REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEURE ET DE LA RECHERCHE SCIENTIFIQUE

Université ABDERAHMANE MIRA – BEJAIA

Faculté des Sciences Exactes

Département Informatique

Mémoire de Master en Informatique

Spécialité : Systèmes d'Information Avancés

Thème

Proposition d'une approche hybride de sélection de caractéristiques dans l'apprentissage automatique

Présenté par :

ZADI Ali ZAIDI Halim

Soutenu devant le jury composé de :

Présidente	Mme KESSIRA	MAA	U. A/Mira Béjaïa
Examinateur	M. OUZEGGANE	MAA	U. A/Mira Béjaïa
Encadrante	Mme ALOUI	MCA	U. A/Mira Béjaïa
Co-Encadrant	M. ATTOUMI	DOCTORANT	U. A/Mira Béjaïa

Promotion : 2023-2024

Remerciements

On dit souvent que le trajet est aussi important que la destination. Les cinq années que nous avons passées à l'université nous ont permis de bien comprendre la signification de cette phrase toute simple. Ce parcours, en effet, ne s'est pas réalisé sans défis et sans soulever de nombreuses questions pour lesquelles les réponses ont nécessité de longues heures de travail.

Avant tout, nous tenons à remercier Dieu, le tout-puissant, d'avoir accordé la volonté, la patience et surtout la santé durant toutes nos années d'études. Que sa guidance nous accompagne tout au long de notre vie future.

Nous tenons à remercier nos encadrants, Mme ALOUI et M ATTOUMI de nous avoir supervisées durant notre projet de fin d'étude, pour leur orientations, leur précieux conseils et encouragements qui nous ont permis de mener à bien ce travail.

Nous souhaitons également exprimer notre gratitude à nos parents et à nos familles, qui ont su nous supporter et encourager tout au long de notre vie, ainsi que pour leur patience, leur soutien et leur aide inestimable.

Nous tenons à exprimer également notre gratitude aux membres de jury pour avoir accepté d'examiner et de juger notre travail. Enfin, nous remercions tous ceux et celles qui, de près ou de loin, ont contribué à l'aboutissement de ce mémoire.

Dédicaces

Je dédie ce mémoire à mes chers parents, qui ont été mes piliers tout au long de ce parcours. À ma mère, dont la force et le dévouement ont été une source constante d'inspiration, et à mon père, dont le soutien inconditionnel m'a donné la confiance nécessaire pour avancer.

À mes frères et a ma sœur, complices de mes joies et compagnons de mes peines, je vous adresse toute ma gratitude.

Enfin, à mes amis, qui ont su être présents à chaque étape, Je leur adresse toute ma gratitude. Ce travail est le fruit de leur collaboration et de leur amitié.

Dédicace Halim

Dédicace

Je dédie ce travail à mes parents, pour leur amour inconditionnel, leur soutien indéfectible et leurs sacrifices qui ont rendu tout cela possible.

À ma sœur Lydia, pour son aide précieuse, ses encouragements constants, et son soutien inébranlable.

À son mari Yannis, pour son soutien et ses conseils avisés.

Merci de croire en moi et de m'accompagner dans chaque étape de ce voyage.

Dédicace Ali

Table des matières

Table des figures

Liste des tableaux

Liste des acronymes

DM Data Mining

KNN k-nearest neighbor

SVM Support Vector Machine

DT Decision Tree

NB Naïve Bayes

RF Random forest

Hz Hertz

ICA Independent Component Analysis

PCA Principal Component Analysis

AdaBoost Adaptive Boosting

NMIFS Normalized mutual information feature selectionl

MIFS Mutual Information Feature Selection

MIFS-U Mutual Information Feature Selection with Unsupervised Learning

mRMR Minimum Redundancy Maximum Relevance

SFS Sequential forward selection

ImpCHI Improved Chisauqre

FAST Fast clustering-based feature selection

MST Minimum spanning tree

ReliefF Relief Feature Selection

CMIM Conditional Mutual Information Maximization

LS Laplacian Score

RCD redundancy-complementariness dispersion

IterativeLS Iterative Laplacian score

SFS Sequential forward selection

FFNN Feedforward Neural Network

SBS Sequential backward selection

TS Tabu Search

AMMLP Adaptive Multilayer Multimodal Perceptron

ML Machine Learning

PCA Principal Component Analysis

OPF Optimum-Path Forest

GA Genetic algorithm

BBA Bibary bat algorithm

WGI Weighted gini index

ROC-AUC Receiver Operating Characteristic Area Under the Curve

UCI University of California Irvine

HGFS Hybrid genetic feature selection

RFE Recursive Feature Elimination

FHFSSA flexible hybrid floating sequential search algorithm

SFBS Sequential Floating Backward Selection

HGA Hybrid genetic algorithm

HAR Human Activity Recognition

Introduction générale

De nos jours, la croissance des bases de données a entraîné une augmentation exponentielle des données récoltées. Ces données comportent généralement un grand nombre de variables et/ou d'instances. Il est donc nécessaire de réduire la dimension des données en sélectionnant les caractéristiques les plus pertinentes pour le problème étudié. Cependant, les méthodes d'analyse, d'apprentissage automatique ou de fouille de données classiques peuvent s'avérer inefficaces ou produire des résultats inexacts.

La sélection de caractéristiques consiste à choisir un sous-ensemble de variables pertinentes parmi l'ensemble global des caractéristiques. Cette problématique peut s'appliquer à diverses tâches de fouille de données et englobe des méthodes qui permettent de sélectionner un sous-ensemble de variables initiales en utilisant différents critères et techniques [4].

Dans le cadre de ce projet, nous nous concentrons sur la sélection de caractéristiques en classification supervisée, qui vise à déterminer la relation entre un ensemble de variables explicatives et une variable cible (la classe), basée sur un nombre fini d'individus.

La sélection de caractéristiques présente plusieurs avantages liés à la réduction de la quantité de données. D'une part, cette réduction facilite la gestion des données et d'autre part, elle permet une meilleure compréhension des résultats produits par un système basé sur ces caractéristiques. Par exemple, dans un problème de classification, ce processus de sélection réduit non seulement le temps d'apprentissage, mais améliore également la compréhension des résultats fournis par le classificateur et peut parfois augmenter la précision de la classification en favorisant les caractéristiques les moins bruitées.

Les contributions de ce mémoire sont les suivantes :

1. Proposition d'une méthode hybride de sélection de caractéristique.
2. Réalisation d'un état de l'art sur des méthodes de sélection de caractéristiques.
3. Étude comparative entre plusieurs sous ensemble de caractéristiques selectionnés.

Le reste de ce mémoire est organisé comme suit : Le Chapitre I donne des généralités sur la sélection de caractéristiques et le machine learning, le Chapitre II expose notre état de l'art sur des méthodes de la littérature qui proposent de la sélection de caractéristiques. Dans le Chapitre III, Nous détaillons l'approche proposée, que nous expérimentons dans le Chapitre IV.

Chapitre I

Généralités sur l'apprentissage automatique et la sélection de caractéristiques

I.1 Introduction

Dans ce chapitre, nous allons explorer en profondeur le concept de la sélection de caractéristiques. Ce processus crucial dans l'apprentissage automatique consiste à identifier et à retenir les variables les plus importantes parmi un ensemble de données, dans le but d'améliorer la performance des modèles et de réduire leur complexité.

Pour commencer, nous définirons de manière générale ce qu'est la sélection de caractéristiques. Ensuite, nous présenterons les concepts fondamentaux de la fouille de données et de l'apprentissage automatique . Nous expliquerons les types d'algorithmes utilisés et leur fonctionnement, ce qui situera la sélection de caractéristiques dans un contexte plus large et montrera son utilité dans ces domaines. Enfin nous nous concentrerons spécifiquement sur la sélection de caractéristiques. Nous expliquerons pourquoi elle est nécessaire et comment elle peut améliorer les performances des modèles d'apprentissage automatique.
En résumé, ce chapitre fournira une vue d'ensemble complète et détaillée de la sélection de caractéristiques, en couvrant à la fois les théories sous-jacentes et les applications pratiques, afin de donner au lecteur une compréhension approfondie de ce domaine essentiel en apprentissage automatique.

I.2 Quelques définitions

Pour mieux comprendre les concepts abordés dans ce chapitre, il est essentiel de définir certains termes clés. Voici quelques définitions qui seront utiles tout au long de notre discussion :

I.2.1 Sélection

La sélection est un processus volontaire et méthodique, parfois inconscient ou automatique, par lequel certains éléments (personnes ou objets) sont choisis en fonction de caractéristiques spécifiques pour atteindre un objectif précis. Dans le domaine de la sélection de caractéristiques, ce processus consiste à identifier les variables les plus pertinentes et significatives parmi un ensemble données [11].

I.2.2 Caractéristique

Une caractéristique, dans le contexte de l'apprentissage automatique et de la fouille de données, est une propriété ou un attribut mesurable des données utilisé pour la modélisation et l'analyse. Parmi d'autres, les caractéristiques sont les variables ou les paramètres que l'on utilise pour entraîner des modèles prédictifs. Elles peuvent être des mesures quantitatives ou qualitatives qui décrivent les différents aspects des objets ou des événements étudiés [12].

I.2.2.1 Pertinence d'une caractéristique

La performance d'un algorithme d'apprentissage est grandement influencée par les caractéristiques choisies pour l'apprentissage. La présence de caractéristiques redondantes ou non pertinentes peut diminuer cette performance. Dans la littérature, on trouve plusieurs définitions de ce qui constitue la pertinence d'une caractéristique. Une variable pertinente est une variable telle que sa suppression entraîne une détérioration des performances du système d'apprentissage
Selon [12] une variable peut être très pertinente, peu pertinente et non pertinente.

- **Très pertinente :** Une variable f_i est considérée comme très pertinente si son absence entraîne une détérioration significative des performances du système de classification utilisé.
- **Peu pertinente :** Une variable f_i est considérée comme peu pertinente si elle n'est pas *très pertinente* et s'il existe un ensemble de caractéristiques $\mathbf{V}$ tel que la performance de $\mathbf{V} \cup \{f_i\}$ soit significativement meilleure que celle de $\mathbf{V}$.

— **Non pertinente :** Les variables qui ne sont ni *très pertinentes* ni *peu pertinentes* sont considérées comme non pertinentes. Ces variables seront généralement éliminées de l'ensemble initial de variables.

I.2.3 Sélection de caractéristiques

La sélection de caractéristiques est une technique qui permet de choisir les caractéristiques, variables ou mesures les plus intéressantes, pertinentes ou informants, d'un système donné pour la réalisation de la tâche pour laquelle il a été conçu. Cette phase est généralement un module important d'un système complexe. Les domaines d'application des techniques de sélection de caractéristiques sont variés par exemple la modélisation, la classification, l'apprentissage automatique (Machine Learning) et l'analyse exploratoire de données (Data Mining) [13].

I.3 Fouille de données

Data Mining (DM), ou exploration de données, est un ensemble de techniques et de méthodologies permettant d'analyser de grandes quantités de données pour extraire des informations utiles, des corrélations et des motifs cachés. Ce procédé s'inscrit dans le cadre de la Business Intelligence et vise à aider les entreprises à résoudre des problèmes, à réduire les risques et à identifier de nouvelles opportunités commerciales [14].

En français, ce processus est connu sous plusieurs appellations :

— Forage de données
— Extraction de connaissances à partir de données

I.3.1 Processus de la Fouille de Données

Le processus de fouille de données suit généralement plusieurs étapes clés, avec une attention particulière à la sélection de caractéristiques :

— **Collecte des Données** : Rassembler les données provenant de diverses sources, telles que les bases de données , les systèmes ERP.
— **Exploration des Données** : Utiliser des techniques de visualisation et des statistiques pour comprendre la structure des données et identifier des motifs intéressants.
— **Préparation des Données** : Nettoyer et transformer les données pour les rendre exploitables. Cela inclut le traitement des valeurs manquantes, la normalisation.

— **Sélection de Caractéristiques** : Utiliser des techniques telles que Recursive Feature Elimination (RFE), les méthodes basées sur l'importance des caractéristiques (comme les forêts aléatoires) ou les méthodes basées sur des critères statistiques pour identifier les caractéristiques les plus pertinentes.
— **Modélisation** : Appliquer des algorithmes de data mining pour construire des modèles à partir des données préparées et des caractéristiques sélectionnées.
— **Évaluation** : Valider et évaluer la performance des modèles pour s'assurer qu'ils répondent aux objectifs fixés.
— **Déploiement** : Intégrer les modèles validés dans les processus opérationnels pour en tirer des bénéfices pratiques[14].

I.3.2 Applications de la Fouille de Données et de la Sélection de Caractéristiques

La fouille de données et la sélection de caractéristiques trouvent des applications dans divers domaines, notamment :

— **Marketing** : Personnalisation des campagnes publicitaires, analyse du comportement des clients, fidélisation.
— **Finance** : Détection des fraudes, gestion des risques, prévisions financières.
— **Santé** : Analyse des dossiers médicaux, prédiction des maladies, optimisation des traitements.
— **Commerce de détail** : Analyse des ventes, gestion des stocks, recommandations de produits[14].

I.4 Apprentissage Automatique (Machine Learning)

L'apprentissage automatique est une branche de l'intelligence artificielle, permet aux systèmes de s'améliorer en apprenant de l'expérience sans programmation explicite. Selon Tom Mitchell [15], un système "apprend" lorsqu'il montre une amélioration mesurable de sa capacité à exécuter une tâche spécifique T grâce à l'expérience E, évaluée par une performance P. Les tâches peuvent varier de simples classifications à des actions complexes comme la conduite autonome. L'expérience provient de données qui peuvent être historiques, simulées ou interactives, et la performance est mesurée par des métriques comme la précision ou la perte.

Le but de l'apprentissage automatique est d'équiper les systèmes artificiels de capacités d'apprentissage similaires à celles des êtres humains. Il vise à développer des algorithmes

capables d'identifier des modèles à partir d'un vaste ensemble de données, souvent désigné comme une base de données d'apprentissage. Le résultat de ce processus est un modèle, appelé classifieur, qui utilise les informations apprises pour catégoriser de nouveaux exemples inconnus. Ces catégories peuvent être numériques, comme dans les cas de régression, ou catégorielles, comme dans les cas de classification.

I.4.1 Types d'apprentissage Automatique

Il existe plusieurs méthodes pour enseigner à une machine comment apprendre à partir de données. Chaque technique présente des spécificités qui la rendent plus ou moins adaptée en fonction des données disponibles et des objectifs visés par l'apprentissage [16]. Nous mentionnerons ici les trois approches les plus pertinentes : l'apprentissage supervisé, l'apprentissage non supervisé, et l'apprentissage semi-supervisé.

I.4.1.1 Apprentissage supervisé

Dans l'apprentissage supervisé, on utilise des ensemble de données (instances) X qui sont étiquetées avec les réponses correctes Y correspondant à chaque cas. L'objectif pour l'algorithme est de prédire correctement les réponses pour chaque nouvelle entrée en se basant sur ces données d'apprentissage. Il s'efforce de générer une prédiction Y' qui doit être aussi proche que possible de la valeur réelle Y, minimisant ainsi l'erreur de prédiction.

Pour atteindre cet objectif, une fonction de prédiction f est optimisée de manière à ce que les prédictions Y' s'approchent autant que possible avec les valeurs cibles Y. Cette fonction f est cruciale car elle est ensuite utilisée pour faire des prédictions sur de nouvelles données non vues durant la phase d'entraînement. L'optimisation de cette fonction peut se faire par plusieurs méthodes, l'une des plus courantes est «la descente de gradient » . Le processus de descente de gradient, par exemple, ajuste itérativement les paramètres de la fonction f en calculant les dérivées partielles pour réduire l'erreur [16].

L'apprentissage supervisé se divise en deux catégories principales : la classification et la régression, ces méthodes sont adaptées en fonction de la nature du problème et des données à disposition, permettant ainsi une modélisation précise et efficace.

- **Définition de la Classification** La classification est une technique où le modèle est entraîné à partir d'un ensemble de données étiquetées pour attribuer une étiquette de classe discrète à des entrées inconnues. Le but est de prédire des catégories distinctes, telles que oui ou non, spam ou non spam, basées sur des observations antérieures.
- **Définition de la Régression**

La régression implique la prédiction d'une quantité continue. Contrairement à la classification qui catégorise les données, la régression est utilisée pour prédire une réponse numérique continue, comme le prix d'une maison ou les températures futures, à partir de variables prédicatives[16].

Ayant exploré les principes fondamentaux de l'apprentissage supervisé et ses deux grandes catégories,il est maintenant essentiel de comprendre comment ces concepts sont mis en œuvre dans des algorithmes spécifiques :

1. **K plus proches voisins**

   La méthode k-nearest neighbor (KNN) repose sur une comparaison directe entre le vecteur de caractéristiques de l'entité à classer et les vecteurs de caractéristiques des entités de référence. Cette comparaison se fait par le calcul des distances entre ces entités. L'entité à classer est ensuite attribuée à la classe la plus fréquente parmi les k entités les plus proches selon la distance utilisée [17].

   Considérons $\mathbf{X}_p = (x_{p1}, x_{p2}, \ldots, x_{pN})$ comme le vecteur de caractéristiques de l'entité p, où N est le nombre de caractéristiques, et p et q sont deux entités à comparer. Les distances couramment utilisées par les classificateurs KNN sont les suivantes :

   — **Distance Euclidienne** :

   $$D(\mathbf{X}_p, \mathbf{X}_q) = \sqrt{\sum_{i=1}^{N} (x_{pi} - x_{qi})^2} \tag{1.1}$$

   — **Distance de Manhattan** :

   $$D(\mathbf{X}_p, \mathbf{X}_q) = \sum_{i=1}^{N} |x_{pi} - x_{qi}| \tag{1.2}$$

   — **Distance de Minkowski** :

   $$D(\mathbf{X}_p, \mathbf{X}_q) = \left(\sum_{i=1}^{N} |x_{pi} - x_{qi}|^r \right)^{1/r} \tag{1.3}$$

   — **Distance de Tchebychev** :

   $$D(\mathbf{X}_p, \mathbf{X}_q) = \max_{i=1,\ldots,N} |x_{pi} - x_{qi}| \tag{1.4}$$

   Dans la figure I-1, à gauche, la classification est directe quel que soit le nombre de voisins sélectionnés : le nouvel objet est classé comme noir. À droite, cependant,

le résultat dépend du nombre de voisins choisis et de la stratégie de classification utilisée. Pour k = 1, le nouvel objet est classé comme gris. Pour k = 3, si les trois voisins sont considérés avec un poids égal, alors le nouvel objet sera noir. Cependant, si les poids sont ajustés en fonction de l'inverse de la distance, le nouvel objet pourrait être classé comme gris. Cela implique que la classe est déterminée en pondérant l'influence de chaque voisin par sa distance : plus un voisin est éloigné, moins il influence la classification.

