithm.	Les auteurs ont proposé une méthode d'échantillonnage basée sur le clustering K-means et les algorithmes génétiques avec l'intention de mettre l'accent sur les performances des classes

		minoritaires qui apparaissent dans les datasets déséquilibrés. Les instances de classe minoritaire furent divisées en clusters avec K-means et les algorithmes génétiques furent utilisés pour obtenir de nouveaux échantillons de chaque cluster en les confirmant efficacement. La combinaison de l'algorithme KNN et SVM avec la technique proposée a révélé l'efficacité qui peut être obtenue en se basant sur les résultats expérimentaux.

Chapitre 3 : Conception et réalisation

Dans un premier temps, ce chapitre détaille les datasets et liste les méthodes d'échantillonnage sujettes à l'étude comparative pour le problème des données déséquilibrées. Puis présente les mesures de performances utilisées suivies des résultats d'application des méthodes et hybridations avec leurs interprétations respectives. Les schémas suivants montrent les étapes suivies lors de la réalisation du travail :





Après analyse et discussion des résultats pour chaque dataset et modèle de classification, une analyse finale générale comparant les résultats des deux datasets a lieu. Dans ce qui suit, l'ensemble des deux schémas est expliqué plus en détails.

1. Datasets

Les datasets utilisés sont PIDD (Pima Indians Diabetes Dataset) et Parkinsons. Ces derniers sont constitués comme suit :

1.1. Pima Indians Diabetes Dataset (PIDD)

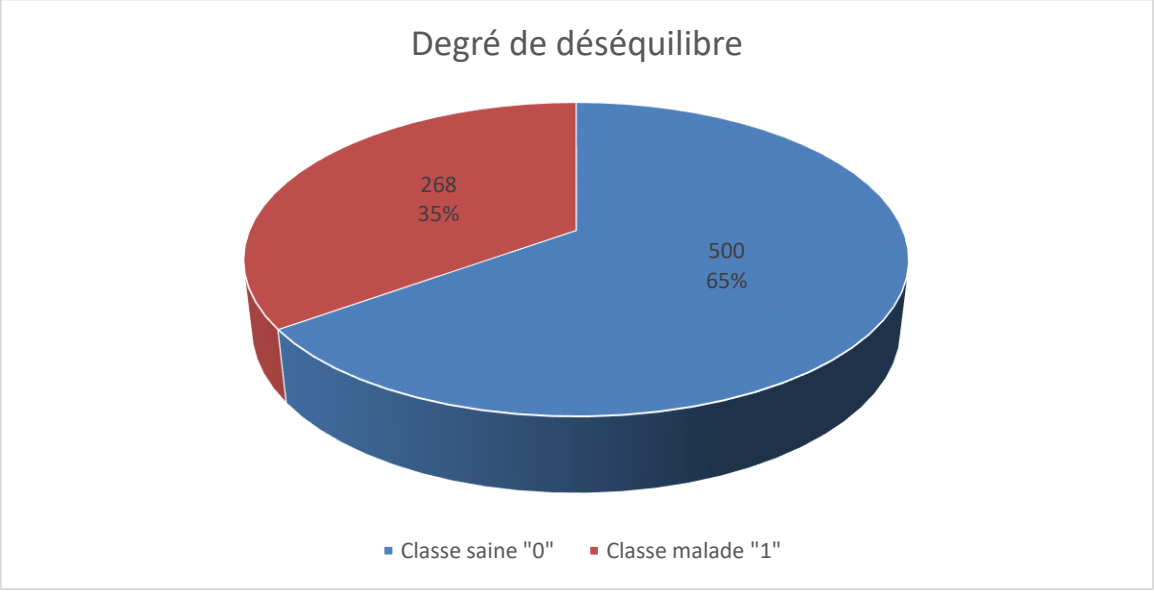
Ce dataset est originaire de l'Institut National de Maladies Diabétiques, Digestives et Rénales. Il s'agit d'une sélection de femmes âgées de 21 ans ou plus d'héritage Pima indien parmi une base de données plus large.

Les caractéristiques utilisées sont :

- Pregnancies : Nombre de fois enceinte.
- Glucose : Concentration du plasma en glucose (test de tolérance au glucose par voie orale de 2 heures).
- BloodPressure : Tension artérielle diastolique (mm Hg).
- SkinThickness : Épaisseur de pliement de peau au niveau des triceps (mm).
- Insulin : Sérum d'insuline de 2 heures (muU/ml).
- BMI : Indice de masse corporelle (poids en kg/ (taille en mètres)^2).
- DiabetesPedigreeFunction : Fonction pedigree de diabetes.
- Age : Âge de l'individu.

Les classes sont représentées par la colonne « status » et sont réparties comme suit :

- 500 instances de classe « 0 » (classe saine) (65%).
- 268 instances de classe « 1 » (classe atteinte de diabète) (35%).



1.2. Parkinson

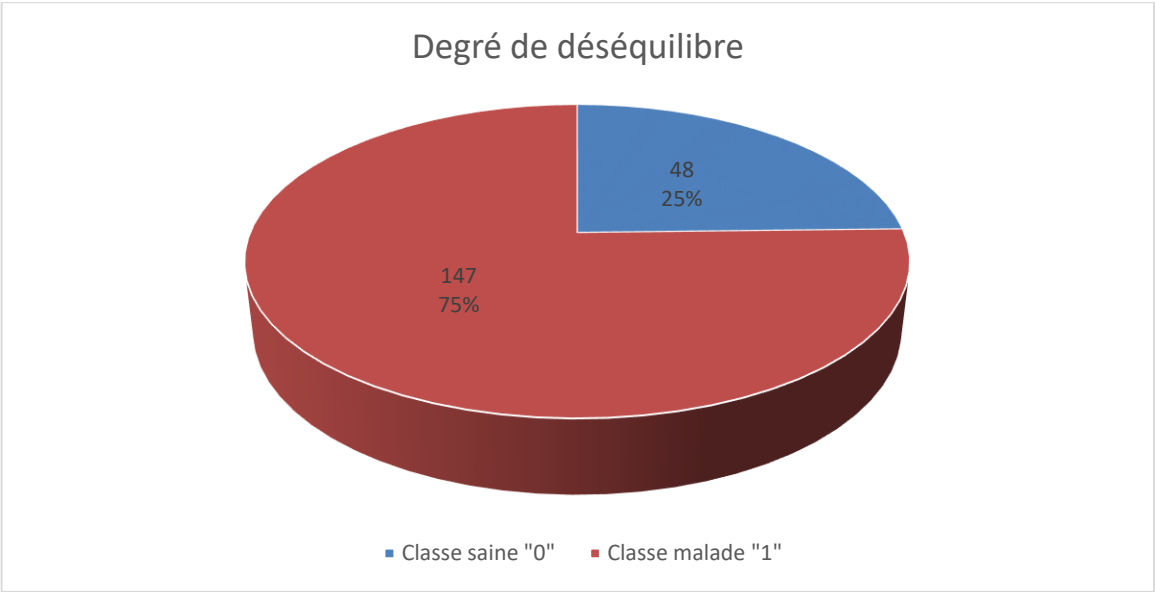
Ce dataset a été créé par Max Little de l'université d'Oxford en collaboration avec le Centre National de Voix et de Parole à Denver, Colorado qui a enregistré les signaux de parole. L'étude originale a publié les méthodes d'extraction de caractéristiques pour les désordres de voix généraux.