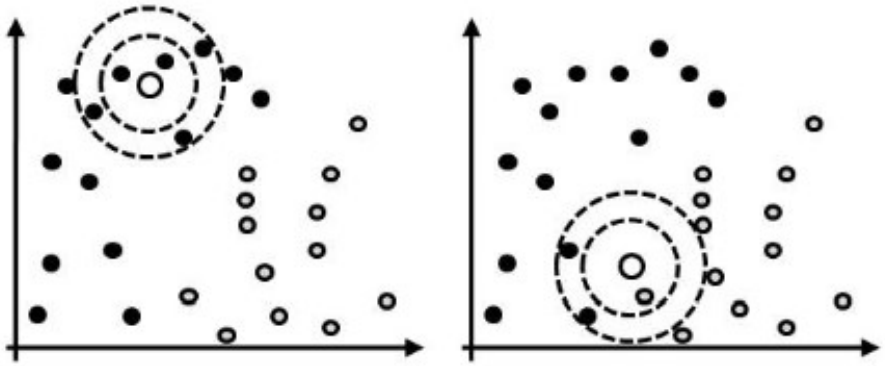

FIGURE I-1 – Exemple de classification avec les Knn [1]

2. **Réseau Neuronaux**

Un réseau de neurone est un algorithme d'apprentissage, proposé la première fois par Mac Cullogh et Pitts en 1943 [18] Inspirés de la structure neurophysiologique du cerveau humain. Un neurone formel est l'unité fondamentale de ces systèmes modélisés. Lorsqu'il reçoit des signaux d'autres neurones, un neurone formel réagit en générant un signal de sortie qui est ensuite transmis à d'autres neurones du réseau. Le signal reçu est une somme pondérée des signaux provenant de différents neurones. La sortie du neurone est alors calculée en appliquant une fonction d'activation à cette somme pondérée[2]

$$y_j = f\left(\sum_{i=1}^{N} w_{ij} x_i\right)$$

où y_j représente la sortie du neurone j, x_i (pour $i = 1 \dots N$) sont les signaux reçus par le neurone j des neurones i, et w_{ij} sont les poids des connexions entre les neurones i et j. En fonction de l'application, la fonction f, appelée fonction d'activation, peut être une fonction identité, sigmoïde, tangente hyperbolique ou une fonction linéaire par morceaux.[18]

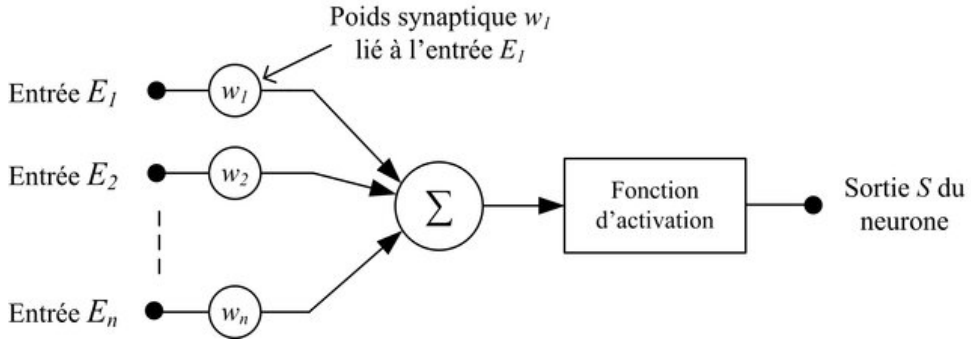

FIGURE I-2 – Schéma de la structure d'un neurone artificiel[2]

3. **Machines à Vecteurs de Support (SVM)**

   L'algorithme des machines à vecteurs de support a été conçu dans les années 1990 par le scientifique russe Vladimir Vapnik[19]. À l'origine, Support Vector Machine (SVM) étaient développés comme un algorithme de classification supervisée binaire. Leur efficacité est particulièrement notable grâce à leur capacité à gérer des problèmes avec un grand nombre de descripteurs, à fournir une solution unique (évitant les problèmes de minima locaux souvent rencontrés avec les réseaux de neurones), et à obtenir de bons résultats sur des cas concrets.[20]

   Dans sa forme initiale, l'algorithme cherche à établir une frontière de décision linéaire entre deux classes. Cependant, ce modèle peut être largement amélioré en projetant les données dans un espace de dimension supérieure, augmentant ainsi leur séparabilité. Le même algorithme peut alors être appliqué dans ce nouvel espace, aboutissant à une frontière de décision non linéaire dans l'espace d'origine.[21]

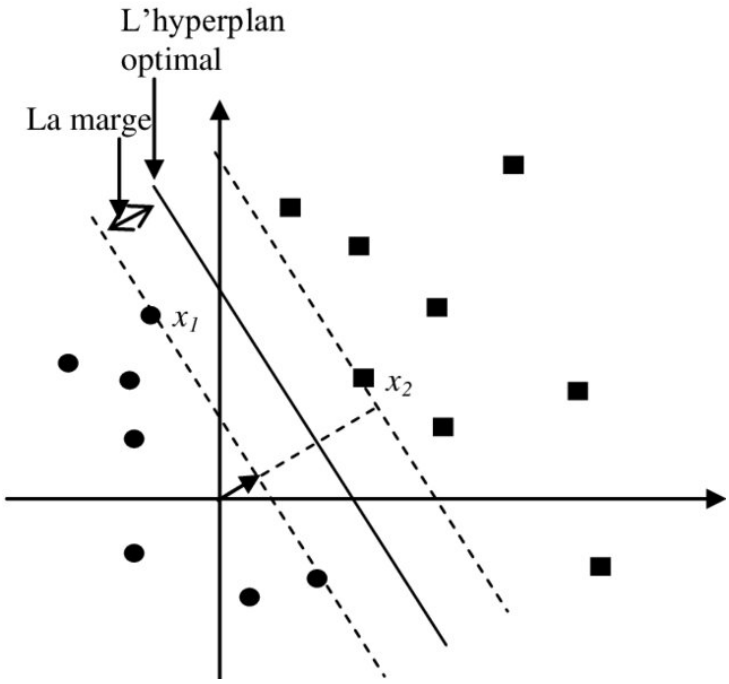

FIGURE I-3 – Exemple d'un hyperplan optimal séparant deux classes [3]

I.4.1.2 Apprentissage non supervisé

Dans l'apprentissage non supervisé, les données fournies à l'algorithme ne comportent pas d'étiquettes, ce qui oblige le système à identifier par lui-même les similitudes ou les motifs dans les données présentées. Cela se révèle extrêmement utile étant donné que les données non étiquetées sont souvent plus abondantes que les données étiquetées.
L'objectif principal de cette méthode peut varier de la simple identification de modèles cachés à des objectifs plus complexes comme la découverte automatique de représentations nécessaires à la classification des données brutes [22].
Les applications de l'apprentissage non supervisé incluent souvent l'analyse de données complexes, où il peut être difficile pour un observateur humain de détecter des patterns sans assistance. Par exemple, en appliquant l'apprentissage non supervisé à la reconnaissance des activités humaines, il est possible d'analyser les données issues de capteurs pour découvrir des modèles de mouvement. Ces données, collectées à partir de smartphones ou de dispositifs portables, peuvent révéler des corrélations entre les séquences de mouvement et des activités spécifiques, telles que marcher, courir ou monter des escaliers, sans que ces activités soient explicitement étiquetées. De plus, en l'absence de réponses prédéfinies, les méthodes non supervisées sont capables de traiter et d'organiser des ensembles de données complexes et volumineux de manière significative. Cette technique est souvent employée dans la détection d'anomalies, telles que les fraudes à la carte de crédit, ou pour alimenter des systèmes de recommandation qui suggèrent des produits à acheter. Dans le cas des images, par exemple, des photos non étiquetées de chiens peuvent

être analysées par un algorithme pour regrouper les images similaires, facilitant ainsi la classification sans intervention humaine préalable [23].
Après avoir défini les principes de base de l'apprentissage non supervisé, il est essentiel de se pencher sur les outils qui permettent de mettre ces principes en pratique. Parmi les algorithmes les plus utilisés en apprentissage non supervisé, nous trouvons :

1. **K-moyennes**

   L'algorithme **k-means** est une méthode populaire de partitionnement non supervisé, introduite par MacQueen en 1967, et qui vise à diviser un ensemble de données en k groupes ou clusters distincts. L'objectif est de minimiser la variance intra-cluster tout en maximisant la variance inter-clusters. Le principe de fonctionnement de l'algorithme est le suivant [24] :

   Définition du nombre de clusters (k) : L'utilisateur spécifie à l'avance le nombre de clusters souhaité, k.
   Initialisation des centres des clusters : Les centres des clusters sont initialisés aléatoirement à partir des points de données ou par des méthodes comme *k-means++* pour une meilleure initialisation.
   Assignation des points : Chaque point de données est affecté au cluster dont le centre est le plus proche, en utilisant une mesure de distance (généralement la distance euclidienne).
   Mise à jour des centres : Une fois tous les points affectés, les centres des clusters sont recalculés en prenant la moyenne des points assignés à chaque cluster.
   Répétition du processus : Les étapes d'assignation et de mise à jour des centres sont répétées jusqu'à convergence, c'est-à-dire jusqu'à ce que les centres des clusters ne changent plus ou que les points ne changent plus de cluster.

2. **Analyse en composantes indépendantes** C'est une technique statistique permettant de révéler les facteurs cachés qui sous-tendent des ensembles de variables aléatoires, de mesures ou de signaux. Independent Component Analysis (ICA) définit un modèle génératif pour les données multivariées observées, qui est généralement donné sous la forme d'une grande base de données d'échantillons. Dans le modèle, les variables de données sont supposées être des mélanges linéaires de certaines variables latentes inconnues, et le système de mélange est également inconnu. Les variables la tentes sont supposées non gaussiennes et mutuellement indépendantes et sont appelées des composants indépendants des données observées. L'ACI est liée à Principal Component Analysis (PCA), mais c'est une tech-

nique beaucoup plus puissante qui est capable de trouver les facteurs sous-jacents des sources lorsque ces méthodes classiques échouent complètement. Ses applications comprennent les images numériques, les bases de documents, les indicateurs automatiques et les mesures psychométriques .

3. **Classification hiérarchique** Les techniques de classification hiérarchique utilisent des itérations basées sur des critères de similarité pour regrouper des ensembles d'éléments en classes cohérentes. Un exemple courant est la méthode de classification ascendante, qui commence par la création d'une matrice de similarité entre toutes les paires d'objets. À chaque étape de l'itération, les deux clusters les plus similaires sont fusionnés en fonction des données de cette matrice. Après chaque fusion, la matrice est mise à jour pour refléter la similarité entre le nouveau cluster formé et les autres clusters restants. Ce processus se répète jusqu'à ce que tous les éléments soient regroupés en un seul cluster. La performance de cette méthode dépend fortement de la métrique de similarité choisie. Un domaine d'application de cette méthode est la recherche d'images, où elle peut être utilisée pour organiser et retrouver des images basées sur leur contenu visuel.

I.4.1.3 Apprentissage semi-supervisé

L'apprentissage semi-supervisé est une méthode intermédiaire entre l'apprentissage supervisé, qui utilise des données étiquetées, et l'apprentissage non supervisé, qui n'utilise que des données non étiquetées. Dans l'apprentissage semi-supervisé, on combine des données étiquetées avec une plus grande quantité de données non étiquetées pour effectuer des tâches telles que la classification ou la régression. Un domaine d'application typique est la reconnaissance d'images, où collecter de nombreuses images est relativement simple(sur internet par exemple), mais les étiqueter peut être coûteux ou prendre beaucoup de temps. Donc utiliser un algorithme qui peut exploiter à la fois les données étiquetées et non étiquetées permet d'améliorer l'efficacité de l'apprentissage.[25]

I.5 Réduction de la dimensionnalité

La réduction de la dimensionnalité d'un ensemble de données devient de plus en plus indispensable en raison de la croissance exponentielle des volumes de données. Dans de nombreux domaines, la résolution de problèmes repose sur un ensemble de caractéristiques (variables). L'augmentation du nombre de ces variables, qui modélisent le problème, pose plusieurs défis, tels que la complexité accrue, l'allongement du temps de calcul et la dégradation des performances du système en présence de données bruitées. Une méthode efficace pour réduire la dimensionnalité consiste à sélectionner les caractéristiques les

plus pertinentes parmi l'ensemble des données. Ces méthodes de réduction se classent généralement en deux catégories :

- **Réduction basée sur la sélection de caractéristiques :** Cette approche consiste à sélectionner les caractéristiques les plus pertinentes parmi l'ensemble de données initiales décrivant le phénomène étudié.
- **Réduction basée sur la transformation des données :** Également appelée extraction de caractéristiques, cette méthode consiste à remplacer l'ensemble initial de données par un nouvel ensemble réduit, construit à partir des caractéristiques initiales.

Dans ce mémoire, nous présenterons les méthodes de réduction par sélection de caractéristiques. Nous détaillerons le processus de sélection, en expliquant les différentes techniques utilisées et leurs critères de sélection. Nous discuterons également des avantages et des inconvénients de ces méthodes, en mettant en lumière les améliorations qu'elles apportent en termes de performance et de précision du modèle

I.5.1 Réduction basée sur une sélection de caractéristiques

La sélection de caractéristiques est généralement définie comme un processus de recherche visant à identifier un sous-ensemble "pertinent" de caractéristiques parmi l'ensemble initial. La pertinence de ce sous-ensemble dépend des objectifs et des critères du système. En général, le problème de sélection de caractéristiques peut être formulé ainsi :

Soit $F = \{f_1, f_2, \ldots, f_N\}$ un ensemble de caractéristiques de taille N, où N représente le nombre total de caractéristiques étudiées. Soit Ev une fonction d'évaluation des sous-ensembles de caractéristiques. On suppose que la valeur maximale de Ev correspond au meilleur sous-ensemble de caractéristiques. L'objectif de la sélection est de trouver un sous-ensemble $F' \subseteq F$ de taille N' ($N' \leq N$ tel que :

$$Ev(F_0) = \max_{Z \subseteq F} Ev(Z) \tag{2.1}$$

où $|Z| = N_0$, et N_0 est soit un nombre défini par l'utilisateur, soit déterminé dynamiquement par une méthode de génération de sous-ensembles.

I.5.2 Processus général de la sélection de caractéristiques

La méthode consiste à générer successivement des sous-ensembles optimaux de caractéristiques pertinentes en éliminant de manière séquentielle, à chaque itération, les carac-

téristiques moins pertinentes.

— Quantifier l'importance des sous-ensembles de caractéristiques en utilisant des mesures de pertinence entre les différentes caractéristiques.
— Cependant, étant donné que le nombre optimal de caractéristiques N_0 n'est généralement pas connu *a priori*, il est nécessaire de définir un critère d'arrêt pour terminer la procédure de sélection une fois que le sous-ensemble optimal F' a été identifié. [4]

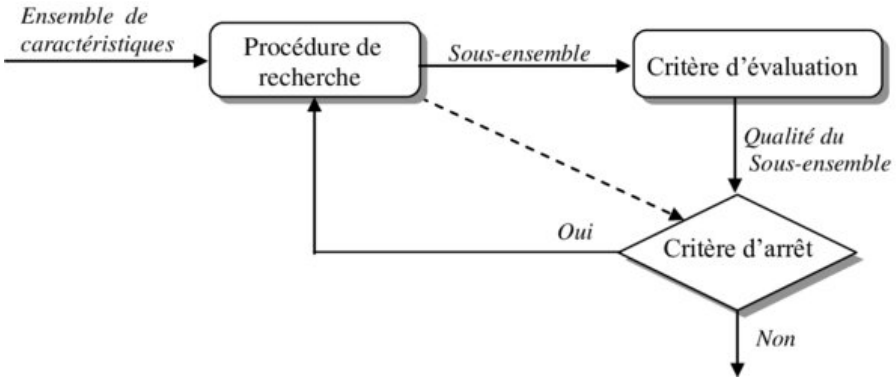

FIGURE I-4 – Processus générale d'un algorithme de sélection de caractéristiques[4]

— **Techniques de Sélection de Caractéristiques**

Les techniques de sélection de caractéristiques permettent d'identifier les variables les plus pertinentes pour améliorer la performance des modèles prédictifs. Elles sont essentielles pour réduire la complexité des modèles et éviter le surapprentissage.

I.5.3 Méthodes de sélection de caractéristiques Supervisées

Ces méthodes utilisent la variable cible (par exemple, supprime les variables non pertinentes), ils se composent par trois méthodes principales : la méthode filtrante 'filter', la méthode enveloppante 'wrapper' et la méthode intégrée 'embedded'.