Les caractéristiques utilisées sont :

- MDVP :Fo (Hz) : Fréquence vocale fondamentale moyenne.
- MDVP :Fhi (Hz) : Fréquence vocale fondamentale maximale.
- MDVP :Flo (Hz) : Fréquence vocale fondamentale minimale.
- MDVP :Jitter (%), MDVP :Jitter (Abs), MDVP :RAP, MDVP :PPQ, Jitter :DDP : Mesures de variance de la fréquence fondamentale.
- MDVP :Shimmer, MDVP :Shimmer (dB), Shimmer:APQ3, Shimmer:APQ5, MDVP:APQ, Shimmer:DDA: Mesures de variance d'amplitude.
- NHR, HNR : Deux mesures de rapport de bruit par composantes tonales de la voix.
- RPDE, D2 : Deux mesures de complexité dynamiques non linéaires.
- DFA : Exposant d'échelonnage fractal du signal.
- Spread1, spread2, PPE : Trois mesures non linéaires de la variance de la fréquence fondamentale.

Les classes sont représentées par la colonne « status » et sont réparties comme suit :

- 48 instances de classe « 0 » (classe saine) (25%).
- 147 instances de classe « 1 » (classe atteinte de parkinson) (75%).



2. Méthodes d'échantillonnage sélectionnées

Les méthodes concernées par l'étude sont :

- Sous-échantillonnage :
  - Instance Hardness Threshold (IHT)
  - One Sided Selection (OSS)
- Sur-échantillonnage :
  - Synthetic Minority Oversampling Technique (SMOTE)
  - Localized Randomized Affine Shadow sampling (LoRAS)

3. Mesures de performance

Les mesures de performance utilisées sont :

- Accuracy
- Precision
- Recall
- F1 score
- AUC-ROC Curve

Dans l'ensemble de ces mesures, les indicateurs suivants sont employés :

- True positive « tp » : Nombre d'instances positives classifiées positives.
- True negative « tn » : Nombre d'instances négatives classifiées négatives.
- False positive « fp » : Nombre d'instances négatives classifiées positives.
- False negative « fn » : Nombre d'instances positives classifiées négatives.
- « N » : Nombre total d'instances du dataset.

À noter que la classe positive est paramétrable et dans le cas de cette étude, il s'agit de la classe d'intérêt (atteinte de maladie) représentée par « 1 ».

3.1. Accuracy

Mesure la capacité du modèle à correctement classifier les instances de toutes les classes.

Formule de calcul : $Accuracy = \frac{tp+tn}{N}$ (1)

3.2. Precision

Mesure la capacité du modèle à ne pas classifier comme positive une instance négative.

Formule de calcul : $Precision = \frac{tp}{tp+fp}$ (2)

3.3. Recall

Mesure la capacité du modèle à retrouver toutes les instances positives. Elle est aussi appelée « Sensitivity » ou « True Positive Rate ».

Formule de calcul : $Sensitivity = TPR = Recall = \frac{tp}{tp+fn}$ (3)

Le taux de faux négatifs (FNR) nous indique quelle proportion de la classe positive a été mal classée par le classificateur.

$$FNR = \frac{fn}{tp+fn} \quad (4)$$

Si on veut classer correctement la classe positive Un TPR plus élevé et un FNR plus faible sont souhaitables.

3.4. Specificity / True Negative Rate

La spécificité nous indique quelle proportion de la classe négative a été correctement classée.

Formule de calcul : $Sepecificity = TNR = \frac{tn}{tn+fp}$ (5)

3.5. False Positive Rate

Le FPR nous indique quelle proportion de la classe négative a été mal classée par le classificateur

Formule de calcul : $FPR = \frac{fp}{tn+fp} = 1 - Specificity$ (6)

Un TNR plus élevé et un FPR plus faible sont souhaitables car nous voulons classer correctement la classe négative.

Parmi ces métriques, la sensibilité et la spécificité sont peut-être les plus importantes et nous verrons plus tard comment elles sont utilisées pour construire une métrique d'évaluation.

3.6. F1 score

Représente la moyenne pondérée de Precision et Recall. Ces deux dernières apportent une contribution relative équitable au F1 score.

Formule de calcul : $F1_{score} = 2 * \frac{precision*recall}{precision+recall}$ (7)

3.7. ROC – AUC

C’est une mesure de performance pour les problèmes de classification à divers réglages de seuil. ROC est une courbe de probabilité et AUC représente le degré ou la mesure de séparabilité. Il indique dans quelle mesure le modèle est capable de faire la distinction entre les classes. Plus l'AUC est élevée, mieux le modèle prédit les instances de classe 0 comme 0 et ceux de 1 comme 1. Par analogie, plus l'AUC est élevée, plus la distinction est meilleure entre les patients atteints de la maladie et les patients sans maladie.

La courbe ROC est tracée avec TPR par rapport au FPR où TPR est sur l'axe des y et FPR est sur l'axe des x.

4. Résultats et interprétations

Les méthodes citées préalablement furent appliquées aux datasets (découpés en 75% apprentissage et 25% test) comme suit :

- IHT
- OSS
- SMOTE
- LoRAS
- IHT et OSS
- LoRAS et SMOTE
- IHT et SMOTE
- OSS et SMOTE
- IHT avec OSS et SMOTE
- LoRAS et IHT
- LoRAS et OSS
- LoRAS avec IHT et OSS

Parmi ces dernières, SMOTE et LoRAS sont présentés comme suit :

Algorithm *SMOTE*(T, N, k)
Input: Number of minority class samples T; Amount of SMOTE $N\%$; Number of nearest neighbors k
Output: $(N/100) * T$ synthetic minority class samples

1. $(*$ *If* N *is less than 100%, randomize the minority class samples as only a random percent of them will be SMOTEd.* $*)$
2. **if** $N < 100$
3. **then** Randomize the T minority class samples
4. $T = (N/100) * T$
5. $N = 100$
6. **endif**
7. $N = (int)(N/100)$ $(*$ *The amount of SMOTE is assumed to be in integral multiples of 100.* $*)$
8. $k =$ Number of nearest neighbors
9. $numattrs =$ Number of attributes
10. $Sample[\,][\,]$: array for original minority class samples
11. $newindex$: keeps a count of number of synthetic samples generated, initialized to 0
12. $Synthetic[\,][\,]$: array for synthetic samples
    $(*$ *Compute* k *nearest neighbors for each minority class sample only.* $*)$
13. **for** $i \leftarrow 1$ **to** T
14. Compute k nearest neighbors for i, and save the indices in the $nnarray$
15. Populate(N, i, $nnarray$)
16. **endfor**

    $Populate(N,\ i,\ nnarray)$ $(*$ *Function to generate the synthetic samples.* $*)$
17. **while** $N \neq 0$
18. Choose a random number between 1 and k, call it nn. This step chooses one of the k nearest neighbors of i.
19. **for** $attr \leftarrow 1$ **to** $numattrs$
20. Compute: $dif = Sample[nnarray[nn]][attr] - Sample[i][attr]$
21. Compute: $gap =$ random number between 0 and 1
22. $Synthetic[newindex][attr] = Sample[i][attr] + gap * dif$
23. **endfor**
24. $newindex{+}{+}$
25. $N = N - 1$
26. **endwhile**
27. **return** $(*$ *End of Populate.* $*)$

End of Pseudo-Code.