I.5.3.1 Méthode Filter (Filtrage)

Pour les modèles de filtre, les caractéristiques sont sélectionnées en fonction des caractéristiques des données sans utiliser les algorithmes d'apprentissage. Le modèle filtre utilise la métrique sélectionnée pour identifier les attributs non pertinents et filtrer les colonnes redondantes du modèle. Il choisit une seule mesure statistique qui convient à donner au modèle un score pour chaque colonne de caractéristique. Les colonnes sont renvoyées et classées en fonction de leurs scores. En choisissant les bonnes caractéristiques, il permet

potentiellement d'améliorer la précision et l'efficacité de la classification. Il n'utilise généralement que les colonnes avec les meilleurs scores pour créer le modèle prédictif. Les colonnes avec des scores mauvais peuvent être laissées dans le dataset et ignorées lorsque vous créez un modèle[26]

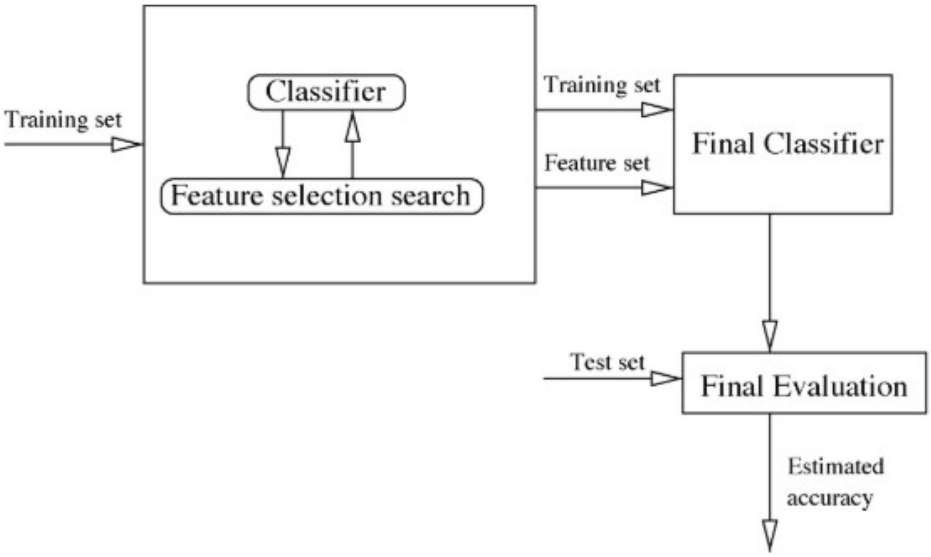

FIGURE I-5 – Le principe général d'une méthode de sélection de type Filter[5]

I.5.3.2 Méthode Enveloppe (Wrapper)

Dans les méthodes Wrapper, le processus de sélection des caractéristiques est basé sur un algorithme d'apprentissage automatique spécifique qui essaie d'adapter le dataset. Il suit approche de recherche Gourmand (en. Greedy), en évaluant toutes les combinaisons possibles de caractéristiques par rapport au critère d'évaluation (en. 'Evaluation Criterion'). Le critère d'évaluation est simplement la mesure de la performance qui dépend du type de problème. Enfin, il sélectionne la combinaison de caractéristiques qui donne les résultats optimaux pour l'algorithme d'apprentissage automatique spécifié[27]

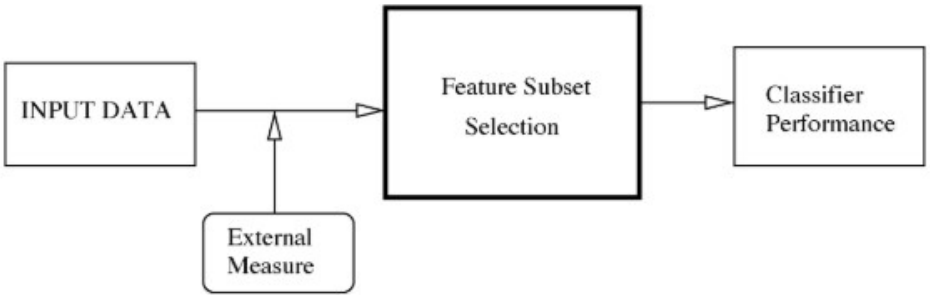

FIGURE I-6 – Le principe général d'une méthode de sélection de type wrapper[5]

I.5.3.3 Méthode Integrée/Hybride

a la différence des méthodes 'Wrapper' et 'Filter', les méthodes 'Integrée' (appelées aussi méthodes Imbriquées) incorporent la sélection des caractéristiques lors du processus d'apprentissage. Un tel mécanisme intégré pour la sélections des caractéristiques peut être trouvé. Dans les méthodes de sélection de type 'Imbriquées', la base d'apprentissage est divisée en deux parties, une base d'apprentissage et une base de validation pour valider le sous-ensemble de caractéristiques sélectionné. En revanche, les méthodes intégrées peuvent se servir de tous les exemples d'apprentissage pour établir le système. Cela constitue un avantage qui peut améliorer les résultats. Un autre avantage de ces méthodes est leur plus grande rapidité par rapport aux approches 'Wrapper' parce qu'elles évitent que le classificateur recommence de zéro pour chaque sous-ensemble de caractéristiques[28] a la différence des méthodes 'Wrapper' et 'Filter', les méthodes 'Integrée' (appelées aussi méthodes Embedded) incorporent la sélection des caractéristiques lors du processus d'apprentissage. Un tel mécanisme intégré pour la sélections des caractéristiques peut être trouvé, par exemple, dans les algorithmes de type SVM, Adaptive Boosting (AdaBoost), ou dans les arbres de décisions. Dans les méthodes de sélection de type 'Embedded', la base d'apprentissage est divisée en deux parties, une base d'apprentissage et une base de validation pour valider le sous-ensemble de caractéristiques sélectionné. En revanche, les méthodes intégrées peuvent se servir de tous les exemples d'apprentissage pour établir le système. Cela constitue un avantage qui peut améliorer les résultats. Un autre avantage de ces méthodes est leur plus grande rapidité par rapport aux approches 'Wrapper' parce qu'elles évitent que le classificateur recommence de zéro pour chaque sous-ensemble de caractéristiques[28]

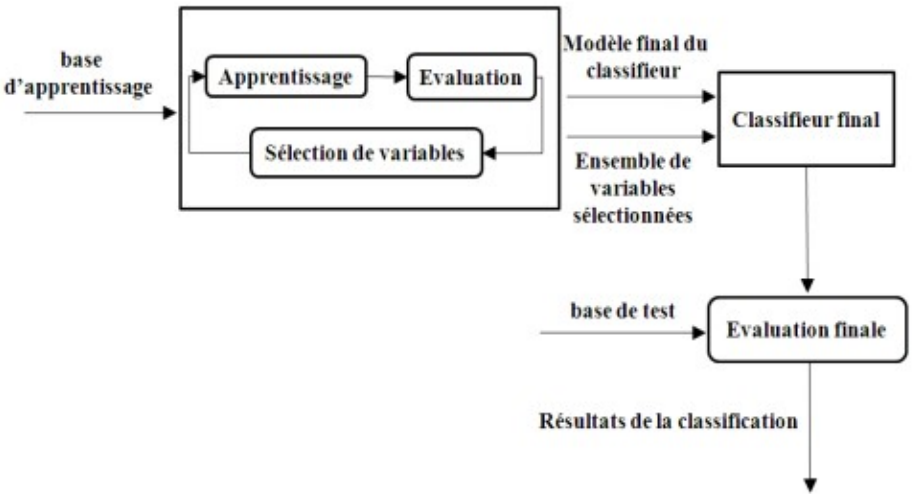

FIGURE I-7 – Le principe général d'une méthode de sélection de type Embedded[5]

I.5.4 Méthodes de sélection de caractéristiques non-Supervisées

Les méthodes non-supervisé sont généralement utilisés pour les tâches de regroupement, ils sont très similaires à la sélection de caractéristiques supervisée, sauf qu'il n'y a aucune information d'étiquette impliquée dans la phase de sélection de caractéristiques et la phase d'apprentissage du modèle. Sans l'étiquette pour définir la pertinence de la caractéristique, la sélection de caractéristiques non supervisée repose sur un autre critère alternatif pendant la phase de sélection de caractéristiques. Un critère couramment utilisé choisit les caractéristiques qui peuvent le mieux préserver la structure multiple des données d'origine.

I.6 Conclusion

La sélection de caractéristiques est une méthode cruciale pour le pré-traitement et la réduction de la dimensionnalité des ensembles de données, jouant un rôle central dans le succès des projets d'exploration de données et d'apprentissage automatique. Ce processus vise à élaborer des modèles plus épurés et accessibles, à améliorer l'efficacité de l'analyse de données et à faciliter la préparation, le nettoyage et la compréhension des données. La sélection de caractéristiques est également un domaine de recherche prolifique et pertinent, avec des applications pratiques significatives dans divers secteurs tels que les statistiques, la reconnaissance de motifs, l'apprentissage automatique et l'analyse de données.

Ce chapitre a offert une vue d'ensemble structurée et approfondie sur la sélection des caractéristiques en lien avec l'apprentissage automatique.

Chapitre II

État de l'art des méthodes de sélection de caractéristiques

II.1 Introduction

L'état de l'art, également connu sous le nom de "state of the art", est une étape fondamentale dans la rédaction d'un mémoire ou d'une thèse. Il s'agit d'un panorama synthétique et organisé des travaux déjà réalisés sur un sujet spécifique, englobant une analyse des publications majeures en relation avec le thème choisi. Dans le domaine de la sélection de caractéristiques en machine learning, l'état de l'art est crucial car il permet de situer la recherche dans son contexte, de justifier la problématique, et d'assurer sa qualité et sa pertinence.

Dans ce chapitre, nous allons explorer l'état actuel de la recherche sur la sélection de caractéristiques en apprentissage automatique. La sélection de caractéristiques joue un rôle crucial dans la construction de modèles d'apprentissage automatique efficaces et précis. Elle permet non seulement de réduire la dimensionnalité des données, mais aussi d'améliorer les performances des modèles et de fournir une meilleure interprétabilité.

Nous commencerons par examiner les différentes techniques de sélection de caractéristiques utilisées en apprentissage automatique, nous passons en revue les algorithmes actuels en mettant l'accent sur leurs avantages, leurs inconvénients et leurs applications typiques. Ensuite, nous comparerons ces techniques en fonction de divers critères tels que la précision, la complexité et le temps d'exécution. Nous discuterons également des applications réelles de la sélection de caractéristiques dans divers domaines.

Enfin, nous aborderons les défis actuels dans le domaine de la sélection de caractéris-

tiques et les directions possibles pour les travaux futurs. Ce chapitre vise à fournir une vue d'ensemble complète et à jour de la sélection de caractéristiques en apprentissage automatique, en mettant en évidence les domaines qui nécessitent davantage de recherche.

II.2 Travaux connexes

Plusieurs études similaires ont été menées dans le domaine de la sélection de caractéristiques. Au cours de ce projet, nous avons eu l'opportunité d'explorer divers travaux de recherche liés à notre sujet. Nous présentons ci-dessous un aperçu des différentes études dans ce contexte, basées sur les techniques précédemment définies, ces travaux se sont concentrés sur trois méthodes principales :

II.2.1 Méthodes de filtrage

Les méthodes de filtrage, offrent un moyen simple et rapide de sélectionner les caractéristiques. Ils sont efficaces pour les grands ensembles de données, fournissant une bonne vue d'ensemble initiale de l'importance des caractéristiques. Cependant, ils peuvent négliger des relations complexes entre les entités [29].
On peut la divisé en deux comme suit :

Univariée -> Score de Fisher ,Gain d'information mutuelle, Variance etc
Multi-variée -> Correlation de Pearson

Les méthodes de filtrage univariées sont le type de méthodes dans lesquelles les caractéristiques individuelles sont classées selon des critères spécifiques. Les N principales caractéristiques sont ensuite sélectionnées. Différents types de critères de classement sont utilisés pour les méthodes de filtrage univariées, par exemple le score de Fisher, les informations mutuelles et la variance de la caractéristique.

Les méthodes de filtrage multivariées sont capables de supprimer les caractéristiques redondantes des données car elles prennent en compte la relation mutuelle entre les caractéristiques.

II.2.1.1 Sélection normalisée de caractéristiques d'information mutuelle (Normalized mutual information feature selection)

Information mutulle (Mutual information)
En théorie de l'information, information mutuelle I (X ; Y) est le degré d'incertitude dans X dû à la connaissance de Y [30].

Estevez et al, ont proposés une méthode de sélection de caractéristiques basée sur des informations mutuelle nommée Normalized mutual information feature selectionl (NMIFS) [31] comme mesure de la pertinence et de la redondance entre les caractéristiques. Les performances de l'algorithme NMIFS ont été comparées avec les résultats de Mutual Information Feature Selection (MIFS), Mutual Information Feature Selection with Unsupervised Learning (MIFS-U) et Minimum Redundancy Maximum Relevance (mRMR) sur quatre ensembles de données : ensemble de données synthétiques uniformes d'hypercube, vague de Breiman ensemble de données de formulaire, ensemble de données de base de spam et ensemble de données de sonar. La méthode NMIFS a surpassé les méthodes MIFS, MIFS-U et mRMR sur plusieurs ensembles de données artificielles et problèmes de référence, à l'exception de l'ensemble de données de Breiman où NMIFS et MIFS-U ont donné des résultats similaires. Cette observation indique que bien que NMIFS offre généralement une meilleure performance, certains ensembles de données spécifiques peuvent ne pas montrer un avantage significatif par rapport à d'autres méthodes avancées comme MIFS-U.

II.2.1.2 Chi-carré Amélioré (ImpCHI)

Chi-square (chi carré) Le test du chi carré, également écrit X2 test, est tout test d'hypothèse statistique où la distribution d'échantillonnage de la statistique de test est une distribution du chi-carré, il mesure la dépendance entre les variables stochastiques. L'utilisation de cette fonction élimine donc les caractéristiques les plus susceptibles d'être indépendantes de la classe et donc non pertinentes pour la classification.
La formule de chi-carré est defini comme suit :

$$\chi_c^2 = \sum \frac{(O_i - E_i)^2}{E_i} [32]$$

ou :

- c = degrés de liberté
- O = valeur(s) observée(s)
- E = valeur(s) attendue(s)

En 2016, Bahassine et al ont proposé une méthode améliorée de classification de textes arabes qui utilise la sélection de caractéristiques du chi carré appelée ci-après Improved Chisauqre (ImpCHI) pour améliorer les performances de classification [33] . En outre, ils ont également comparé ce chi carré amélioré avec trois mesures traditionnelles de sélection de caractéristiques, à savoir l'information mutuelle, le gain d'information et le chi carré en utilisant le classificateur SVM. En termes de performances, la meilleure mesure f obtenue pour ce modèle est de 90,50 %, lorsque le nombre de caractéristiques est de 900.

II.2.1.3 Sélection rapide de caractéristiques basée sur le clustering

L'algorithme Fast clustering-based feature selection (FAST) fonctionne en deux étapes. Dans la première étape, les caractéristiques sont divisées en clusters en utilisant des méthodes de regroupement de la théorie des graphes. Dans la deuxième étape, la caractéristique la plus représentative qui est fortement liée aux classes cibles est sélectionnée dans chaque cluster pour former un sous-ensemble de caractéristiques.

[34] Dans cette étude Song et al utilisent un algorithme pour la de sélection caractéristiques appelée algorithme de sélection de caractéristiques basé sur le clustering rapide (FAST) . L'algorithme utilise la technique graphique Minimum spanning tree (MST) pour regrouper les caractéristiques. Cet algorithme FAST avait effectivement supprimé les caractéristiques non pertinentes et caractéristiques redondantes en utilisant une mesure d'incertitude symétrique. Pour choisir les caractéristiques optimales, l'algorithme FAST utilise la méthode basée sur les clusters (cluster-based methods).

Arbre couvrant de poids minimal C'est un sous-ensemble des arêtes d'un graphe connecté et pondéré par les arêtes qui relie tous les sommets ensemble sans aucun cycle et avec le poids total d'arête minimum possible.

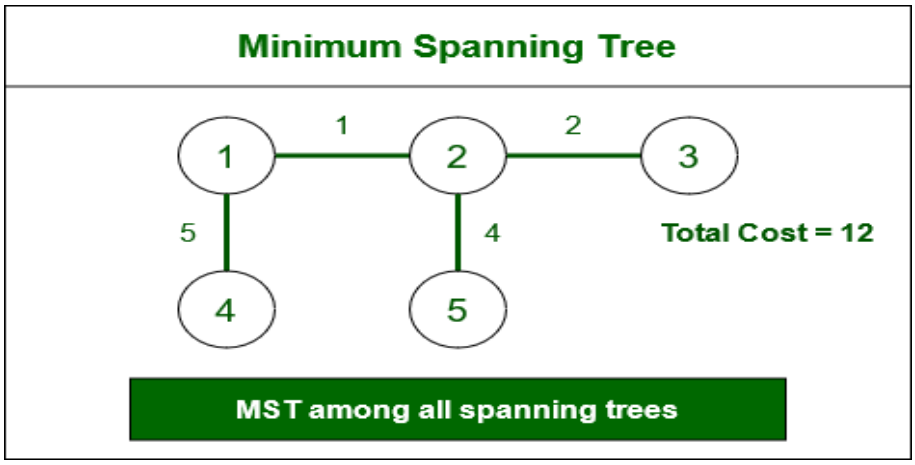

FIGURE II-1 – Arbre couvrant de poids minimal [6]

II.2.1.4 Dispersion redondance-complémentarité

Chen et al [7], ont développé une nouvelle approche de sélection de caractéristiques nommée redundancy-complementariness dispersion (RCD) utilisant l'inter corrélation d'ordre élevé. Pour illustrer l'efficacité de la méthode proposée, une étude comparative approfondie a été réalisée en la comparant à sept méthodes représentatives de sélection de caractéristiques, notamment mRMR,Relief Feature Selection (ReliefF) et Conditional Mutual Information Maximization (CMIM) et appliquées avec quatre classificateurs fréquemment

utilisés sur dix ensembles de données (datasets). De plus, les informations mutuelles ont également été utilisées pour la sélection de caractéristiques dans les problèmes de classification multi-étiquettes et les systèmes de détection d'intrusion.

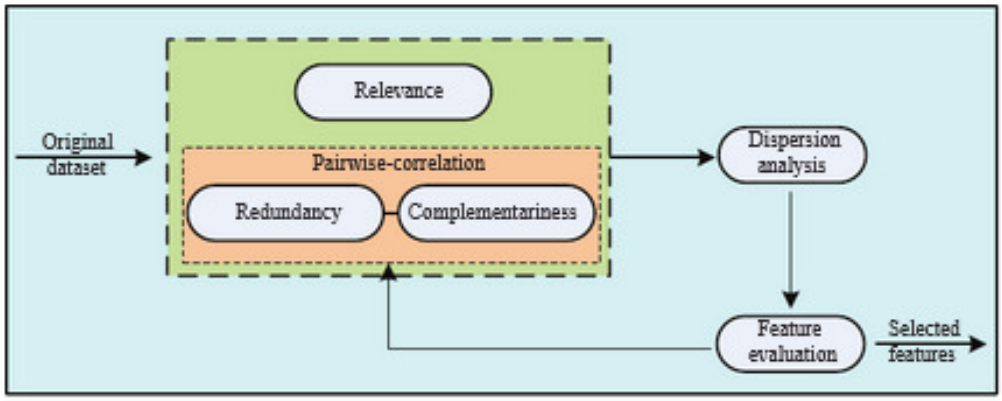

FIGURE II-2 – un nouveau cadre de sélection des carctéristiques [7]

II.2.1.5 Score laplacien iteratif

Le score Laplacien (LS) Le score laplacien est une méthode de sélection de caractéristiques qui peut être utilisé dans des scénarios supervisés ou non supervisés.

Dans cette étude [35] , Liu et al, proposent une méthode Laplacian Score (LS) améliorée appelée Iterative Laplacian score (IterativeLS). Cette méthode met à jour de manière itérative le graphique du quartier le plus proche pour évaluer l'importance d'une entité en fonction de sa capacité à préserver la localité. Contrairement à LS, le concept clé d'IterativeLS consiste à améliorer progressivement le graphe du voisin le plus proche en éliminant les caractéristiques les moins pertinentes à chaque itération. Les résultats expérimentaux sur plusieurs ensembles de données de grande dimension tels que les ensembles de données UCI et les ensembles de données de visage démontrent l'efficacité de la méthode proposée et la surperformance de l'algorithme d'origine sur les tâches de classification et de clustering.

II.2.2 Méthodes d'enveloppe

Les méthodes d'enveloppe sont basées sur un algorithme de classification qui permet d'évaluer les solutions potentielles (sous-ensembles de caractéristiques) générées par un algorithme de recherche, ce qui entraîne des coûts de calcul plus élevés. Bien que cela présente un désavantage, ils sont souvent plus efficaces et doivent être appliqués autant que possible.[36]

II.2.2.1 Sélection séquentielle avant

Marcano-Cedeño et al, ont présenté une méthode de sélection de caractéristiques basée sur Sequential forward selection (SFS) et Feedforward Neural Network (FFNN)[37] pour estimer l'erreur de prédiction comme critère de sélection Dans ce travail, les ensembles de données Iris, Wine et Breast Cancer Wisconsin ont été utilisés pour valider leurs méthode. Ces ensembles de données couvrent des données de faible, moyenne et haute dimension.Adaptive Multilayer Multimodal Perceptron (AMMLP) est une nouvelle méthode appliquée afin de classer ces données. Les résultats obtenus par SFS-FFNN avec AMMLP en termes de précision de classification étaient supérieurs à ceux obtenus par les autres algorithmes récents de sélection de caractéristiques en termes de rapidité et de précision.

II.2.2.2 Sélection séquentielle arrière

Dans cette étude récente de 2023 Aregbesola, Samuel Olamide et al [38], ont proposés un nouveau cadre pour déterminer l'ensemble optimal de caractéristiques requis pour prédire avec précision la déformation permanente des agrégats non liés. Pour éviter le surajustement, un grand ensemble de données est utilisé et les étapes de prétraitement des données sont soigneusement documentées. Initialement, 10 algorithmes Machine Learning (ML) différents sont appliqués pour prédire la déformation permanente et le modèle le plus performant est sélectionné en fonction de mesures, telles que l'erreur quadratique moyenne. Un algorithme Sequential backward selection (SBS) est ensuite associé au modèle choisi pour identifier les caractéristiques les plus pertinentes pour prédire la variable cible. De plus, l'ensemble de caractéristiques sélectionné est utilisé pour prédire la déformation permanente et comparé aux performances obtenues à partir de la réduction de dimensionnalité basée sur la PCA. Enfin, la précision de prédiction du modèle basé sur la sélection de caractéristiques est discutée sur la base des données expérimentales mesurées.

II.2.2.3 Algorithme génétique

Genetic algorithm (GA) , introduit par John Holland en 1975 [39], est un algorithme de recherche d'optimisation adaptative permettant de trouver une solution optimale inspiré de la sélection naturelle en biologie systèmes. Les gènes d'un organisme sont regroupés dans des structures appelées chromosomes, un ensemble de chromosomes est appelé population.

Les GA ont été appliqués avec succès à la sélection de caractéristiques [40]. Dans cet article, Alsukk et al, proposent une solution appelée diverse algorithme génétique a été

proposée qui aborde la question de la diversité des modifier les opérateurs de sélection et de croisement [41]. Le particulier la question de la diversité mérite davantage d'efforts de recherche en raison potentiel prometteur de GA dans la gestion du problème de sélection de caractéristiques. De plus, l'un des problèmes classiques lorsque la gestion des GA est leur lente convergence.
Dans ce contexte, Guha et al.[42] ont proposé un modèle innovant de sélection de caractéristiques, appelé CGA (Clonal Genetic Algorithm), pour la reconnaissance d'actions humaines visuelles. Publié dans la revue Neural Computing and Applications, cet article met en avant l'efficacité de l'algorithme CGA dans la réduction de la dimensionnalité des données tout en conservant les caractéristiques les plus discriminantes pour la classification des actions humaines. L'approche CGA est particulièrement utile dans les applications où la quantité de données est importante, car elle permet de traiter efficacement des ensembles de données volumineux tout en minimisant le risque de surajustement.

II.2.2.4 Algorithme de chauves souris binaires

Bibary bat algorithm (BBA) C'est une technique de sélection de caractéristiques inspirée de la nature et basée sur le comportement des chauves-souris.

Dans cette étude [43] , Nakamura et al, ont associer la capacité d'exploration des chauves-souris à la vitesse du classificateur Optimum-Path Forest (OPF) afin de déterminer les caractéristiques qui optimisent la précision dans un ensemble de tests. D'après des expériences réalisées sur cinq ensembles de données publiques, il ont été prouvé que l'approche suggérée peut être plus efficace que certaines techniques bien établies basées sur les swarm-based techniques.

II.2.2.5 Recherche tabou

Un algorithme d'optimisation métaheuristique appelé Tabu Search (TS) utilise une structure de mémoire pour orienter la recherche dans des zones inconnues. Fred W. Glover l'a introduit en 1986 et il est particulièrement bénéfique pour résoudre des problèmes d'optimisation combinatoire. TS associe l'étude locale à des structures de mémoire afin d'éviter de revenir sur les solutions déjà explorées et d'explorer de nouvelles zones de l'espace de recherche.

Maha et al, ont suggéré une étude Tabou avec une fenêtre temporelle pour améliorer l'optimisation des trajets à différents moments dans le domaine du tourisme urbain, dans le but d'améliorer la planification des itinéraires des véhicules dans le réseau routier du tourisme des villes. Selon les résultats, cet algorithme a la capacité d'améliorer le temps

passé sur la route par les clients du tourisme urbain et d'améliorer l'efficacité du tourisme urbain des clients[44].

II.2.3 Méthodes hybride

Cette approche est une combinaison de méthodes basées sur des filtres et des wrappers. L'approche par filtrage sélectionne un ensemble de caractéristiques candidats à partir de l'ensemble de caractéristiques d'origine et l'ensemble de caractéristiques candidats est affiné par l'approche wrapper. Il exploite les avantages de ces deux approches [45].

II.2.3.1 Coefficient de gini

Dans cet article, [46] une méthode de sélection de caractéristiques intégrées utilisant notre indice de Gini pondéré Weighted gini index (WGI) est proposée par Haoyue, Zhou et al. Ses résultats de comparaison avec les méthodes de sélection des caractéristiques du Chi-carré, de la statistique F et de l'indice de Gini montrent que la statistique F et le Chi carré atteignent les meilleures performances lorsque seules quelques caractéristiques sont sélectionnées. À mesure que le nombre de caractéristiques sélectionnées augmente, la méthode proposée a la plus grande probabilité d'obtenir les meilleures performances. L'aire sous une courbe caractéristique de fonctionnement du récepteur Receiver Operating Characteristic Area Under the Curve (ROC-AUC) et la mesure F sont utilisées comme critères d'évaluation. Les résultats expérimentaux avec deux ensembles de données montrent que les performances du ROC-AUC peuvent être élevées, même si seules quelques caractéristiques sont sélectionnées et utilisées, et ne changent que légèrement à mesure que de plus en plus de caractéristiques sont sélectionnées. Cependant, les performances de F-measure n'atteignent d'excellentes performances que si 20 % ou plus des caractéristiques sont choisies.

II.2.3.2 Algorithme de recherche séquentielle flottante hybride

Dans cet article, Hutchinson et al, [47] proposent un nouvel algorithme hybride intitulé flexible hybrid floating sequential search algorithm (FHFSSA), qui combine à la fois les principes de recherche par filtre et par l'enveloppe. Ils montrent qu'il est possible d'échanger une réduction significative du temps de recherche contre une diminution négligeable de la précision de la classification.

Les résultats expérimentaux sont rapportés sur deux ensembles de données, les données WAVEFORM) du référentiel University of California Irvine (UCI) et les données SPEECH de British Telecom.

Le principal avantage de l'hybridation de recherche flottante proposée est la possibilité de gérer de manière flexible le compromis entre la qualité du résultat et le temps de calcul.

II.2.3.3 Hybride Relief + K-means + SFBS

[48] Bins et Draper, ont développé une technique pour réduire un grand nombre de caractéristiques à un sous-ensemble plus petit sans compromettre l'importance des caractéristiques ou la précision de la classification. Leur algorithme comporte trois étapes : d'abord, les caractéristiques non pertinentes sont éliminées à l'aide d'une version modifiée de l'algorithme de relief. Ensuite, les caractéristiques redondantes sont supprimées via le regroupement K-means. Enfin, un algorithme de sélection de caractéristiques combinatoire, utilisant le Sequential Floating Backward Selection (SFBS), est appliqué pour obtenir des sous-ensembles optimaux. Cette approche vise à filtrer les caractéristiques à chaque étape pour obtenir le sous-ensemble le plus petit possible. Le système a été testé sur trois ensembles de données différents, incluant des images aériennes, des chiffres et des images d'animaux. dans leur dérniere experimentation ils ont réduit un dataset comportant 4096 caracteristiques jusqu'à 5% de sa taille originale.

II.2.3.4 Algorithme génetique hybride

Huang, Jinjie et al, ont introduit un Hybrid genetic algorithm (HGA) en deux étapes (filtre-enveloppe) pour augmenter la précision de la classification.[49] Alors que l'étage de filtrage en tant que boucle interne tente d'optimiser le critère MIFS amélioré avec l'information mutuelle conditionnelle sans paramètre, l'étage d'encapsulage en tant que boucle externe tente d'optimiser la statistique kappa. (La statistique kappa est couramment utilisée pour mesurer l'accord de deux classificateurs). L'approche Hybrid genetic feature selection (HGFS) proposée a été étudiée sur ses performances sur des ensembles de données, y compris l'ensemble de données sur le vin du référentiel UCI Machine Learning et des ensembles de données synthétiques. Tous les résultats sont comparés à d'autres méthodes de sélection de caractéristiques telles que celle de Battiti, MIFS et la méthode de sélection de caractéristiques RFE.

Après avoir examiné en détail tous les algorithmes présentés ci-dessus, il est maintenant temps de présenter une taxonomie inspirée de Saúl et al, [50] dans sa division démontrée par la figure suivante :

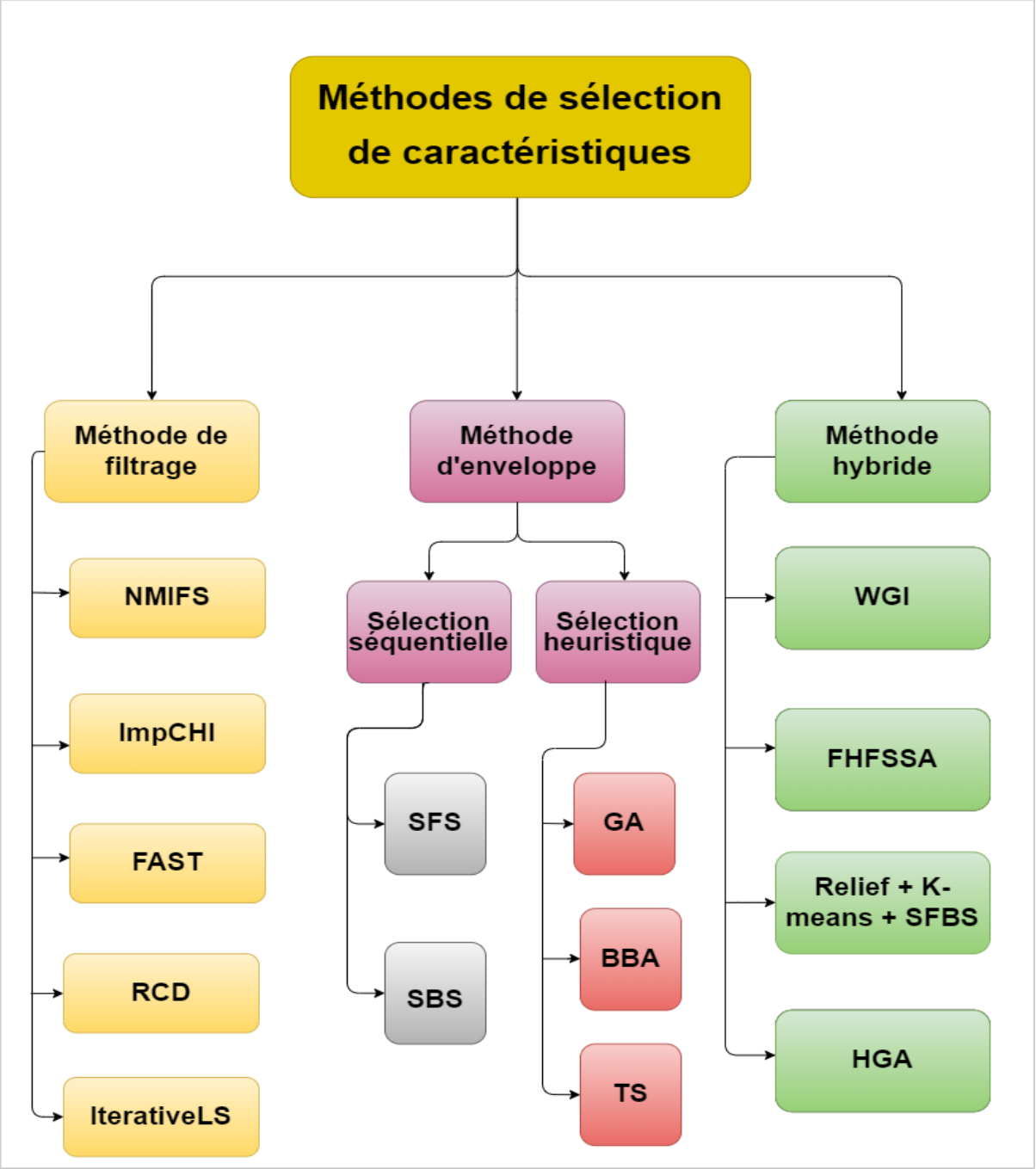

FIGURE II-3 – Taxonomie de la séléction de caractéristiques dans le machine learning

II.3 Avantages et inconvénients des diverses méthodes

Dans cette section , on vous présentent à l'aide d'un tableau les différents Avantages et inconvénients des différentes classes de méthodes

Méthodes de sélection de caractéristiques	**Avantages**	**Inconvénients**
Méthode de filtrage	Efficacité économique Rapidité d'exécution Génération de modèles robustes Capacité de généralisation supérieure Adaptabilité aux grands ensembles de données	Absence d'interaction avec le modèle de classification Ignorance des dépendances entre les caractéristiques Performances de calcul parfois inférieures
Méthode d'enveloppe	Interaction avec le classificateur Recherche exhaustive de l'espace des caractéristiques Prise en compte des dépendances entre les caractéristiques Meilleure généralisation	Coût de calcul élevé Durée de fonctionnement plus longue Risque de surajustement accru Impraticabilité pour un grand nombre de caractéristiques Manque de garantie d'optimalité
Méthode hybride	Efficacité de calcul accrue Rapidité d'exécution Interaction avec le modèle de classification Risque de surajustement réduit Meilleure généralisation avec de grands ensembles de données	Difficultés liées à l'identification d'un petit ensemble de caractéristiques Besoin de paramétrage

TABLE II.1 – Avantages et inconvénients de classes de méthodes

II.4 Tableau comparatif des différentes méthodes revues dans la littérature

Dans cette section, nous présentons un tableau comparatif des différentes méthodes de sélection de caractéristiques analysées dans la littérature.