Figure 1: Algorithme SMOTE en pseudocode [CA02]

Algorithm 1: Localized Random Affine Shadowsample (LoRAS) Oversampling

Inputs:

`C_maj`: Majority class parent data points

`C_min`: Minority class parent data points

Parameters:

`k`: Number of nearest neighbors to be considered per parent data point
(default value : 30 if $|C_{\min}| >= 100$, 5 otherwise)

$|S_p|$: Number of generated shadowsamples per parent data point
$\left(\text{default value : } \max\left(\left\lceil \frac{2|F|}{k} \right\rceil, 40\right)\right)$

L_σ: List of standard deviations for normal distributions for adding noise to each feature
(default value : $[0.005, \ldots, 0.005]$)

N_{aff}: Number of shadow points to be chosen for a random affine combination
(default value : $|F|$)

N_{gen}: Number of generated LoRAS points for each nearest neighbors group
$\left(\text{default value : } \frac{|C_{\text{maj}}| - |C_{\min}|}{|C_{\min}|}\right)$

`embedding`: Type of Embedding used to choose minority class neighbourhood (regular or t-embedding)
(default value : 'regular')

`perplexity`: Perplexity of t-embedding (applicable only if `embedding`='t-embedding')
(default value : 30)

Constraint:

$N_{aff} < k * |S_p|$

Initialize `loras_set` as an empty list

For each minority class parent data point `p` in `C_min` **do**

- `neighborhood` ← calculate `k`-nearest neighbors of `p`, as per selected `Embedding` parameter and append `p`
- Initialize `neighborhood_shadow_sample` as an empty list
- **For** each parent data point `q` in `neighborhood` **do**
  - `shadow_points` ← draw $|S_p|$ shadowsamples for `q` drawing noises from normal distributions with corresponding standard deviations L_σ containing elements for every feature
  - Append `shadow_points` to `neighborhood_shadow_sample`
- **Repeat**
  - `selected_points` ← select N_{aff} random shadow points from `neighborhood_shadow_sample`
  - `affine_weights` ← create and normalize random weights for `selected_points`
  - `generated_LoRAS_sample_point` ← `selected_points` · `affine_weights`
  - Append `generated_LoRAS_sample_point` to `loras_set`
- **Until** N_{gen} resulting points are created;

Return resulting set of generated LoRAS data points as `loras_set`

Figure 2 : Algorithme LoRAS en pseudocode [SBNDA20]

Les tableaux et courbes suivantes intègrent les résultats de mesures de performances en pourcentages de chaque méthode et hybridation pour chaque dataset en utilisant les réseaux de neurones (MLP) et les machines à vecteurs de support (SVM). Les diagrammes en barres suivant chaque tableau sont à titre illustratif et offrent une meilleure visualisation des résultats.

- Avant échantillonnage :

Datasets	Classifiers	Accuracy	Precision	Recall	F1_Score
PIDD	MLP	70,83	63,93	53,42	58,21
	SVM	77,6	85,71	49,32	62,61
Parkinson	MLP	87,76	85,37	100	92,11
	SVM	81,63	79,55	100	88,61

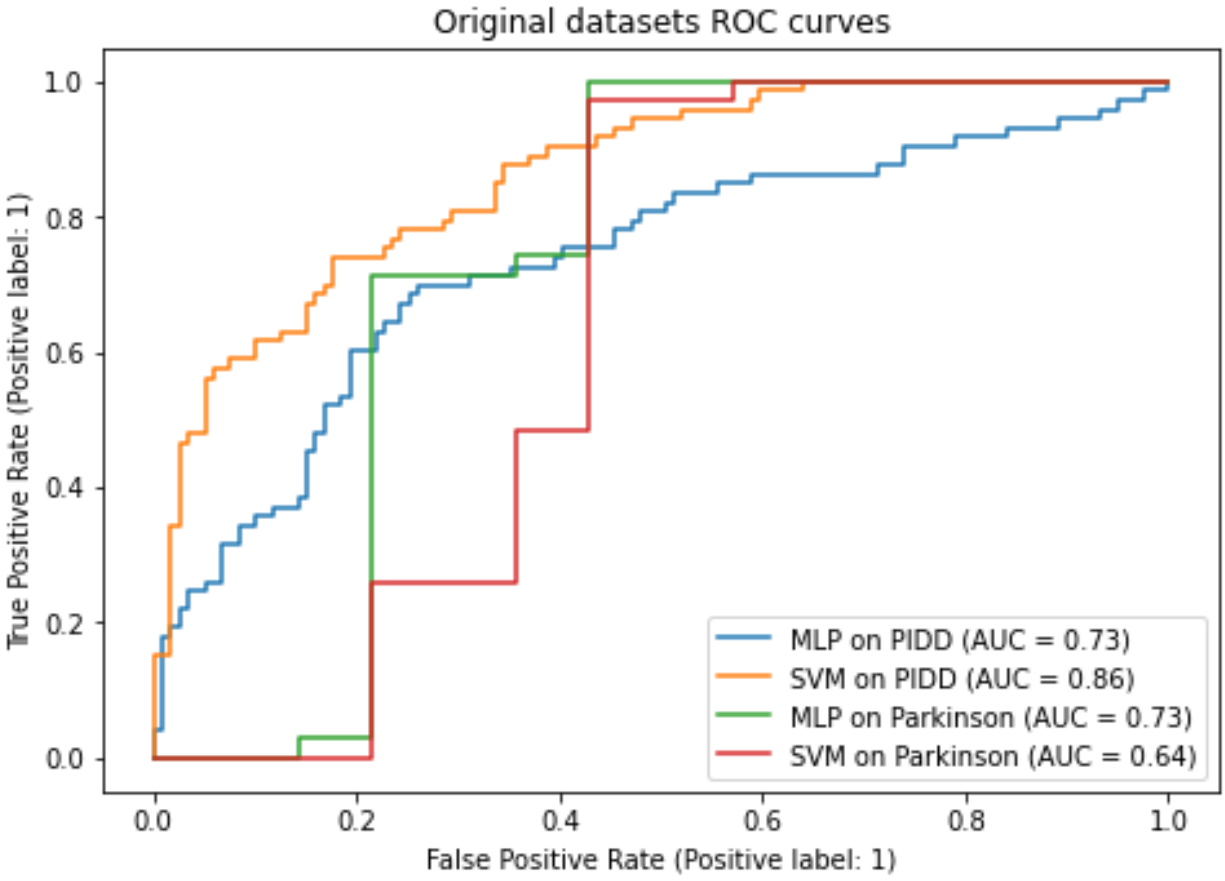


Figure 3 : Courbes ROC des modèles MLP et SVM en utilisant les datasets originaux

- Après échantillonnage :
    - PIDD :
        - MLP :

Methods	Accuracy	Precision	Recall	F1_Score
IHT	70.83	58.25	82.19	68.18
OSS	72.4	64.29	61.64	62.94
SMOTE	67.71	57.53	57.53	57.53
LoRAS	62.5	50.57	60.27	55
IHT et OSS	66.15	53.64	80.82	64.48
LoRAS et SMOTE	67.71	60.38	43.84	50.79
IHT et SMOTE	68.23	55.36	84.93	67.03
OSS et SMOTE	70.31	60.26	64.38	62.25
IHT avec OSS et SMOTE	67.71	55.24	79.45	65.17
LoRAS et IHT	59.9	44.12	20.55	28.04
LoRAS et OSS	66.67	55.56	61.64	58.44
LoRAS avec IHT et OSS	59.9	43.33	17.81	25.24

Figure 4 : Mesures de performances des méthodes d'échantillonnage en utilisant un MLP sur le dataset PIDD