Algorithme	**Référence de l'article**	**Principe de l'algorithme**	**Applications typiques**	**Indicateur**	**Algorithmes associés**
NMIFS	[31]	Utilise l'information mutuelle normalisée pour évaluer la pertinence des caractéristiques	Classification, reconnaissance de formes	Information mutuelle normalisée	Réseau de neurones
Imp-Chi	[33]	Améliore le test du Chi-carré pour mieux gérer les données déséquilibrées	Classification de textes	Chi-carré	Machine à vecteurs de support (SVM)
FAST	[51]	Utilise des techniques de clustering pour sélectionner rapidement des sous-ensembles de caractéristiques	Analyse de données de haute dimension	Cohérence intra-cluster	K-means, SVM
RCD	[7]	Sélectionne des caractéristiques basées sur leur redondance et complémentarité	Classification, analyse de données	Dispersion de redondance et de complémentarité	Régression linéaire, SVM
Iterative LS	[35]	Utilise le score de Laplacian itératif pour évaluer la pertinence des caractéristiques	Reconnaissance de formes, analyse de données	Score Laplacian	Réseaux de neurones, KNN
SFS	[37]	Ajoute séquentiellement des caractéristiques jusqu'à ce que le critère de performance soit atteint	Classification, régression	Précision, F-score	Réseau de neurones, KNN

TABLE II.2 – Tableau comparatif entre les différents méthodes revue dans l'état de l'art

Algorithme	**Référence de l'article**	**Principe de l'algorithme**	**Applications typiques**	**Indicateur**	**Algorithmes associés**
SBS	[38]	Supprime séquentiellement des caractéristiques jusqu'à ce que le critère de performance soit atteint	Optimisation de modèles, réduction de dimension	Précision, F-score	Régression logistique, KNN
GA	[39]	Utilise des techniques évolutionnaires pour sélectionner les caractéristiques	Optimisation, classification	Fonction de fitness	Réseau de neurones, SVM
BBA	[43]	Utilise des comportements de chauve-souris pour optimiser la sélection des caractéristiques	Classification, optimisation	Fonction de fitness	Régression logistique, SVM
WGI	[45]	Utilise l'indice de Gini pondéré pour sélectionner les caractéristiques	Classification, sélection de caractéristiques	Indice de Gini pondéré	Réseau de neurones, arbres de décision
FHFSSA	[47]	Combine des méthodes de recherche séquentielle flottante et hybride	Sélection de caractéristiques statistiques	Précision, F-score	Régression logistique, KNN
Relief+K-means+SFBS	[48]	Combine Relief, K-means et SFBS pour une sélection de caractéristiques robuste	Classification, reconnaissance de formes	Score Relief, cohérence intra-cluster	K-plus proches voisins (KNN) Régression logistique Arbre de décision
HGA	[52]	Combine les algorithmes génétiques avec d'autres méthodes pour une sélection de caractéristiques optimisée	Optimisation, sélection de caractéristiques	Fonction de fitness	Réseau de neurones, SVM

TABLE II.3 – Suite :Tableau comparatif entre les différents méthodes revue dans l'état de l'art

Avant de conclure ce chapitre nous présentons quelques défies dans le domaine de la sélection de caractéristiques :

- **Choix des caractéristiques :** Il est souvent complexe de sélectionner les caractéristiques à intégrer dans un modèle de Machine Learning. Il est important que les caractéristiques soient pertinentes, non linéairement dépendantes et non linéairement corrélées afin d'éviter les problèmes de multicollinéarité et d'overfitting.[53]

- **Omission de caractéristiques importantes :** L'absence de caractéristiques essentielles peut conduire à une diminution de la précision et à une réduction de la solidité du modèle. [54] Il est donc primordial de choisir les traits les plus adaptés et de les incorporer dans le modèle.
- **Inclusion de caractéristiques inutiles :** L'ajout de propriétés superflues peut accroître la complexité du modèle et causer des problèmes de suradaptation. Il est donc essentiel de choisir les caractéristiques de manière méticuleuse et de les évaluer afin de déterminer leur influence sur les performances du modèle [55].
- **Biais dans la sélection des caractéristiques :** Des biais peuvent influencer la sélection des caractéristiques, comme des préjugés ou des erreurs de collecte de données. Il est donc primordial de mettre en place des actions visant à réduire ces préjugés et à garantir la qualité des informations [56].
- **Complexité des algorithmes de sélection des caractéristiques :** Il est possible que les algorithmes de sélection des caractéristiques soient complexes et exigent une compréhension approfondie des concepts mathématiques qui les sous-tendent. [55] Il est donc crucial de sélectionner des algorithmes adaptés à la nature des données et aux objectifs du modèle.
- **Gestion des données de grande dimension :** [57] La gestion des données de grande taille peut poser des difficultés lorsqu'il s'agit de choisir les caractéristiques. Il est donc crucial de sélectionner des algorithmes adéquats afin de gérer ces données et d'évaluer l'influence de la dimensionnalité sur les performances du modèle.
- **Mise en œuvre des techniques de sélection des caractéristiques :** Il est possible que la mise en place des techniques de sélection des caractéristiques soit complexe et demande une bonne compréhension des concepts et des algorithmes. Il est donc crucial de sélectionner des méthodes adéquates et de les mettre en place de manière efficace [58].

II.5 Conclusion

Dans cette conclusion, nous avons exploré en profondeur une multitude d'articles de la littérature se penchant sur le domaine complexe de la sélection de caractéristiques. Nous avons minutieusement classé ces articles en trois catégories distinctes : Filtrage, Enveloppe et Hybride, afin d'offrir une structure claire à notre analyse. En examinant chacune de ces catégories, nous avons pris soin de discuter, d'analyser et de peser les avantages et les inconvénients des différentes méthodes présentées dans la littérature.

Grâce à notre approche, nous avons également réussi à créer un tableau comparatif complet, qui offre une vision d'ensemble synthétique des diverses méthodes. Ce tableau est un outil précieux pour mettre en évidence les avantages et les inconvénients de chaque méthode, offrant ainsi aux chercheurs et aux professionnels la possibilité de prendre des décisions éclairées concernant la sélection de caractéristiques.

Finalement, afin de faciliter la compréhension et la clarté, nous avons suggéré une taxonomie, une classification systématique qui permet de mieux situer chaque méthode dans le contexte complexe de la sélection de caractéristiques. La structure conceptuelle de cette taxonomie permet de comparer et de comprendre les différentes approches, tout en soulignant les liens et les différences entre elles.

En résumé, grâce à notre étude approfondie et à notre analyse minutieuse, nous avons pu établir un panorama complet du domaine de la sélection de caractéristiques, ce qui offre aux chercheurs et aux praticiens une fondation solide pour leurs futures recherches et leurs décisions.

Chapitre III

Proposition d'une approche hybride pour la sélection de caractéristique

III.1 Introduction

Dans ce chapitre, nous présentons une solution hybride innovante combinant TS et RFE pour optimiser les modèles d'apprentissage automatique. Notre objectif principal est de maximiser la précision des modèles tout en minimisant le nombre de caractéristiques, répondant ainsi aux défis de l'optimisation multi-objective dans le domaine de l'apprentissage automatique. La première phase de notre approche utilise la recherche taboue, une technique méta-heuristique puissante capable d'explorer efficacement de vastes espaces de solutions [59]. En tirant parti de la mémoire adaptative de la recherche taboue, nous identifions rapidement un ensemble initial de caractéristiques prometteuses. Cette phase initiale pose les bases d'une sélection de caractéristiques robuste en évitant les pièges des minima locaux et en offrant une exploration exhaustive de l'espace de recherche.

La seconde phase de notre approche affine cette sélection initiale à l'aide de la RFE. Cette méthode permet une évaluation fine des caractéristiques en éliminant progressivement les moins pertinentes, tout en tenant compte des interactions complexes entre elles. La RFE assure une sélection finale de caractéristiques qui maximisent la performance du modèle tout en maintenant sa simplicité.

En combinant ces deux techniques, notre solution hybride exploite les points forts de chacune pour offrir une méthode de sélection de caractéristiques à la fois efficace et robuste. Cette approche est particulièrement bien adaptée aux ensembles de données complexes et de grande dimension, où les relations entre les caractéristiques sont souvent

non linéaires et difficiles à modéliser.

Les sections suivantes détailleront les principes théoriques de TS et de la RFE, avant de décrire notre implémentation hybride en détail.

III.2 Recherche taboue

TS est un algorithme itératif qui explore l'espace des solutions en gardant une liste taboue des solutions récemment explorées pour éviter les cycles et échapper aux optima locaux. Les recherches locales examinent une solution potentielle à un problème et ses voisins immédiats dans l'espoir de trouver une solution améliorée. L'objectif principal de la recherche taboue est de proposer une approche globale basée sur cette technique pour traiter les problèmes d'ordonnancement complexes. Cette méthode utilise une heuristique de descente améliorée c'est a dire elle utilise une mémoire à court terme, appelée liste taboue, pour enregistrer les mouvements interdits et guider le processus de recherche, permettant ainsi d'éviter les minima locaux [60].

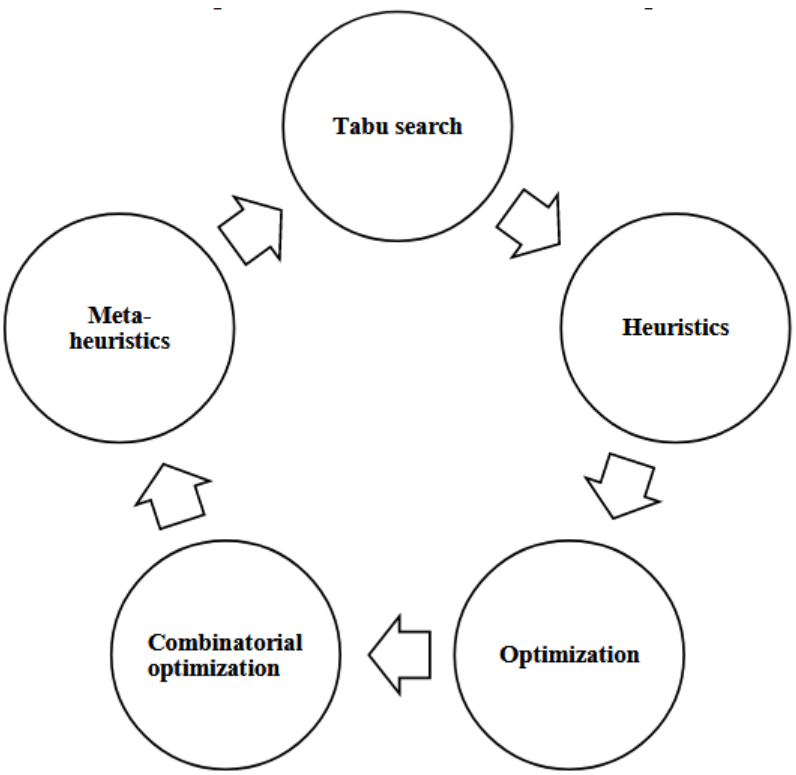

FIGURE III-1 – Algorithme de la recherche taboue [8]

III.2.1 Phase d'initialisation

III.2.1.1 Définition des Paramètres Initiaux

La recherche taboue nécessite plusieurs paramètres initiaux pour fonctionner efficacement. Ces paramètres incluent :

— **Taille de la liste taboue** : La taille de la mémoire adaptative qui stocke les mouvements interdits. Une taille appropriée permet d'éviter les cycles tout en permettant une exploration diversifiée de l'espace de recherche.
— **Nombre d'itérations** : Le nombre total d'itérations que l'algorithme doit effectuer. Ce paramètre peut être fixé ou dépendre d'un critère de convergence.
— **Critère d'arrêt** : Les conditions sous lesquelles l'algorithme doit s'arrêter, comme un nombre maximal d'itérations ou une amélioration négligeable de la solution.
— **Fonction d'évaluation** : La fonction utilisée pour évaluer la qualité des solutions. Cette fonction est essentielle pour guider l'algorithme vers les solutions optimales.

III.2.1.2 Génération de la solution initiale

La génération d'une solution initiale est une étape clé dans l'initialisation de la recherche taboue. Cette solution peut être générée de manière aléatoire ou en utilisant une heuristique spécifique qui fournit une solution de départ raisonnable. L'objectif est de fournir une base de départ pour l'algorithme à partir de laquelle il peut commencer à explorer l'espace de solutions [61].

III.2.1.3 Initialisation de la liste taboue

La liste taboue est une structure de mémoire qui conserve les mouvements récents pour éviter de revisiter les mêmes solutions. Lors de l'initialisation, cette liste est vide, mais sa taille est prédéfinie. Chaque fois qu'un mouvement est effectué, il est ajouté à la liste taboue, et les mouvements les plus anciens sont retirés lorsque la liste atteint sa capacité maximale.

III.2.1.4 Évaluation de la solution initiale

Une fois la solution initiale générée, elle doit être évaluée en utilisant la fonction d'évaluation définie. Cette évaluation fournit une mesure de la qualité de la solution, qui servira de référence pour les itérations suivantes de l'algorithme.

III.2.1.5 Préparation des mécanismes de recherche

Enfin, les mécanismes de recherche doivent être préparés. Cela inclut la définition des mouvements possibles (ou des voisins) à partir de la solution actuelle et la mise en place de stratégies pour explorer ces mouvements de manière efficace. Les mécanismes de diversification et d'intensification doivent également être définis pour équilibrer l'exploration globale et l'exploitation locale de l'espace de recherche [62].

III.2.2 Fonction objectif

Notre fonction objectif est définie pour maximiser la précision de la classification tout en minimisant le nombre de caractéristiques. Cette fonction peut être formulée comme suit :

$$f(x) = \alpha \cdot \text{accuracy}(x) - \beta \cdot \text{number\_of\_features}(x)$$

où α et β sont des coefficients de pondération ajustés pour équilibrer les deux objectifs.

III.2.3 Algorithme de recherche taboue

À partir d'une solution initiale, le principe général de la recherche tabou est illustré dans l'algorithme 1.

Algorithm 1 Schéma général d'un algorithme tabou

```
Engendrer une configuration initiale s
s ← s*
T ← ∅                                                    ▷ liste tabou
while Condition d'arrêt non satisfaite do
    m ← meilleur mouvement parmi ceux non tabou ou ceux vérifiant un critère d'aspiration
    Modifier s en effectuant le mouvement m
    Mettre T à jour
    if f(s) < f(s*) then
        s* ← s
    end if
end while
Retourner s
```

III.2.4 Motivation

TS est une méta-heuristique efficace basée sur une procédure de recherche locale enrichie de mécanismes pour éviter les cycles et surmonter les minima locaux. Son originalité réside dans l'acceptation du meilleur voisin, même s'il est moins performant que la solution

actuelle, pour éviter de se retrouver piégé dans un minimum local. La mémoire adaptative, qui conserve les dernières solutions inspectées, empêche les répétitions cycliques en déclarant ces solutions taboues. Cela dirige l'exploration vers des régions non visitées du domaine de solutions [61], garantissant une exploration exhaustive. La capacité de la recherche taboue à explorer de vastes espaces de solutions tout en maintenant une flexibilité et une adaptabilité élevées en fait un choix idéal pour notre approche d'optimisation de la sélection de caractéristiques. En combinant ces avantages, la recherche taboue permet de réduire efficacement le nombre de caractéristiques tout en améliorant la performance globale des modèles d'apprentissage automatique.

III.3 Recursive Feature Elimination

RFE est une stratégie de sélection de caractéristiques utilisée en apprentissage automatique pour optimiser la performance des modèles en réduisant la dimensionnalité des données. Elle adopte une approche itérative pour identifier les caractéristiques les plus pertinentes, éliminant progressivement celles qui sont les moins utiles. Ce processus utilise un estimateur, qui définit le modèle d'apprentissage automatique de base pour évaluer et classer les caractéristiques. À chaque itération, le modèle est entraîné, et les caractéristiques sont évaluées. Celles qui contribuent le moins à la qualité de la prédiction sont supprimées [27]. Ce cycle se répète jusqu'à ce qu'un nombre prédéfini de caractéristiques soit atteint, automatisant ainsi l'amélioration de la composition des caractéristiques et améliorant la capacité prédictive des modèles d'apprentissage automatique.
L'objectif principal de RFE est d'optimiser la performance du modèle tout en minimisant le nombre de caractéristiques utilisées, en trouvant le sous-ensemble de caractéristiques le plus pertinent pour la tâche de modélisation.

III.3.1 Estimateur

Dans le contexte de RFE, un estimateur est un algorithme ou un modèle d'apprentissage automatique utilisé pour entraîner les données et évaluer l'importance de chaque caractéristique. Cet estimateur peut être un modèle comme une régression linéaire, une machine à vecteurs de support SVM, un arbre de décision, ou tout autre algorithme capable de fournir une mesure d'importance des caractéristiques. À chaque étape itérative du processus RFE, l'estimateur détermine quelles caractéristiques doivent être éliminées pour améliorer la performance globale du modèle. L'estimateur peut être ajusté en fonction de la nature des données et de la tâche de modélisation, jouant ainsi un rôle crucial dans la sélection des caractéristiques les plus pertinentes [27].

III.3.2 Processus de fonctionnement

Le fonctionnement de la RFE peut être décrit à travers les étapes suivantes [9] :

1. **Sélection d'un Modèle de Base** : On commence par sélectionner un modèle d'apprentissage automatique. Ce modèle sera utilisé pour évaluer l'importance des caractéristiques.
2. **Entraînement du Modèle** : Le modèle est entraîné en utilisant toutes les caractéristiques disponibles dans l'ensemble de données.
3. **Évaluation de l'Importance des Caractéristiques** : Après l'entraînement, on évalue l'importance de chaque caractéristique en utilisant les coefficients (pour les modèles linéaires) ou les importances des caractéristiques (pour les modèles basés sur les arbres).
4. **Élimination de la Caractéristique la Moins Importante** : La caractéristique ayant la plus faible importance est éliminée de l'ensemble de caractéristiques.
5. **Répétition du Processus** : Les étapes 2 à 4 sont répétées de manière récursive jusqu'à ce que le nombre souhaité de caractéristiques soit atteint.

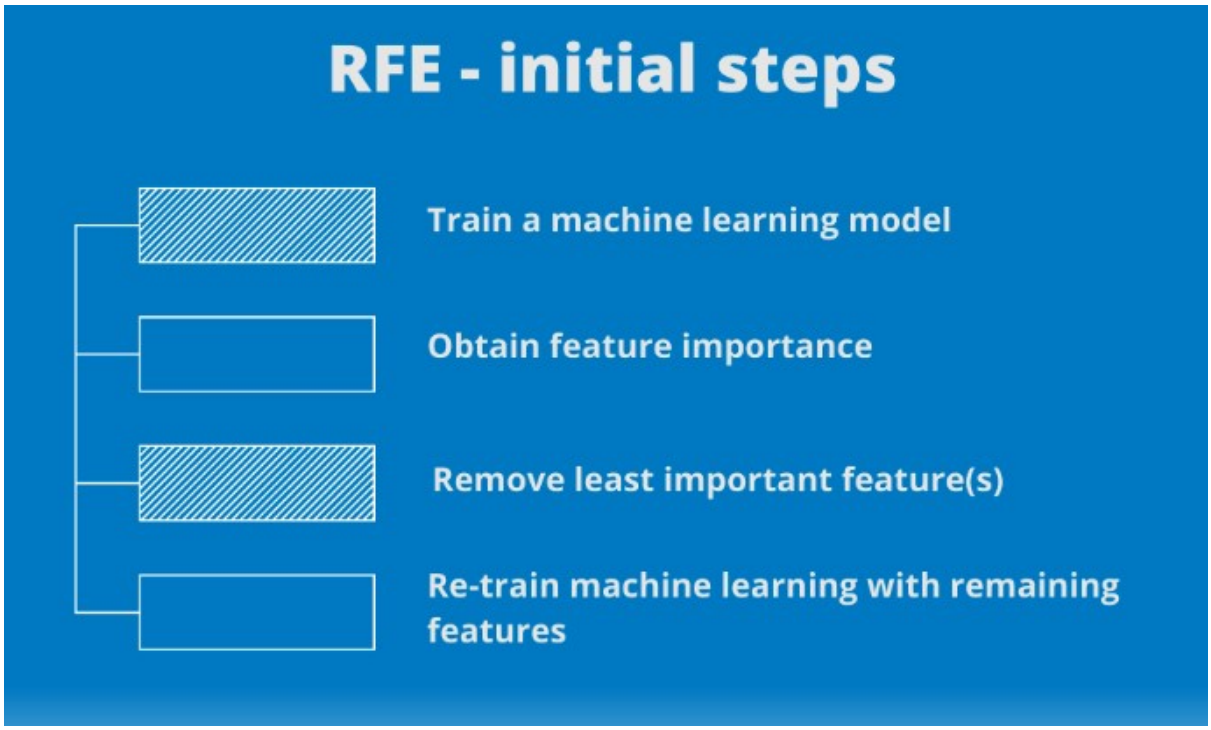

FIGURE III-2 – Processus de fonctionnement de RFE [9]

III.3.3 Motivation

Comparée à d'autres méthodes de sélection de caractéristiques, la RFE présente plusieurs avantages significatifs qui ont motivé notre choix. Tout d'abord, elle prend en compte les interactions entre les caractéristiques, ce qui signifie qu'elle évalue comment les caractéristiques fonctionnent ensemble plutôt que séparément. Ensuite, elle est adaptable aux ensembles de données complexes, ce qui signifie qu'elle peut gérer des données avec de nombreuses caractéristiques et des relations compliquées, ce qui est le cas de nos données, en utilisant des modèles avancés pour trouver les caractéristiques les plus importantes. Ces avantages sont essentiels pour améliorer la performance des modèles d'apprentissage automatique, en particulier lorsque les relations entre les caractéristiques peuvent être non linéaires et complexes. C'est précisément cette capacité à gérer les interactions complexes et les ensembles de données volumineux qui nous a conduits à choisir la RFE pour notre approche.