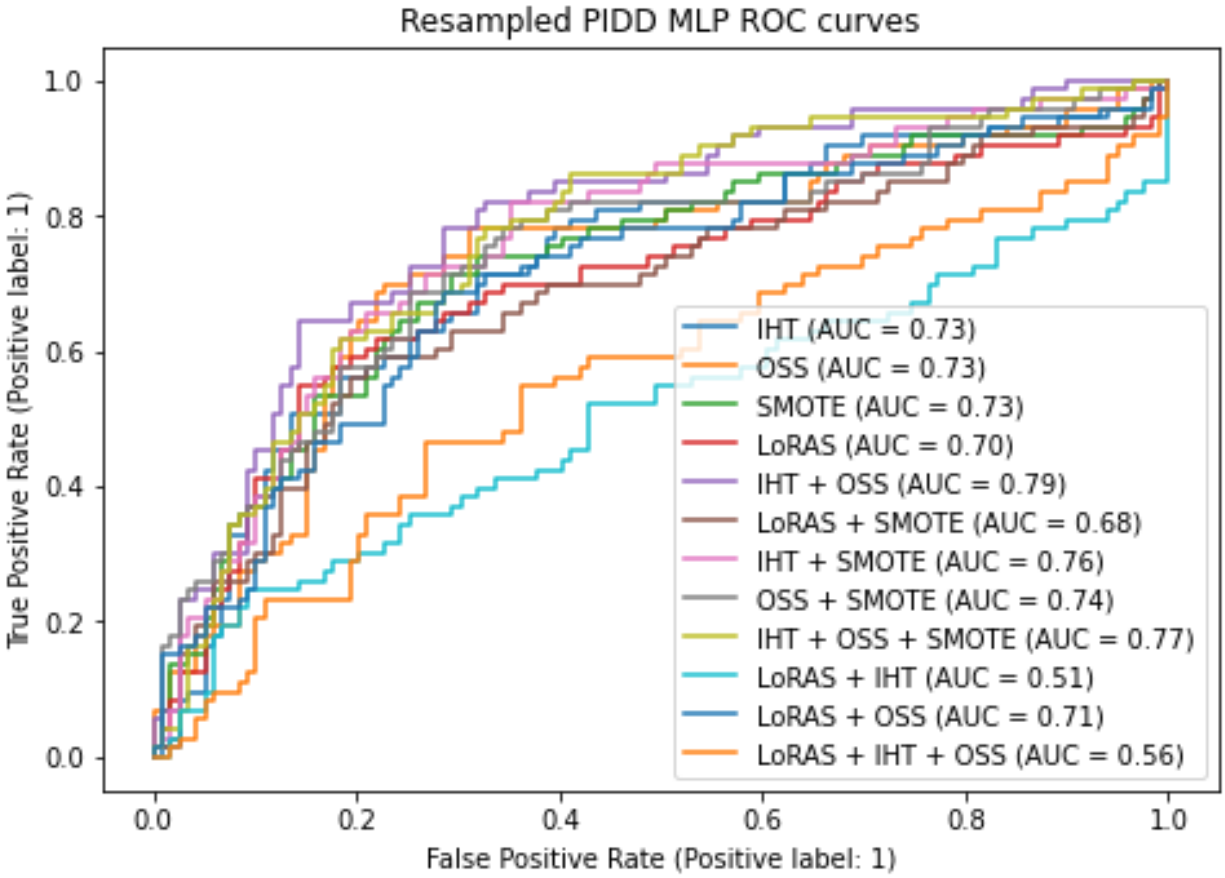


Figure 5 : Courbes ROC des modèles MLP en utilisant le dataset PIDD après échantillonnage

Dans ces résultats on note (voir plus bas pour d'avantage de détails) :

- En termes d'accuracy, IHT est la meilleure des méthodes de base étant donné qu'elle n'en donne pas une pire qu'avec le dataset original.
- En termes de precision, c'est OSS qui l'emporte avec une amélioration par rapport au cas du dataset original.
- En termes de recall, IHT apporte une amélioration plus importante que les autres méthodes de base.
- IHT dispose donc du meilleur F1 score parmi ces dernières.
- La combinaison d'IHT avec OSS semble exprimer le manque de compatibilité entre les deux méthodes étant donné que cela fait baisser considérablement les scores d'accuracy et précision.
- La combinaison de LoRAS et SMOTE entraine une perte de de recall, et aisni donc de F1 score par rapport à leurs utilisations séparées.
- La combinaison d'IHT et SMOTE a premit d'obtenir le meilleur recall de tous les cas d'application au détriment de toutes les autres métriques. En revanche il suffit de remplacer IHT par OSS pour constater une nette baisse de performances par rapport au cas où elles sont appliquées séparément.
- Le rajout d'IHT à cette dernière combinaison a permis d'améliorer les scores de recall et f1 score mais pas d'accuracy et precision.
- La combinaison de LoRAS avec IHT est la pire possible, vu qu'elle réduit considérablement les scores de toutes les métriques. D'où l'on peut noter une importante incompatibilité entre les deux méthodes. En revanche, on remarque que c'est l'inverse en remplaçant IHT par OSS, surtout avec l'amélioration des scores de la combinaison de LoRAS et IHT en y rajoutant OSS.

- SVM :

Methods	Accuracy	Precision	Recall	F1_Score
IHT	73.44	60.38	87.67	71.51
OSS	77.6	75.86	60.27	67.18
SMOTE	75.52	65.85	73.97	69.68
LoRAS	67.71	54.62	89.04	67.71
IHT et OSS	73.44	60.38	87.67	71.51
LoRAS et SMOTE	73.96	63.86	72.6	67.95
IHT et SMOTE	73.44	60.38	87.67	71.51
OSS et SMOTE	70.83	60	69.86	64.56
IHT avec OSS et SMOTE	73.44	60.38	87.67	71.51
LoRAS et IHT	71.35	61.84	64.38	63.09
LoRAS et OSS	70.83	57.52	89.04	69.89
LoRAS avec IHT et OSS	71.35	61.84	64.38	63.09

Figure 6 : Mesures de performances des méthodes d'échantillonnage en utilisant un SVM sur le dataset PIDD