III.4 Processus d'hybridation

Notre approche hybride combine TS et RFE pour optimiser la sélection de caractéristiques, cette hybridation consiste à appliquer d'abord la recherche taboue, de manière ittérative, pour réduire le nombre de caractéristiques, jusqu'à ce que l'algorithme commence à perdre en performance, on fait appel alors à RFE pour compenser les lacunes de TS, et ainsi assurer un affinage supplémentaire.

III.4.1 Phase 1 : Recherche taboue itérative

Dans la première phase, nous appliquons directement l'algorithme de la TS sur l'ensemble de caractéristiques à séléctionner, et comme cet algorithme engendre un nombre important de caractéristiques, comme nous le verrons dans la partie expérimentale, nous avons décidé de modifer cet algorithme. Cette modification consiste à : après la première sélection avec TS, nous appliquons le même algorithme sur le sous-ensemble selectionné. Ainsi de suite, jusqu'à atteindre un nombre de caractéristiques assez faible, ou bien une performance médiocre.

III.4.2 Phase 2 : Sélection récursive de caractéristiques (RFE)

Après avoir obtenu les différents sous-ensemble avec TS, nous choisissons le plus adéquat d'entre eux en tennant compte de la performance et du nombre de caractéristiques selectionnées. Ensuite, le sous-ensemble choisie, est directement introduit à RFE pour

affiner davantage la sélection. Tout comme pour TS, nous séléctionnons différents tailles de sous-ensemble avec RFE pour choisir la meilleure configuration.

De cette manière, nous avons différentes tailles et types de sous ensemble de caractéristiques selectionnées, et tout dépend du domaine d'application, les ittérations, et le nombre de caractéristique change.

III.4.3 Combinaison des deux phases

L'intégration de la recherche taboue et de la RFE permet d'optimiser le processus de sélection de caractéristiques de manière plus efficace et robuste :

- **Réduction initiale** : La recherche taboue réduit l'ensemble des caractéristiques initiales à un sous-ensemble plus gérable et pertinent.
- **Affinage Précis** : La RFE affine ce sous-ensemble réduit en supprimant les caractéristiques moins importantes, en tenant compte des interactions complexes entre elles.

III.4.4 Avantages de l'Approche hybride

Cette approche hybride présente plusieurs avantages :

- **Efficacité** : La réduction initiale par la recherche taboue permet de simplifier le processus de sélection en diminuant le nombre de caractéristiques à évaluer.
- **Précision** : La RFE apporte une précision supplémentaire en affinant le sous-ensemble réduit, améliorant ainsi la performance globale du modèle.
- **Robustesse** : En combinant les deux méthodes, nous obtenons des modèles d'apprentissage automatique plus robustes et performants, capables de gérer des ensembles de données complexes et volumineux.

III.5 Évaluation des résultats

Pour évaluer l'efficacité de notre approche , nous utilisons plusieurs métriques de performance :

III.5.1 Exactitude (Accuracy)

Elle indique la proportion totale de prédictions correctes par rapport au total des cas testés. La formule pour calculer l'exactitude est :

$$\text{Exactitude} = \frac{\text{Nombre de prédictions correctes}}{\text{Nombre total de prédictions}}$$

Cette métrique donne une vue d'ensemble de l'efficacité du modèle.

III.5.2 Précision

Elle représente la proportion de prédictions positives qui sont correctement identifiées. C'est une mesure essentielle pour évaluer la qualité des prédictions positives du modèle, exprimée par :

$$\text{Precision} = \frac{\text{Vrais Positifs}}{\text{Vrais Positifs} + \text{Faux Positifs}}$$

III.5.3 Rappel

Il mesure la capacité du modèle à identifier tous les cas pertinents dans le dataset. Ce critère est crucial pour s'assurer que le modèle capture autant de cas positifs que possible, et est défini par :

$$\text{Rappel} = \frac{\text{Vrais Positifs}}{\text{Vrais Positifs} + \text{Faux Négatifs}}$$

III.5.4 F1-Score

Cette métrique combine la précision et le rappel en une seule mesure harmonique, très utile pour équilibrer ces deux aspects. Elle est calculée comme suit :

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

III.6 Conclusion

Dans ce chapitre, nous avons détaillé notre approche hybride pour la sélection de caractéristiques, combinant la Recherche Taboue itérative avec la Recursive Feature Elimination

(RFE). Cette combinaison permet de tirer parti des forces de chaque méthode pour optimiser la sélection des caractéristiques, augmentant ainsi la précision des modèles tout en réduisant leur complexité.

Après avoir conçu et présenté notre solution, qui repose sur une architecture robuste et bien pensée, il est maintenant temps de passer à la mise en œuvre pratique de notre projet. Le prochain chapitre sera donc consacré à l'implémentation de cette architecture, où nous mettrons en pratique les concepts et techniques présentés pour valider l'efficacité de notre approche.

Chapitre IV

Expérimentation et résultats obtenus

IV.1 Introduction

Après avoir finalisé la phase de formalisation de notre méthode, ce chapitre se concentre sur la phase d'implémentation, constituant la dernière partie de ce rapport. Cette phase a pour but de mettre en pratique notre approche. Nous commencerons par décrire les ressources matérielles utilisées et préciser l'environnement de développement du système. Ensuite, nous détaillerons les étapes suivies dans le processus d'expérimentation. Enfin, nous analyserons les résultats obtenus.

Dans ce chapitre, nous présentons les résultats obtenus en appliquant notre approche de sélection de caractéristiques sur deux domaines de recherches très actifs qui sont :**la reconnaissance de l'activité humaine** , et **la classification de la maladie de Parkinson** .

IV.2 Ressources Matérielles

Pour mener à bien notre implémentation, nous avons utilisé les ressources matérielles suivantes :

- **Ordinateur Portable** :
  - **Modèle** : HP 250 G7
  - **Processeur** : Intel(R) Core(TM) i3-8130U CPU @ 2.20GHz
  - **Mémoire RAM** : 8 Go DDR4
  - **Stockage** : SSD 256 Go Samsung

IV.3 Langages et outils et bibliothèques utilisées

Pour l'implémentation de notre projet, nous avons utilisé Python version 3.8.5 comme langage de programmation et Jupyter Notebook comme environnement de

développement intégré (IDE).

— **Python** : Python est un langage de programmation libre de grande qualité, développé par Guido van Rossum et publié pour la première fois en 1991. Python est un langage interprété qui peut être exécuté sans nécessiter de compilation. Le système de typage dynamique, la gestion automatique de la mémoire et une bibliothèque multifonctionnelle complète sont ses caractéristiques. Ce langage supporte diverses approches de programmation et peut être utilisé dans tous les systèmes d'exploitation. Python est un langage idéal pour les novices, mais il est également extrêmement stimulant pour les utilisateurs expérimentés [63]. Ce langage s'est propulsé en tête de la gestion d'infrastructure, d'analyse de données et dans le domaine du développement de logiciels. En effet, parmi ses qualités, Python permet notamment aux développeurs de se concentrer sur ce qu'ils font plutôt que sur la manière dont ils le font. Il a libéré les développeurs des contraintes de formes qui occupaient leur temps avec les langages plus anciens. Ainsi, développer du code avec Python est plus rapide qu'avec d'autres langages [64]

— **Google Colab** : Google Colab est un environnement de développement intégré (IDE) en ligne gratuit et open-source pour les langages de programmation Python, R, Julia, et Scala. Développé par Google, il offre une expérience collaborative et interactive pour l'apprentissage, la recherche et l'analyse de données, similaire à Jupyter Notebook[65]. Comme Jupyter, Google Colab permet de créer des documents avec des sorties interactives telles que HTML, images, vidéos, LaTeX et types MIME personnalisés. Il prend en charge plus de 40 langages de programmation et donne accès gratuitement aux ressources GPU et TPU de Google pour accélérer les tâches d'apprentissage automatique. Google Colab se distingue par sa facilité d'utilisation. Il suffit d'avoir un fichier notebook sur Google Drive pour commencer, sans besoin de configuration supplémentaire. L'intégration avec Google Drive facilite aussi le stockage, le partage et le contrôle de version des notebooks [66]. Cependant, après quelques années d'utilisation, certains utilisateurs ont relevé des limites comme le manque de certaines fonctionnalités importantes et des problèmes de support. Google Colab reste néanmoins un outil unique et très populaire, surtout pour les phases exploratoires de la recherche en data science et

machine learning.

IV.3.1 Les bibliothèques python utilisées

— **Scikit-learn** : (Sklearn) est la bibliothèque la plus utile et la plus robuste pour l'apprentissage automatique. Il fournit une sélection d'outils efficaces pour l'apprentissage automatique et la modélisation statistique, notamment la classification, la régression, le regroupement et la réduction de la dimensionnalité via une interface de cohérence en Python. Cette bibliothèque, qui est en grande partie écrite en Python, est construite sur NumPy, SciPy et Matplotlib [67].

— **Matplotlib** : est une bibliothèque complète pour la création de visualisations statiques, animées et interactives en Python [68].

— **Numpy** : La bibliothèque NumPy permet d'effectuer des calculs numériques avec NumPy propose des fonctions mathématiques complètes, des générateurs de nombres aléatoires, des routines d'algèbre linéaire, des transformées de Fourier. La syntaxe de haut niveau de NumPy le rend accessible et productif pour les programmeurs de tous horizons ou niveaux d'expérience [69].

— **Pandas** : est un outil d'analyse et de manipulation de données open source rapide, puissant, flexible et facile à utiliser, construit sur le langage de programmation Python [70].

IV.4 Démarche expérimentale

Dans cette rubrique, nous allons présenter les datasets utilisés et les différents résultats obtenus par notre modèle hybride. Nous avons utilisé deux ensembles de données principaux : le Human Activity Recognition (HAR) et le Parkinson's Disease Classification. Ces ensembles de données, vastes et riches en informations, ont permis d'évaluer l'efficacité et la robustesse de notre méthode.

IV.4.1 Présentation de la reconnaissance d'activité humaine

Human Activity Recognition (HAR) est une méthode permettant d'identifier l'activité d'une personne en utilisant des capteurs sensibles pour détecter les mouvements. Avec la croissance du nombre d'utilisateurs de smartphones et de leurs capacités (capteurs), les utilisateurs portent de plus en plus leurs téléphones avec eux. Ces faits augmentent l'importance et la popularité de la HAR [71].

Cette technologie est essentielle dans de nombreux domaines d'applications, tels que la

surveillance, la sécurité, la santé, l'assistance à la vie et bien d'autres, Il existe principalement deux types de la reconnaissance d'activité humaine :

— la reconnaissance d'activité humaine basée sur la vidéo.
— la reconnaissance d'activité humaine basée sur les capteurs [71].

Les méthodes et techniques utilisées pour la reconnaissance d'activité humaine comprennent principalement l'apprentissage automatique, l'analyse de signaux, la vision par ordinateur et les systèmes de reconnaissance, qui permettent d'analyser les données collectées par différents types de capteurs pour identifier les activités humaines [72].

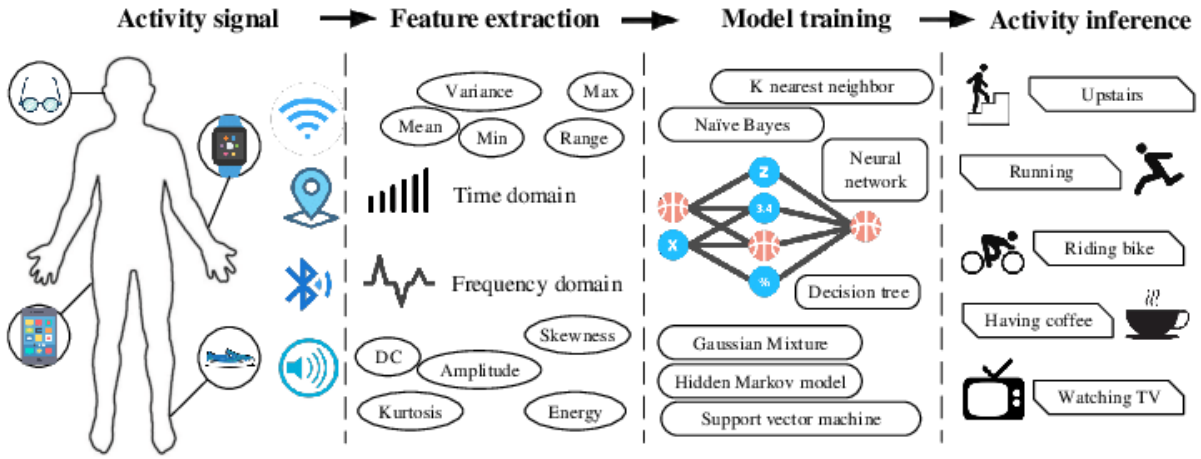

FIGURE IV-1 – Une illustration de la reconnaissance d'activité basée sur des capteurs utilisant des approches conventionnelles de reconnaissance de motifs. [10]

IV.4.2 Présentation du dataset tèlèchargè

Les expériences ont été réalisées avec un groupe de 30 volontaires âgés de 19 à 48 ans. Chaque personne a effectué six activités (MARCHER, MONTER LES ÉTAGES, DESCENDRE LES ÉTAGES, ASSISE, DÉBOUT, POSE) en portant un smartphone (Samsung Galaxy S II) à la taille. À l'aide de son accéléromètre et de son gyroscope intégrés, nous avons capturé l'accélération linéaire sur 3 axes et la vitesse angulaire sur 3 axes à une fréquence constante de 50 Hertz (Hz). Les expériences ont été enregistrées sur vidéo pour étiqueter les données manuellement. L'ensemble de données obtenu a été divisé au hasard en deux ensembles, où 70% des volontaires ont été sélectionnés pour générer les données de formation et 30% pour les données de test. Les signaux des capteurs (accéléromètre et gyroscope) ont été prétraités en appliquant des filtres de bruit, puis échantillonnés dans des fenêtres coulissantes à largeur fixe de 2,56 secondes et avec un chevauchement de 50% (128 lectures/fenêtre). Le signal d'accélération du capteur, qui comporte des composantes gravitationnelles et de mouvement du corps, a été séparé à l'aide d'un filtre

passe-bas de Butterworth en accélération du corps et gravité. La force gravitationnelle est supposée avoir uniquement des composantes basse fréquence, c'est pourquoi un filtre avec une fréquence de coupure de 0,3 Hz a été utilisé. À partir de chaque fenêtre, un vecteur de caractéristiques a été obtenu en calculant des variables du domaine temporel et fréquentiel.