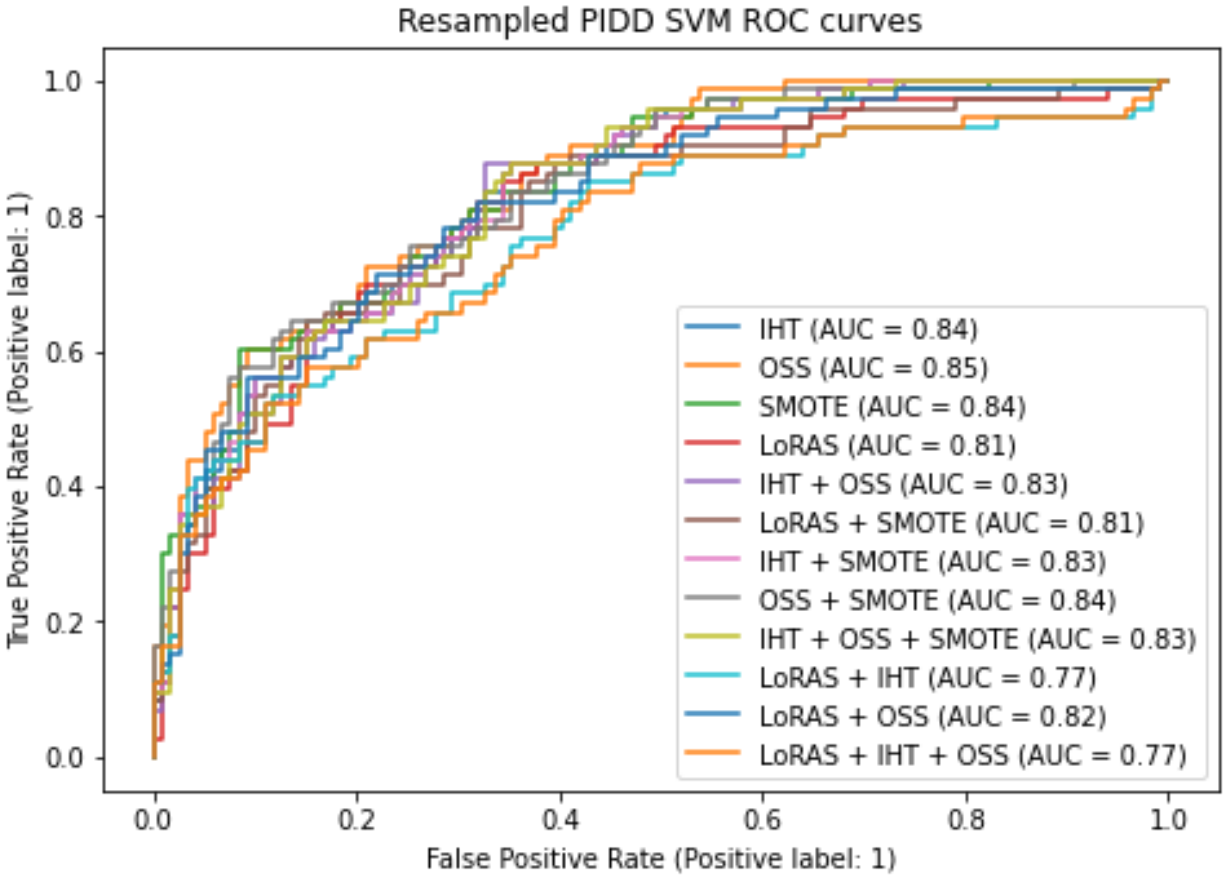


Figure 7 : Courbes ROC des modèles SVM en utilisant le dataset PIDD après échantillonnage

Dans ces résultats on note (voir plus bas pour plus de détails) :

- En termes d'accuracy et precision, OSS se place en premier.
- En termes de recall c'est LoRAS qui l'emporte.
- En termes de f1 score SMOTE est la meilleure des méthodes de base.
- Le rajout d'autres méthodes d'échantillonnage n'a eu aucune influence aux scores d'IHT.
- La combinaison de LoRAS avec SMOTE donne des résultats plus ou moins égaux à la moyenne des deux pour toutes les métriques.
- Le rajout de SMOTE à OSS a engendré un important déclin de scores sur toutes les mesures. Tandis que celui de LoRAS a rapporté les résultats aux alentours de la moyenne des deux pour les mesures accuracy et precision, gardé le recall d'OSS et offert une légère amélioration en f1 score.

- Parkinson :
  - MLP :

Methods	Accuracy	Precision	Recall	F1_Score
IHT	69.39	95.45	60	73.68
OSS	83.67	82.93	97.14	89.47
SMOTE	79.59	82.05	91.43	86.49
LoRAS	77.55	90	77.14	83.08
IHT et OSS	69.39	95.45	60	73.68
LoRAS et SMOTE	55.1	84.21	45.71	59.26
IHT et SMOTE	46.94	90.91	28.57	43.48
OSS et SMOTE	42.86	88.89	22.86	36.36
IHT avec OSS et SMOTE	69.39	95.45	60	73.68
LoRAS et IHT	71.43	81.82	77.14	79.41
LoRAS et OSS	67.35	88	62.86	73.33
LoRAS avec IHT et OSS	69.39	83.33	71.43	76.92

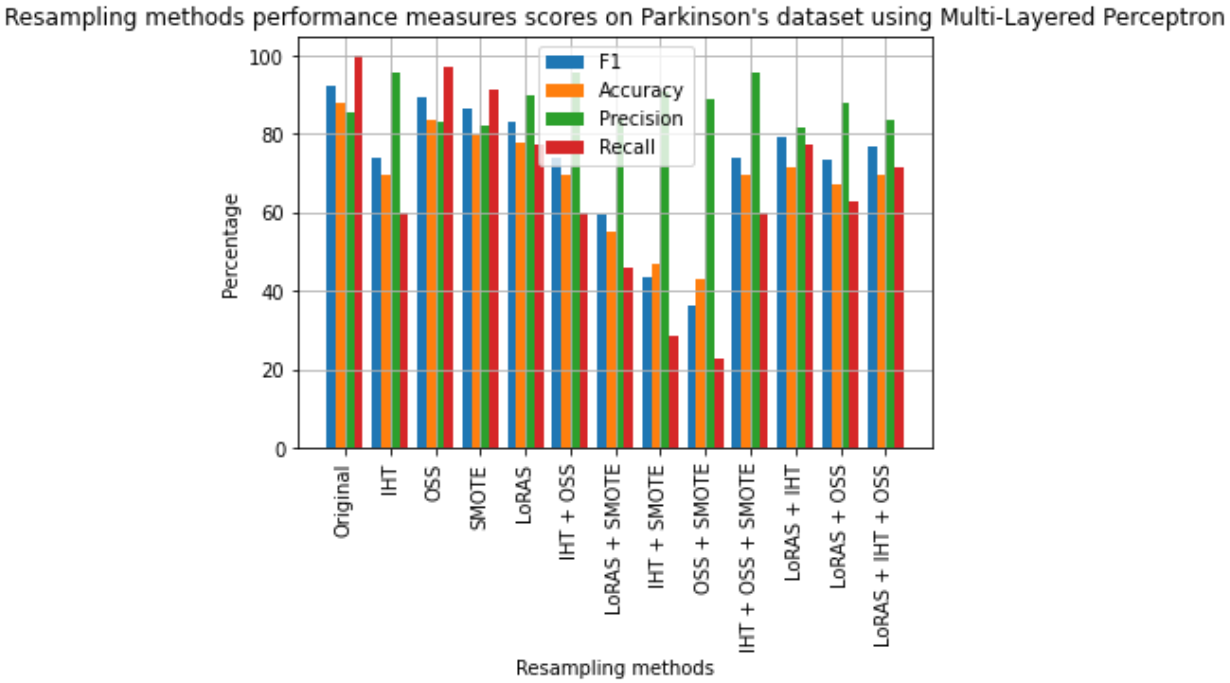


Figure 8 : Mesures de performances des méthodes d'échantillonnage en utilisant un MLP sur le dataset Parkinson