Pour notre Expérimentation, nous avons opté pour la har basée sur les capteurs, car contrairement à la har basée sur les caméras, les capteurs assurent la confidentialité des utilisateurs, et n'entrave pas leurs vie privée. Dans ce qui suit, nous donnons une brève définition des capteurs utilisés : Accéléromètre, Gyroscope, Magnétometre

tBodyAcc-mean()-X	tBodyAcc-mean()-Y	tBodyAcc-mean()-Z	tBodyAcc-std()-X	tBodyAcc-std()-Y	tBodyAcc-std()-Z	tBodyAcc-mad()-X	tBodyAcc-mad()-Y	tBodyAcc-mad()-Z	tBodyAcc-max()-X
0.28858451	-0.020294171	-0.13290514	-0.9952786	-0.98311061	-0.91352645	-0.99511208	-0.98318457	-0.92352702	-0.93472378
0.27841883	-0.016410568	-0.12352019	-0.99824528	-0.97530022	-0.96032199	-0.99880719	-0.97491437	-0.95768622	-0.94306751
0.27965306	-0.019467156	-0.11346169	-0.99537956	-0.96718701	-0.97894396	-0.99651994	-0.96366837	-0.97746859	-0.93869155
0.27917394	-0.026200646	-0.12328257	-0.99609149	-0.9834027	-0.9906751	-0.99709947	-0.98274984	-0.9893025	-0.93869155
0.27662877	-0.016569655	-0.11536185	-0.99813862	-0.98081727	-0.99048163	-0.99832113	-0.97967187	-0.99044113	-0.94246912
0.27719877	-0.01009785	-0.10513725	-0.99733496	-0.99048681	-0.99542003	-0.9976274	-0.99021769	-0.9955489	-0.94246912
0.27945388	-0.019640776	-0.11002215	-0.99692104	-0.96718593	-0.98311783	-0.99700268	-0.96609671	-0.98311627	-0.94098663
0.27743247	-0.030488303	-0.12536043	-0.99655926	-0.96672843	-0.98158533	-0.99648525	-0.96631315	-0.98298176	-0.94098663
0.27729342	-0.021750698	-0.12075082	-0.99732847	-0.96124532	-0.98367156	-0.99759576	-0.95723623	-0.98437928	-0.94059758
0.28058569	-0.0099602983	-0.10606516	-0.99480344	-0.9727584	-0.98624387	-0.99540462	-0.97366322	-0.98564195	-0.94002751
0.27688027	-0.012721805	-0.10343832	-0.99481511	-0.97307692	-0.98535702	-0.99550927	-0.97394796	-0.98517247	-0.94002751
0.27622817	-0.021441302	-0.10820234	-0.99824595	-0.98721376	-0.99272659	-0.99825127	-0.98599654	-0.99318188	-0.94390578
0.278457	-0.020414761	-0.11273172	-0.99913488	-0.98468004	-0.99627424	-0.99907654	-0.98293702	-0.99641031	-0.94390578
0.27717497	-0.014712802	-0.10675647	-0.99918834	-0.99052638	-0.99336501	-0.99921135	-0.99068725	-0.99216753	-0.94332286
0.29794572	0.027093908	-0.061668123	-0.98864079	-0.8166986	-0.90190653	-0.98895795	-0.79428042	-0.8880146	-0.92597669
0.27920345	-0.023020143	-0.12208028	-0.99683904	-0.97484812	-0.98338551	-0.99709389	-0.97333193	-0.98406535	-0.9417158
0.27903836	-0.014800378	-0.11684896	-0.99694116	-0.98186562	-0.98257653	-0.99721998	-0.98161964	-0.98133604	-0.9417158
0.2801349	-0.013916951	-0.10637048	-0.99769492	-0.98751567	-0.99040744	-0.99801432	-0.98795448	-0.99219012	-0.94207598
0.27773106	-0.018210718	-0.10918803	-0.99749074	-0.99322197	-0.99612795	-0.99790305	-0.99271072	-0.99649182	-0.94487012
0.27556818	-0.016979698	-0.11142918	-0.99781139	-0.9905223	-0.99762104	-0.99820522	-0.98946983	-0.99719303	-0.94566163

FIGURE IV-2 – Le dataset UCI HAR

IV.4.3 Prétraitement des données

Le prétraitement des données est le processus de transformation des données brutes en un format compréhensible et utilisable. On dit souvent que les données sont le nouveau pétrole, mais nous n'utilisons pas le pétrole directement à la source. Il doit être traité et raffiné avant d'être utilisé à d'autres fins. Il en va de même pour les données : elles doivent être traitées et nettoyées avant d'être utilisées efficacement. C'est une étape cruciale dans l'exploration de données car nous ne pouvons pas travailler avec des données brutes. La qualité des données doit être vérifiée avant d'appliquer des algorithmes d'apprentissage automatique ou d'exploration de données. Les opérations de prétraitement que nous effectuons sur les deux datasets incluent :

1. **Nettoyage des données manquantes** : Cette étape consiste à identifier et à traiter les valeurs manquantes dans le jeu de données. Les approches courantes incluent la suppression des enregistrements incomplets, l'imputation des valeurs manquantes à l'aide de la moyenne, de la médiane ou d'autres méthodes statistiques, ou encore l'utilisation de techniques de modélisation pour estimer ces valeurs.
2. **Normalisation des données** : La normalisation consiste à redimensionner les valeurs des caractéristiques pour qu'elles se situent dans une même échelle, généralement entre 0 et 1, ou avec une moyenne nulle et un écart-type de 1 (standardisation). Cela est particulièrement important pour les algorithmes sensibles à l'échelle des données, tels que les SVM ou les réseaux neuronaux.
3. **Suppression des caractères spéciaux** : Cette étape vise à nettoyer les données textuelles en supprimant les caractères non pertinents, tels que les symboles, les ponctuations, ou autres caractères spéciaux, qui pourraient perturber l'analyse ou la modélisation. Cela est souvent nécessaire dans les tâches de traitement du langage naturel (NLP).
4. **Suppression des valeurs dupliquées** : Ici, l'objectif est d'identifier et de supprimer les enregistrements en double dans le jeu de données. Les valeurs dupliquées peuvent introduire du biais dans l'analyse et affecter la performance des modèles prédictifs.
5. **Séparation des données** : Cette étape consiste à diviser le jeu de données en ensembles d'entraînement, de validation et de test. Généralement, on attribue 70-80% des données à l'entraînement, 10-15% à la validation (pour ajuster les hyperparamètres), et 10-15% pour évaluer les performances finales du modèle.

IV.4.4 Implémentation de Taboo Search en hybridant avec RFE

Nous avons implémenté notre méthode TS, et comme mentionné dans le chapitre précédent, nous avons fait une hybridation entre le TS et RFE, premièrement nous avons fait une classification en utilisant les caractéristiques sélectionnés par la recherche tabou itérative et cela pour tester sa performance par rapport aux autres métriques utilisées , après, nous avons fait une calculé en hybridant TS avec RFE afin d'obtenir des résultats mieux que les premiers.

IV.4.5 Classification

Pour évaluer l'efficacité de la méthode proposée , les algorithmes de classification doivent être entraînés et testés à l'aide de différents ensembles de données. Les algorithmes les

plus utilisés sont Random forest (RF), Naïve Bayes (NB), Decision Tree (DT),KNN, SVM car ils sont reconnus comme ayant de bons résultats dans la tâche de classification. Pour cela, nous avons effectué une classification sur le dataset avec les deux algorithmes de classification SVM et NB avec les métriques d'évaluations les plus courantes (F1 Score, Exactitude, Précision, Rappel).

IV.4.5.1 Classification en appliquant la recherche tabou itérative

Les résultats présentés dans le tableau ci-dessous ont été obtenus à partir des caractéristiques obtenues en utilisant la méthode de recherche tabou itérative. Nous avons fait 5 itérations, avec la sélection de 287 sous-ensemble , puis 127, puis 73, 42, 28 et en fin 18. Nous avons obtenu de bons résultats en termes de F1 score, d'exactitude, de précision et de rappel, ce qui indique que notre méthode est efficace. Cependant, nous avons remarqué qu'à partir de 127 caractéristiques, les résultats commencent à chuter légèrement. C'est pour cela qu'on va continuer notre expérimentation à partir de la deuxième itération, car c'est le meilleur compromis entre la précision et le nombre de features.

Nombre de caractéristiques	**Exactitude (%)**	**Score F1 (%)**	**Précision (%)**	**Rappel (%)**
252	97.8641	97.9491	97.9466	97.9519
127	97.0388	97.1210	97.1397	97.1111
73	91.6990	91.9677	92.0233	91.9258
42	88.3495	88.5959	88.8120	88.7087
28	86.6019	86.7297	87.1186	86.8594
18	83.3495	83.3468	83.8716	83.4818

TABLE IV.1 – Résultats des Modèles de Classification Basés sur les caractéristiques Sélectionnées par la Méthode de Recherche Taboue Itérative.

IV.4.5.2 Classification en appliquant la recherche tabou itterative en l'hybridant avec RFE

Après avoir sélectionné les sous ensemble avec TS, nous appliquons RFE pour le raffinement, nous avons opté pour 5 différents sous ensemble avec cinq taille pour étudier l'impact qu'aura cette nouvelle sélection hybride par rapport aux résultats obtenus avec ts seule, c'est cinq tailles sont : 20, 40, 60, 80, 100.

Nous remarquons que les meilleurs résultats sont obtenus avec random forest RF, et nous jugeons que prendre 40 caractéristiques avec une précision de 97,73 est le meilleur

compromis entre nombre de features et performance du modèle

Modèle										
№ caractéristiques	RF		KNN		SVM		NB		DT	
	Score F1	Précision	Score F1	Précision	Score F1	Précision	Score F1	Précision	Score F1	Précision
100	97.96	98.00	94.97	95.04	96.86	96.90	75.36	79.54	92.97	92.98
80	97.80	97.84	94.74	94.85	96.71	96.82	77.54	80.57	93.12	93.12
60	97.65	97.66	94.91	95.17	96.43	96.54	83.71	84.08	93.16	93.14
40	97.73	97.76	93.49	93.61	95.90	96.12	83.25	83.76	93.56	93.55
20	96.86	86.90	93.71	93.80	94.53	95.79	84.72	86.28	92.80	92.70

TABLE IV.2 – Résultats des Modèles de Classification en appliquant la Recherche Tabou Itterative en l'hybridant avec RFE.

IV.5 Classification de la maladie de Parkinson

IV.5.1 Présentation du domaine

La maladie de Parkinson est une maladie du cerveau qui progresse lentement et affecte principalement la capacité à contrôler les mouvements. Elle se manifeste par des tremblements, une rigidité des muscles, une lenteur des mouvements, et des problèmes d'équilibre. Les personnes atteintes peuvent aussi avoir des problèmes de sommeil, de dépression et des troubles des fonctions automatiques du corps, comme la digestion. Ces nombreux symptômes rendent la maladie difficile à diagnostiquer et à traiter.
la classification précise de la maladie de Parkinson repose sur une intégration judicieuse de méthodes cliniques, biologiques et d'apprentissage automatique. Les avancées dans ces domaines sont essentielles pour améliorer le diagnostic, personnaliser les traitements et ouvrir de nouvelles voies de recherche thérapeutique [73].

IV.5.2 Présentation du Dataset

Les données utilisées dans cette étude proviennent de l'UCI Machine Learning Repository, plus précisément du dataset intitulé **Parkinson's Disease Classification** [74].

id	gender	PPE	DFA	RPDE	numPulses	numPeriodsPulses	meanPeriodPulses	stdDevPeriodPulses	locPctJitter	locAbsJitter	rapJitter	ppq5Jitter	ddpJitter	locShimmer	locDbShimmer
0	1	0.85247	0.71826	0.57227	240	239	0.00806353	8.68E-05	0.00218	1.76E-05	0.00067	0.00129	0.002	0.05883	0.517
0	1	0.76686	0.69481	0.53966	234	233	0.008258256	7.31E-05	0.00195	1.61E-05	0.00052	0.00112	0.00157	0.05516	0.502
0	1	0.85083	0.67604	0.58982	232	231	0.00833959	6.04E-05	0.00176	1.47E-05	0.00057	0.00111	0.00171	0.09902	0.897
1	0	0.41121	0.79672	0.59257	178	177	0.010857733	0.000182739	0.00419	4.55E-05	0.00149	0.00268	0.00446	0.05451	0.527
1	0	0.3279	0.79782	0.53028	236	235	0.008161574	0.002668863	0.00535	4.37E-05	0.00166	0.00227	0.00499	0.0561	0.497
1	0	0.5078	0.78744	0.65451	226	221	0.007631204	0.002696381	0.00783	5.97E-05	0.00232	0.00312	0.00697	0.07752	0.678
2	1	0.76095	0.62145	0.54543	322	321	0.005990989	0.000107266	0.00222	1.33E-05	0.00036	0.00094	0.00108	0.03203	0.28
2	1	0.83671	0.62079	0.51179	318	317	0.006073855	0.000135739	0.00282	1.71E-05	0.00034	0.00088	0.00103	0.063	0.539
2	1	0.80826	0.61766	0.50447	318	317	0.006057188	6.93E-05	0.00161	9.73E-06	0.00027	0.00068	0.00081	0.02783	0.244
3	0	0.85302	0.62247	0.54855	493	492	0.003910221	3.99E-05	0.00075	2.93E-06	9.00E-05	0.00025	0.00027	0.0567	0.512
3	0	0.80657	0.67256	0.61745	488	487	0.003956114	5.38E-05	0.00083	3.29E-06	0.0001	0.00026	0.00029	0.06639	0.641
3	0	0.82653	0.58326	0.44555	498	497	0.003872688	3.26E-05	0.00069	2.68E-06	7.00E-05	0.00021	0.00022	0.02531	0.218
4	0	0.8726	0.78996	0.78026	492	491	0.003924152	6.72E-05	0.0028	1.10E-05	0.00077	0.00184	0.0023	0.20811	1.814
4	0	0.81148	0.76831	0.70809	305	304	0.006316424	0.003245324	0.00341	2.16E-05	0.00093	0.00141	0.0028	0.13878	1.326
4	0	0.80978	0.77992	0.6918	291	290	0.006624185	0.002756584	0.00457	3.03E-05	0.00159	0.00292	0.00477	0.13069	1.222
5	1	0.81471	0.61483	0.33216	300	299	0.006433293	3.88E-05	0.00085	5.45E-06	0.00017	0.00042	0.00051	0.04046	0.354
5	1	0.83269	0.62018	0.37051	286	285	0.006754263	5.17E-05	0.00111	7.52E-06	0.00024	0.00059	0.00072	0.02995	0.266
5	1	0.82016	0.63124	0.37031	266	265	0.007256724	4.86E-05	0.00086	6.28E-06	0.0002	0.00045	0.00059	0.02734	0.241
6	1	0.78067	0.66085	0.44583	283	282	0.006824086	0.000138247	0.00177	1.21E-05	0.00025	0.00061	0.00075	0.0481	0.422
6	1	0.79774	0.71199	0.36714	289	288	0.006693036	6.49E-05	0.00122	8.19E-06	0.0002	0.00049	0.00061	0.08552	0.741
6	1	0.82169	0.62901	0.36176	292	291	0.006623612	2.80E-05	0.00084	5.58E-06	0.00018	0.00041	0.00055	0.02324	0.205

FIGURE IV-3 – Le dataset Parkinson's Disease Classification

IV.5.2.1 Nombre d'échantillons

Les données utilisées dans cette étude ont été recueillies auprès de 188 patients atteints de MP (107 hommes et 81 femmes) âgés de 33 à 87 ans [74].

IV.5.2.2 Nombre et types de caractéristiques

Le dataset contient 754 caractéristiques mesurant divers aspects de la voix, notamment :

- La fréquence fondamentale moyenne (MDVP :Fo(Hz))
- La fréquence fondamentale maximale (MDVP :Fhi(Hz))
- La fréquence fondamentale minimale (MDVP :Flo(Hz))
- Plusieurs mesures de variation de la fréquence fondamentale (MDVP :Jitter(%), MDVP :Jitter(Abs), MDVP :RAP, MDVP :PPQ, Jitter :DDP)
- Plusieurs mesures de variation de l'amplitude (MDVP :Shimmer, MDVP :Shimmer(dB), Shimmer :APQ3, Shimmer :APQ5, MDVP :APQ, Shimmer :DDA)
- Deux mesures du rapport signal-bruit (NHR, HNR)
- Deux mesures de la complexité dynamique non linéaire (RPDE, D2)
- Un exposant d'échelle fractale (DFA)
- Trois mesures non linéaires de variation de la fréquence fondamentale (spread1, spread2, PPE)

IV.5.3 Prétraitement des données

C'est une étape essentielle qui prépare les données de manière à ce qu'elles soient adaptées à l'entraînement du modèle.

Suppression des valeurs manquantes : cette étape consiste à retirer toutes les observations d'un ensemble de données qui contiennent au moins une valeur manquante, afin d'assurer que seules les données complètes et utilisables sont utilisées pour l'analyse ou la modélisation. **Transformation et normalisation des données** Normalisation des caractéristiques pour assurer une contribution équitable de chaque mesure dans les analyses ultérieures. Cette étape est effectuée en redimensionnant les données pour qu'elles aient une moyenne de zéro et une variance de un.

IV.5.4 Implémentation de Taboo Search en hybridant avec RFE

Dans cette rubrique, nous appliquons la même procédure que celle utilisée pour le premier dataset. Nous commençons par sélectionner les caractéristiques à l'aide de la recherche taboue itérative, puis nous affinons cette sélection avec RFE. Cette approche nous permet de comparer les performances des modèles et d'optimiser les résultats.

IV.5.4.1 Classification en appliquant la recherche tabou itterative

Les résultats présentés dans le tableau ci-dessous proviennent de l'application de notre modèle hybride combinant la recherche taboue itérative et la Recursive Feature Elimina-

tion (RFE). Cette approche a permis de diminuer le nombre de caractéristiques tout en conservant les plus pertinentes pour la classification. Nous avons évalué les performances du modèle Random Forest en utilisant des métriques telles que la précision, le score F1 et le rappel.

Nombre de caractéristiques	**Exactitude (%)**	**Score F1 (%)**	**Précision (%)**	**Rappel (%)**
381	87.5	80.97	87.70	77.63
192	84.86	76.41	83.94	73.24
103	84.86	76.97	82.91	74.12
44	86.18	77.91	87.98	74.12
25	82.23	70.85	80.82	67.98
15	79.6	63.48	78.38	61.84

TABLE IV.3 – Résultats des Modèles de Classification Basés sur les caractéristiques Sélectionnées par la Méthode de Recherche Taboue Itérative.

IV.5.4.2 Classification en appliquant la recherche tabou itterative en l'hybridant avec RFE

Le tableau 4.4 présente les résultats obtenus à partir des caractéristiques sélectionnées en utilisant la méthode de la recherche taboue itérative et RFE. Les performances des différents modèles RF, KNN, SVM, NB, DT ont été évaluées en termes de précision, score F1 .

Modèle										
№ caractéristiques	RF		KNN		SVM		NB		DT	
	Score F1	Précision	Score F1	Précision	Score F1	Précision	Score F1	Précision	Score F1	Précision
100	82.61	89.75	77.93	83.59	77.57	84.28	75.36	79.54	94.36	75.91
80	78.61	86.87	80.74	84.47	77.57	84.28	77.54	80.57	74.57	75.06
60	79.3	85.22	79.3	85.22	77.57	84.28	83.71	84.08	71.35	72.51
40	80.68	86.85	78.79	85.95	77.57	84.28	83.25	83.76	75.81	76.32
20	77.57	84.28	79.5	87.28	77.57	84.28	84.72	86.28	70.52	71.84

TABLE IV.4 – Résultats des Modèles de Classification en appliquant la recherche tabou itterative en l'hybridant avec RFE.

IV.6 Comparaison avec des méthode de l'état de l'art

Dans cette section, nous comparons l'approche proposée avec d'autres approche de la littérature dans les deux domaines d'application, à savoir : la reconnaissance d'activité humaine, et la maladie de parkinson.

IV.6.1 Reconnaissance d'activité humaine

Pour la HAR, nous avons selectionnées deux méthodes conccurentes de la littératures qui sont proposées par Bashar et al. [42], et Guha et al. [75]. Nous les comparons avec la notre en terme de précision et de nombre de caractéristiques selectionnées. Le Tableau IV.5 montre que notre approche surpasse les deux autres approches dans les deux métriques. En effet, pour l'approche [42], nous avons selectionnés 103 caractéristiques en moins, et nous avons amélioré la précision d'environ 2%. Et comparé à l'approche de [75] nous avons réduit les caractéristiques de 29 et amélioré la précision de plus de 2.5 %. Cela montre l'efficacité de notre approche, et l'atteinte des deux objectifs qui sont la maximisation de la précision et la minimalisation des caractéristiques selectionnées.