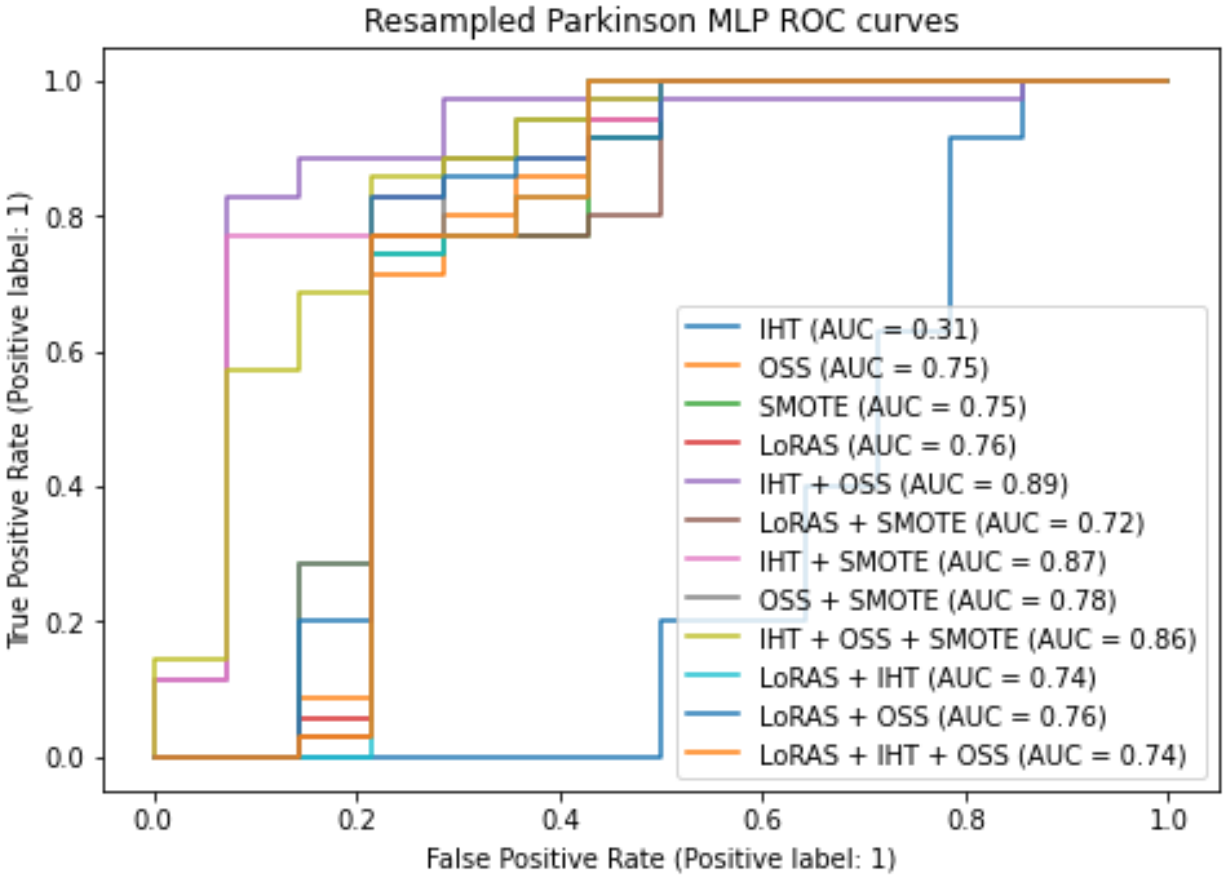


Figure 9 : Courbes ROC des modèles MLP en utilisant le dataset Parkinson après échantillonnage

Dans ces résultats on note (voir plus bas pour plus de détails) :

- Toutes les méthodes d'échantillonnage semblent nuire aux scores de toutes les métriques par rapport au dataset original à l'exception d'IHT qui rapporte une importante amélioration en termes de precision.
- Le rajout d'autres méthodes d'échantillonnage n'a eu aucune influence aux scores d'IHT.
- La combinaison de LoRAS avec SMOTE donne des résultats pires du cas où ils sont appliqués séparément pour toutes les métriques à l'exception de precision où le score est plus ou moins égal à la moyenne des deux.
- Le rajout d'OSS à SMOTE fait chuter violemment les scores de toutes les métriques plus que dans le cas où les méthodes sont appliquées séparément, sauf pour precision où l'on note une importante amélioration. Tandis que sont rajout à LoRAS engendre la même chose à l'exception de precision qui reste entre les valeures des deux méthodes séparées.

- SVM :

Methods	Accuracy	Precision	Recall	F1_Score
IHT	71.43	86.21	71.43	78.13
OSS	81.63	79.55	100	88.61
SMOTE	69.39	81.25	74.29	77.61
LoRAS	26.53	33.33	2.86	5.26
IHT et OSS	71.43	86.21	71.43	78.13
LoRAS et SMOTE	69.39	81.25	74.29	77.61
IHT et SMOTE	71.43	86.21	71.43	78.13
OSS et SMOTE	71.43	81.82	77.14	79.41
IHT avec OSS et SMOTE	71.43	86.21	71.43	78.13
LoRAS et IHT	69.39	81.25	74.29	77.61
LoRAS et OSS	65.31	80	68.57	73.85
LoRAS avec IHT et OSS	77.55	83.33	85.71	84.51

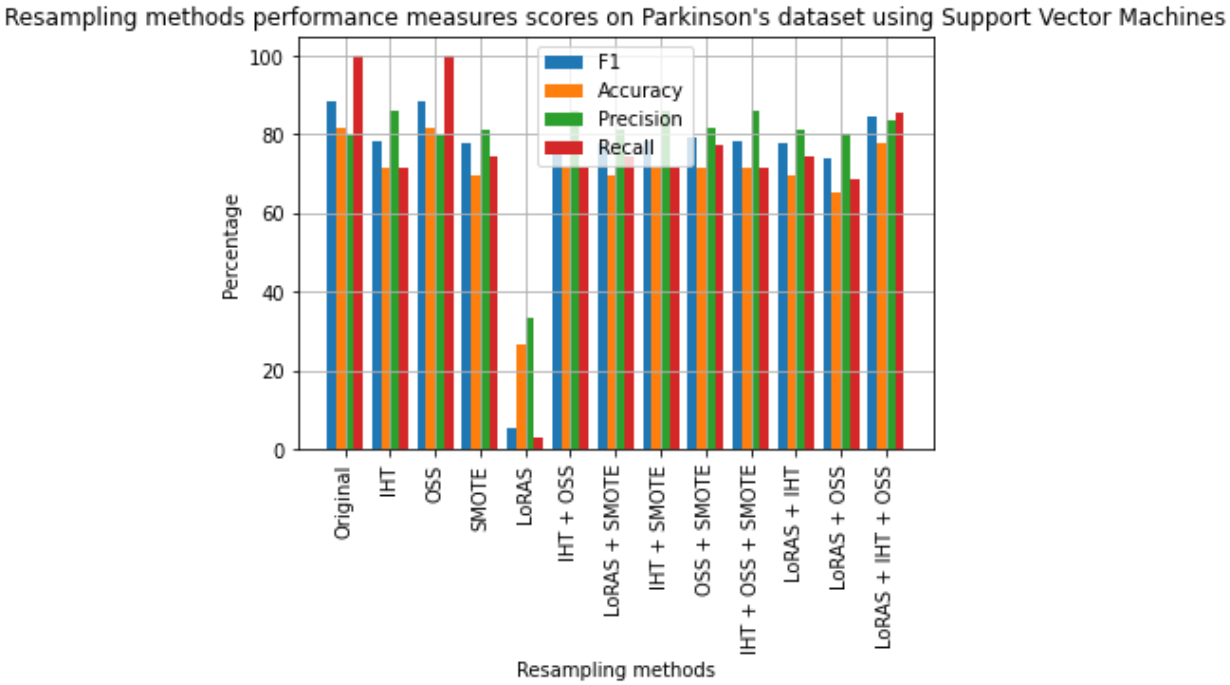


Figure 10 : Mesures de performances des méthodes d'échantillonnage en utilisant un SVM sur le dataset Parkinson

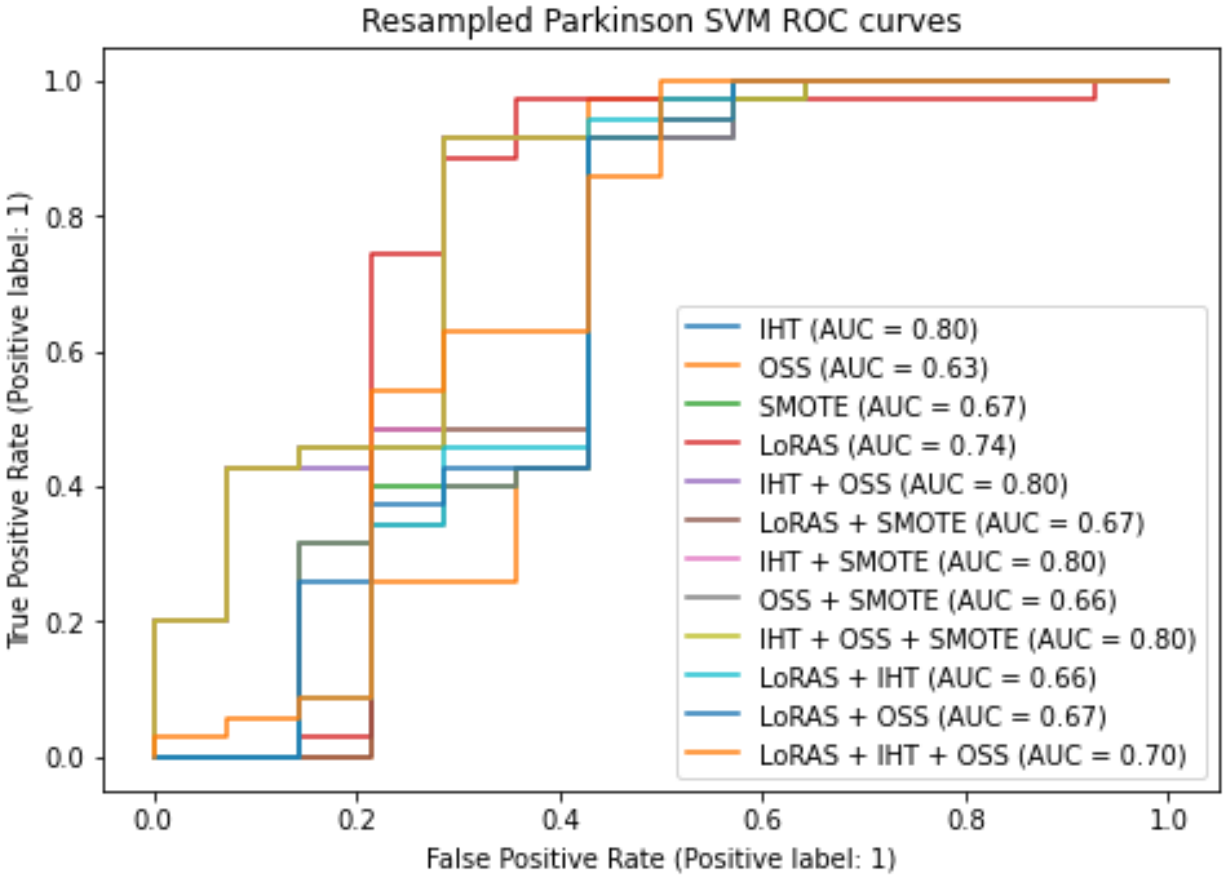


Figure 11 : Courbes ROC des modèles SVM en utilisant le dataset Parkinson après échantillonnage

Dans ces résultats on note (voir plus bas pour plus de détails) :

- Toutes les méthodes d'échantillonnage semblent nuire aux scores de toutes les métriques par rapport au dataset original à l'exception d'IHT qui rapporte une importante amélioration en termes de precision.
- Le rajout d'autres méthodes d'échantillonnage n'a eu aucune influence aux scores d'IHT à l'exception de LoRAS ainsi qu'OSS lorsqu'il y est combiné (LoRAS + IHT + OSS) où l'on note une importante réduction de scores en termes d'accuracy, precision et f1 score avec une certaine amélioration de recall dans le premier cas (LoRAS + IHT), et une amélioration du score des toutes les métriques à part precision qui se voit réduit pour ce qui est du deuxième (LoRAS + IHT + OSS).
- Le rajout de LoRAS n'a eu aucune influence aux scores de SMOTE.
- La combinaison d'OSS avec SMOTE donne des résultats plus ou moins égaux à la moyenne des deux pour toutes les métriques à l'exception de precision où l'on note une amélioration par rapport aux cas d'applications séparées.
- La combinaison de LoRAS avec OSS engendre des résultats compris aux alentours des moyennes des deux à l'exception de f1 score où le score est plus proche de celui d'OSS ainsi que precision où l'on note une amélioration par rapport au cas séparés.

Tous les résultats illustrent les points suivants :

- La performance de chacune des **méthodes** utilisées est liée à l'effectif du **dataset** : un nombre **plus important** d'instances offre un meilleur **apprentissage** du **modèle** de classification (effectif dataset **PIDD** > effectif dataset **parkinson**) et à l'inverse, un nombre **réduit** peut y **nuire**.
- Les **modèles** de classification ont chacun une **affinité relative** à la **nature des classes** : dans le cas de cette expérimentation, le modèle **svm** est plus efficace en matière de classification **binaire** (2 classes).
- L'application d'algorithmes d'échantillonnages entraine de façon générale :
    - L'orientation de **concentration** du modèle de classification vers la classe d'intérêt (**classe positive**), résultant en un **« recall »** plus important (une meilleure capacité à retrouver toutes les instances **positives**).
    - Baisse de **« accuracy »**, naturellement due à la réduction de **concentration** du modèle sur la **classe négative**, engendrant un déclin de sa classification.
    - Baisse de **« precision »** en utilisant le dataset **pidd**, et inversement avec celui de **parkinson**, montrant qu'un nombre accru de **caractéristiques** permet une **discrimination** plus fine de la classe **positive** par rapport celle **négative**.
    - Inévitablement, toute variance dans les mesures « **precision** » et « **recall** » en entrainent une dans « **f1 score** » qui est une **moyenne** pondérée des deux. Cette dernière donne une vue plus globale quant au pouvoir discriminatif du modèle de classification.
- **OSS** et **IHT** permettent de maintenir un « **recall** » considérable par rapport au dataset **original** (**IHT** pour un **effectif élevé** et **OSS** pour un **réduit**). Il en est de l'inverse pour ce qui est de « **precision** ».
- Un dataset à **effectif réduit** nuit considérablement aux performances de **LoRAS** avec utilisation d'un **svm**. Un **mlp** permet de remédier au problème.
- **SMOTE** permet en général **d'harmoniser** toutes les mesures en les **rapprochant** l'une de l'autre, qu'il soit appliqué **seul** ou **hybridé** avec d'autres algorithmes d'échantillonnage, même s'il réduit aussi la mesure « **accuracy** » étant donné qu'il fait pencher le modèle de classification en faveur de la **classe minoritaire** (**classe positive**). En revanche, il ne fait **qu'accroitre** considérablement la mesure « **precision** » et **décroitre** « **recall** » de même lors de son application sur le dataset **parkinson** (**effectif réduit**) avec un modèle **mlp**, d'où il est possible de tirer une **similarité** entre **SMOTE** et **LoRAS** quand à l'influence de **l'effectif** du dataset et du **modèle** de classification choisi.
- **L'hybridation** d'algorithmes de **sous** et **sur-échantillonnage** a une influence plus importante sur un dataset à **effectif réduit**.

La valeur de AUC varie entre [0, 1] et généralement supérieure à 0,5, Un excellent modèle a une AUC proche de 1, ce qui signifie qu'il a une bonne mesure de séparabilité. Un mauvais modèle a une AUC proche de 0, ce qui signifie qu'il a la pire mesure de séparabilité. En fait, cela signifie que le résultat est réciproque. Il prédit les 0 comme des 1 et les 1 comme des 0. Et lorsque l'AUC est de 0,5, cela signifie que le modèle n'a aucune capacité de séparation de classe. Lorsque l'AUC est de 0,7, cela signifie qu'il y a 70 % de chances que le modèle soit capable de faire la distinction entre la classe 0 et la classe 1.

Dans notre étude et en se référant aux résultats de la figure 3 avant échantillonnage une interprétation est donnée selon le changement des paramètres après échantillonnage (le type du modèle, datasets, et méthodes d'échantillonnage) :

- Dans la figure 5 on remarque que l'échantillonnage a amélioré la valeur de AUC =0.73 et surtout avec des hybridations des approches citant « IHT+OSS+SMOTE » et « IHT+OSS » avec AUC =0.77et AUC=0.79 respectivement. Les approches IHT, OSS, SMOTE et LORAS appliquées séparément n'ont pas apporté un plus, c'est-à-dire que la valeur de AUC =0.73 n'a pas changé. Par contre les approches « LoRAS+IHT » et « LoRAS+HT+OSS » ont donné des valeurs inefficaces et insignifiantes (AUC=0.56).
- Dans la figure 7 on remarque que l'échantillonnage n'a pas amélioré la valeur de AUC=0.86 où presque toutes les hybridations ont une valeur AUC se rapprochant de 0.86. Cela signifie que le modèle SVM est performant et capable de distinguer entre la classe positive et négative à 86% de chance.
- Dans la figure 9 on remarque que l'échantillonnage a amélioré la valeur de AUC =0.73 et surtout avec des hybridations des approches citant « IHT+OSS+SMOTE », « IHT+SMOTE » et « IHT+OSS » avec AUC =0.86, AUC=0.89 et AUC= 0.89 respectivement. Les approches OSS, SMOTE, LORAS appliquées séparément n'ont pas apporté un plus c'est-à-dire que la valeur de AUC est proche de 0.73 sauf l'approche HT qui a donné une valeur inefficace.
- Dans la figure 11 on remarque que l'échantillonnage a amélioré la valeur de AUC =0.64 avec presque dans toutes les approches et leurs hybridations et surtout avec « IHT+OSS+SMOTE », « IHT+SMOTE » et « IHT+OSS » avec AUC = 0.80 partout.

Conclusion et perspectives

En commençant par expliquer brièvement le principe de classification, des données déséquilibrées et quelques méthodes respectives traitant ces deux concepts, ce projet a offert une vue d'ensemble sur les différentes méthodes d'échantillonnages utilisées pour remédier au problème par le biais d'une étude comparative recensant les scores des mesures de performance pour chacune d'entre elles.

L'étude fut réalisée sur 2 datasets médicaux (PIDD et parkinson) étant données la sensibilité du domaine.

L'interprétation des résultats a permis de soutirer des informations montrant le comportement et l'influence de chacun des modèles de classification et méthodes d'échantillonnage par rapport aux données utilisées.

En ce qui concerne les perspectives, il peut être intéressant de considérer l'utilisation d'autres datasets (notamment ceux développés dans les labos de l'université), modèles de classification ou méthodes de balancement comme les méthodes ensemblistes pour voir la différence et éventuellement généraliser les résultats.

De même, la fusion des modèles pourrait s'avérer utile compte tenu du fait qu'elle permet la complétude entre ces derniers et de là l'obtention de meilleurs résultats.

Bibliographie

[CA02] Chawla et al., SMOTE: Synthetic Minority Over-sampling Technique, SMOTE, page 329, 2002.

[CCV14] Silvia Cateni, Valentina Colla, and Marco Vannucci. A method for resampling imbalanced datasets in binary classification tasks for real-world problems. Neurocomputing, 135 :32–41, 2014.

[F11] Francisco Fernández-Navarro and al. 'A dynamic oversampling procedure based on sensitivity for multi-class problems'. Pattern Recognition. Volume 44, Issue 8, August 2011, Pages 1821-1833.

[GHCH11] Ming Gao, Xia Hong, Sheng Chen, and Chris J Harris. A combined smote and pso based rbf classifier for two-class imbalanced problems. Neurocomputing, 74(17) :3456–3466, 2011.

[GSM12] Vicente García, Javier Salvador Sánchez, and Ramón Alberto Mollineda. On the effectiveness of preprocessing methods when dealing with different levels of class imbalance. KnowledgeBased Systems, 25(1) :13–21, 2012.

[H13] Hualong Yu and al. 'ACOSampling: An ant colony optimization-based undersampling method for classifying imbalanced DNA microarray data'. Neurocomputing. Volume 101, 4 February 2013, Pages 309-318.

[HADK20A] Hellal Amel et Dorsaf Kabtani, Conception et réalisation d'un système de classification pour le diagnostic du diabète, Description détaillée des techniques de classification, Introduction, page 17, juin 2020.

[HADK20B] Hellal Amel et Dorsaf Kabtani, Conception et réalisation d'un système de classification pour le diagnostic du diabète, Description détaillée des techniques de classification, Machines à vecteurs de support (SVM), page 20, juin 2020.

[HCI11] Hamza Cherif Ikram, 'Classification des tracés CardioTocoGraphiques (CTG) d'un fœtus à l'aide de classifieurs multiples'. Thèse soutenue à l'Université Abou Bakr Belkaid de Tlemcen, 2011.

[J09] Jason Van Hulse and al. 'Knowledge discovery from imbalanced and noisy data'. Data & Knowledge Engineering. Volume 68, Issue 12, December 2009, Pages 1513-1542.

[LYCL10] CY Lee, MR Yang, LY Chang, and ZJ Lee. A hybrid algorithm applied to classify unbalanced data. In Networked Computing and Advanced Information Management (NCM), 2010 Sixth International Conference on, pages 618–621. IEEE, 2010.

[M08] Mazurowskia and al. ‘Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance’. Neural Networks. Volume 21, Issues 2–3, March–April 2008, Pages 427- 436.

[RD13] M Mostafizur Rahman and DN Davis. Addressing the class imbalance problem in medical datasets. International Journal of Machine Learning and Computing, 3(2) :224–228, 2013.

[S09] Show-Jane Yen and Yue-Shi Lee. ‘Cluster-based undersampling approaches for imbalanced data distributions’. Expert Systems with Applications. Volume 36, Issue 3, April 2009, Pages 5718–5727.

[SBNDA20] Saptarshi Bej, Narek Davtyan et al., LoRAS: An oversampling approach for imbalanced datasets, LoRAS: localized randomized afne shadowsampling, page 8, 2020.

[Y12] Yang Yong and al. ‘The Research of Imbalanced Data Set of Sample Sampling Method Based on K-Means Cluster and Genetic Algorithm’. Energy Procedia. Volume 17, Part A, 2012, Pages 164-170.

[ZH15] Zahra Hadji, Apprentissage des données en distribution déséquilibrée par les méthodes d’ensemble, Introduction générale, page 1, juin 2015.

Webographie

[WEB1] https://machinelearningmastery.com/undersampling-algorithms-for-imbalanced-classification/ 02/07/2021

[WEB2] https://towardsdatascience.com/instance-hardness-threshold-an-undersampling-method-to-tackle-imbalanced-classification-problems-6d80f91f0581 02/07/2021

[WEB3] https://analyticsindiamag.com/hands-on-guide-to-loras-a-better-oversampling-algorithm/ 02/07/2021