Méthode	**Caractéristiques selectionées**	**Précision (%)**
HAR-DNN[42]	143	95.79
CGA [75]	69	95.18
Proposée	**40**	**97.76**

TABLE IV.5 – Comparison de l'approche proposée avec d'autres méthode de la littérature sur le domaine de la HAR

IV.6.2 Maladie de parkinson

Pour la maladie de parkinson, nous avons également choisi deux méthodes de la littérature pour la comparaison en terme de nombre de caractéristiques selectionnées, et de précision. Ces méthodes en questions sont proposées par : Sakar et al. [76] et Gunduz et al. [77]. Notre approche surpasse ces deux autres dans les deux métriques, avec une amélioration de la précision de 1.28% et 1.78%, et une diminution de caractéristiques de 30 et 596 comparé à [76] et [77] respectivement. Cela confirme une autre fois, et dans un autre domaine que notre approche est meilleure.

Méthode	**Caractéristiques selectionées**	**Précision (%)**
mRMR[76]	50	86.00
DLBP [77]	614	84.50
Proposée	**20**	**87.28**

TABLE IV.6 – Comparison de l'approche proposée avec d'autres méthode de la littérature sur le domaine de la maladie de parkinson

IV.7 Conclusion

Dans ce dernier chapitre de notre projet, nous avons implémenté notre approche qui répond parfaitement aux objectifs fixés au début. Notre méthode de sélection de caractéristiques est basée sur l'hybridation de deux techniques : la Recherche Taboue itérative TS et la Recursive Feature Elimination RFE.
La Recherche Taboue itérative nous permet d'explorer de vastes espaces de solutions et d'éviter les minima locaux, tandis que la RFE affine cette sélection en évaluant l'importance des caractéristiques et en tenant compte des interactions complexes entre elles. Cette combinaison permet d'améliorer la performance de classification en éliminant les caractéristiques redondantes ou non informatives.
Les résultats obtenus ont été remarquables, mettant en évidence l'efficacité et la pertinence de notre méthode dans la résolution du problème posé. Cette approche montre un grand potentiel pour résoudre des défis complexes et ouvre de nouvelles perspectives.

Conclusion et perspectives

Ce mémoire a exploré l'importance de la sélection de caractéristiques dans le machine learning, en particulier pour améliorer la précision des modèles tout en réduisant leur complexité. Dans un contexte où les ensembles de données sont de plus en plus vastes et complexes, notre travail s'est focalisé sur le développement d'une méthode hybride combinant la Recherche Taboue itérative et la Recursive Feature Elimination (RFE).

La Recherche Taboue Itérative nous aide à parcourir de larges espaces de solutions et à éviter les pièges des minima locaux, tandis que la RFE affine cette sélection en estimant la valeur des attributs et en considérant les interactions complexes entre eux. Cette approche combinée permet d'optimiser les performances de classification en supprimant les attributs superflus ou non pertinents.

Les résultats obtenus montrent que notre approche hybride surpasse les méthodes traditionnelles de sélection de caractéristiques, telles que les méthodes de filtre ou de wrapper sans méta-heuristiques. Non seulement notre méthode améliore-t-elle la précision des modèles, mais elle permet également de réduire le nombre de caractéristiques nécessaires, simplifiant ainsi les modèles et réduisant les coûts de calcul.

En particulier, notre méthode a montré une robustesse accrue sur des ensembles de données complexes et à haute dimensionnalité, où les relations entre les caractéristiques peuvent être non linéaires et difficiles à détecter avec des méthodes plus simples. L'application de notre méthode aux vastes et riches ensembles de données du Human Activity Recognition (HAR) et de la classification de la maladie de Parkinson a démontré des améliorations significatives, validant ainsi l'efficacité de notre approche.

Cependant, il est essentiel de reconnaître certaines limitations de notre étude. Bien que nos ensembles de données soient vastes, des travaux futurs pourraient bénéficier de l'intégration de données provenant de différents domaines et types de modèles de langage. Il serait également pertinent d'explorer l'application de notre méthode à des contenus académiques tels que les mémoires, les articles et les thèses de doctorat pour garantir

l'authenticité des travaux soumis.

En conclusion, notre approche hybride de sélection de caractéristiques, alliant Recherche Taboue itérative et RFE, offre une solution robuste et efficace pour les modèles de machine learning par rapport a l'absence de l'étape de selection de caractéristique . Elle ouvre la voie à des améliorations futures et à des applications potentielles dans divers domaines, notamment là où la précision et l'efficacité sont cruciales.
Ce travail contribue ainsi de manière significative à l'optimisation des processus de sélection de caractéristiques et à l'amélioration des performances des modèles d'apprentissage automatique.

Bibliographie

[1] V. Silva, F. Rodrigues, Z. Vale, and J. Gouveia, "An electric energy consumer characterization framework based on data mining techniques," vol. 20, pp. 596 – 602.

[2] B. Larras, "CMOS analog implementation of clique-based neural networks."

[3] S. Benhammada, "Etude comparative de méthodes de sélection de caractéristiques en apprentissage automatique. proposition d'une variante." [Online]. Available : http://rgdoi.net/10.13140/RG.2.1.1677.0088

[4] M. Dash and H. Liu, "Feature selection for classification," vol. 1, no. 1, pp. 131–156, publisher : Elsevier. [Online]. Available : https://www.sciencedirect.com/science/article/pii/S1088467X97000085

[5] S. Salcedo-Sanz, G. Camps-Valls, F. Perez-Cruz, J. Sepulveda-Sanchis, and C. Bousono-Calzon, "Enhancing genetic feature selection through restricted search and walsh analysis," vol. 34, no. 4, pp. 398–406. [Online]. Available : http://ieeexplore.ieee.org/document/1347292/

[6] "What is Minimum Spanning Tree (MST)," Mar. 2023, (Consulté le 2024-05-21 à 13 :25 :52). [Online]. Available : https://www.geeksforgeeks.org/what-is-minimum-spanning-tree-mst/

[7] Z. Chen, C. Wu, Y. Zhang, Z. Huang, B. Ran, M. Zhong, and N. Lyu, "Feature selection with redundancy-complementariness dispersion," *Knowledge-Based Systems*, vol. 89, pp. 203–217, Nov. 2015. [Online]. Available : https://www.sciencedirect.com/science/article/pii/S0950705115002567

[8] N. Subash, M. Ramachandran, S. Vimala, and P. Vidhya, "An Investigation on Tabu Search Algorithms Optimization," *Electrical and Automation Engineering*, vol. 1, no. 1, pp. 13–20, Mar. 2022. [Online]. Available : http://restpublisher.com/wp-content/uploads/2022/03/An-Investigation-on-Tabu-Search-Algorithms-Optimization-2.pdf

[9] E. L. Korn and B. Freidlin, "A note on controlling the number of false positives," vol. 64, no. 1, pp. 227–231. [Online]. Available : https://academic.oup.com/biometrics/article/64/1/227-231/7331591

[10] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition : A survey," *Pattern Recognition Letters*, vol. 119, pp. 3–11, 2019, deep Learning for Pattern Recognition. [Online]. Available : https://www.sciencedirect.com/science/article/pii/S016786551830045X

[11] Feature engineering : définition et importance en machine learning. [Online]. Available : https://datascientest.com/feature-engineering

[12] G. H. John, R. Kohavi, and K. Pfleger, "Irrelevant features and the subset selection problem," in *Machine Learning Proceedings 1994*. Elsevier, pp. 121–129. [Online]. Available : https://linkinghub.elsevier.com/retrieve/pii/B9781558603356500234

[13] K. Menghour and L. Souici-Meslati, "Sélection de caractéristiques pour le filtrage de spams." pp. 349–360.

[14] Qu'est-ce que le data mining. [Online]. Available : https://www.oracle.com/fr/database/data-mining-definition.html

[15] T. M. Mitchell, "Artificial neural networks," vol. 45, no. 81, p. 127, publisher : Boston, MA : McGraw-Hill. [Online]. Available : http://www.cs.cmu.edu/~epxing/Class/10701-10s/Lecture/lecture7.pdf

[16] N. Le Roux, "Avancées théoriques sur la représentation et l'optimisation des réseaux de neurones," accepted : 2012-03-07T01 :20 :31Z. [Online]. Available : https://papyrus.bib.umontreal.ca/xmlui/handle/1866/6449

[17] P. Indyk and R. Motwani, "Approximate nearest neighbors : towards removing the curse of dimensionality," vol. 126, pp. 604–613.

[18] J. Figueroa Barraza, E. López Droguett, and M. Ramos Martins, "FS-SCF network : Neural network interpretability based on counterfactual generation and feature selection for fault diagnosis," vol. 237, p. 121670. [Online]. Available : https://linkinghub.elsevier.com/retrieve/pii/S0957417423021723

[19] C. Cortes and V. Vapnik, "Support-vector networks," vol. 20, no. 3, pp. 273–297. [Online]. Available : http://link.springer.com/10.1007/BF00994018

[20] A. H. Mohammad, T. Alwada'n, and O. Al-Momani, "Arabic text categorization using support vector machine, naïve bayes and neural network," vol. 5, no. 1, p. 16. [Online]. Available : https://link.springer.com/10.7603/s40601-016-0016-9

[21] D. Srivastava and L. Bhambhu, "Data classification using support vector machine," vol. 12, pp. 1–7.

[22] H. S. Ghennani and W. Medjdoub, "Utilisation de l'apprentissage automatique pour la sécurité d'un réseau de radio cognitive." accepted : 2018-10-17T11 :17 :42Z. [Online]. Available : http://dspace1.univ-tlemcen.dz//handle/112/13235

[23] R. M. Cormack, "A review of classification," vol. 134, no. 3, p. 321. [Online]. Available : https://www.jstor.org/stable/2344237?origin=crossref

[24] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1. Berkeley, CA : University of California Press, 1967, pp. 281–297.

[25] Z. Zhao and H. Liu, "Semi-supervised feature selection via spectral analysis," in *Proceedings of the 2007 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, pp. 641–646. [Online]. Available : https://epubs.siam.org/doi/10.1137/1.9781611972771.75

[26] X. Deng, Y. Li, J. Weng, and J. Zhang, "Feature selection for text classification : A review," vol. 78, no. 3, pp. 3797–3816. [Online]. Available : https://doi.org/10.1007/s11042-018-6083-5

[27] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection : A data perspective," vol. 50, no. 6, pp. 1–45. [Online]. Available : http://arxiv.org/abs/1601.07996

[28] S. Wang, J. Tang, and H. Liu, "Feature selection," in *Encyclopedia of Machine Learning and Data Mining*, C. Sammut and G. I. Webb, Eds. Springer US, pp. 1–9. [Online]. Available : https://doi.org/10.1007/978-1-4899-7502-7_101-1

[29] R. Kassel, "Corrélations de Pearson et de Spearman : Tout comprendre," Jun. 2023. [Online]. Available : https://datascientest.com/correlations-de-pearson-et-de-spearman

[30] B. Swingle, "Rényi entropy, mutual information, and fluctuation properties of Fermi liquids," *Physical Review B*, vol. 86, no. 4, p. 045109, Jul. 2012. [Online]. Available : https://link.aps.org/doi/10.1103/PhysRevB.86.045109

[31] P. A. Estevez, M. Tesmer, C. A. Perez, and J. M. Zurada, "Normalized Mutual Information Feature Selection," *IEEE Transactions on Neural Networks*, vol. 20, no. 2, pp. 189–201, Feb. 2009, conference Name : IEEE Transactions on Neural Networks. [Online]. Available : https://ieeexplore.ieee.org/document/4749258

[32] "Chi-Squared Test of Independence - AI ML Analytics," (Consulté le 2024-05-15 à 09 :12 :18). [Online]. Available : https://ai-ml-analytics.com/chi-squared-test-of-independence/

[33] S. Bahassine, A. Madani, M. Al-Sarem, and M. Kissi, "Feature selection using an improved Chi-square for Arabic text classification," *Journal of King Saud University - Computer and Information Sciences*, vol. 32, May 2018.

[34] Q. Song, J. Ni, and G. Wang, "A Fast Clustering-Based Feature Subset Selection Algorithm for High-Dimensional Data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 1, pp. 1–14, Jan. 2013, conference Name : IEEE Transactions on Knowledge and Data Engineering. [Online]. Available : https://ieeexplore.ieee.org/document/5989810

[35] L. Zhu, L. Miao, and D. Zhang, "Iterative Laplacian Score for Feature Selection," in *Pattern Recognition*, C.-L. Liu, C. Zhang, and L. Wang, Eds. Berlin, Heidelberg : Springer Berlin Heidelberg, 2012, vol. 321, pp. 80–87, series Title : Communications in Computer and Information Science. [Online]. Available : http://link.springer.com/10.1007/978-3-642-33506-8_11

[36] P. Pudil and P. Somol, "Identifying the most Informative Variables for Decision-Making Problems - a Survey of Recent Approaches and Accompanying Problems," *Acta Oeconomica Pragensia*, vol. 16, no. 4, pp. 37–55, Aug. 2008. [Online]. Available : http://aop.vse.cz/doi/10.18267/j.aop.131.html

[37] A. Marcano-Cedeño, J. Quintanilla-Domínguez, M. G. Cortina-Januchs, and D. Andina, "Feature selection using Sequential Forward Selection and classification applying Artificial Metaplasticity Neural Network," in *IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society*, Nov. 2010, pp. 2845–2850, iSSN : 1553-572X. [Online]. Available : https://ieeexplore.ieee.org/abstract/document/5675075

[38] S. O. Aregbesola, J. Won, S. Kim, and Y.-H. Byun, "Sequential backward feature selection for optimizing permanent strain model of unbound aggregates," *Case Studies in Construction Materials*, vol. 19, p. e02554, Dec. 2023. [Online]. Available : https://www.sciencedirect.com/science/article/pii/S2214509523007349

[39] D. E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*. Reading, Mass : Addison-Wesley Pub. Co, 1989.

[40] F. Brill, D. Brown, and W. Martin, "Fast generic selection of features for neural network classifiers," *IEEE Transactions on Neural Networks*, vol. 3, no. 2, pp. 324–328, Mar. 1992, conference Name : IEEE Transactions on Neural Networks. [Online]. Available : https://ieeexplore.ieee.org/document/125874

[41] A. AlSukk, R. N. Khushaba, and A. Al-Ani, "Enhancing the diversity of genetic algorithm for improved feature selection," in *2010 IEEE International Conference on Systems, Man and Cybernetics*. Istanbul, Turkey : IEEE, Oct. 2010, pp. 1325–1331. [Online]. Available : http://ieeexplore.ieee.org/document/5642445/

[42] S. K. Bashar, A. Al Fahim, and K. H. Chon, "Smartphone based human activity recognition with feature selection and dense neural network," in *2020 42nd Annual*

International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC). IEEE, 2020, pp. 5888–5891.

[43] R. Y. M. Nakamura, L. A. M. Pereira, K. A. Costa, D. Rodrigues, J. P. Papa, and X.-S. Yang, "BBA : A Binary Bat Algorithm for Feature Selection," in *2012 25th SIBGRAPI Conference on Graphics, Patterns and Images*, Aug. 2012, pp. 291–297, iSSN : 2377-5416. [Online]. Available : https://ieeexplore.ieee.org/abstract/document/6382769

[44] M. Gmira, M. Gendreau, A. Lodi, and J.-Y. Potvin, "Tabu search for the time-dependent vehicle routing problem with time windows on a road network," *European Journal of Operational Research*, vol. 288, no. 1, pp. 129–140, 2021. [Online]. Available : https://www.sciencedirect.com/science/article/pii/S0377221720304872

[45] H.-H. Hsu, C.-W. Hsieh, and M.-D. Lu, "Hybrid feature selection by combining filters and wrappers," *Expert Systems with Applications*, vol. 38, no. 7, pp. 8144–8150, Jul. 2011. [Online]. Available : https://www.sciencedirect.com/science/article/pii/S0957417410015198

[46] H. Liu, M. Zhou, and Q. Liu, "An embedded feature selection method for imbalanced data classification," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 3, pp. 703–715, May 2019. [Online]. Available : https://ieeexplore.ieee.org/document/8677302/

[47] P. Somol, J. Novovičová, and P. Pudil, "Flexible-Hybrid Sequential Floating Search in Statistical Feature Selection," in *Structural, Syntactic, and Statistical Pattern Recognition*, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, D.-Y. Yeung, J. T. Kwok, A. Fred, F. Roli, and D. De Ridder, Eds. Berlin, Heidelberg : Springer Berlin Heidelberg, 2006, vol. 4109, pp. 632–639, series Title : Lecture Notes in Computer Science. [Online]. Available : http://link.springer.com/10.1007/11815921_69

[48] J. Bins and B. Draper, "Feature selection from huge feature sets," in *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, vol. 2, Jul. 2001, pp. 159–165 vol.2. [Online]. Available : https://ieeexplore.ieee.org/document/937619

[49] J. Huang and P. Rong, "A Hybrid Genetic Algorithm for Feature Selection Based on Mutual Information," in *Information Theory and Statistical Learning*, F. Emmert-Streib and M. Dehmer, Eds. Boston, MA : Springer US, 2009, pp. 125–152. [Online]. Available : http://link.springer.com/10.1007/978-0-387-84816-7_6

[50] S. Solorio-Fernández, J. A. Carrasco-Ochoa, and J. F. Martínez-Trinidad, "A new hybrid filter–wrapper feature selection method for clustering based on

ranking," *Neurocomputing*, vol. 214, pp. 866–880, Nov. 2016. [Online]. Available : https://linkinghub.elsevier.com/retrieve/pii/S0925231216307718

[51] L. Song, A. Smola, A. Gretton, K. M. Borgwardt, and J. Bedo, "Supervised feature selection via dependence estimation," in *Proceedings of the 24th international conference on Machine learning*. ACM, pp. 823–830. [Online]. Available : https://dl.acm.org/doi/10.1145/1273496.1273600

[52] S. H. Huang, "Supervised feature selection : A tutorial." vol. 4, no. 2, pp. 22–37. [Online]. Available : https://www.researchgate.net/profile/Samuel-Huang-3/publication/275228384_Supervised_feature_selection_A_tutorial/links/5a1d720f0f7e9b2a531726a3/Supervised-feature-selection-A-tutorial.pdf

[53] h. Mezili, "Vers une amélioration de la détection d'intrusion par les méthodes de sélection des fonctionnalités à l'aide des arbres de décision," Réseaux et Télécommunications, UNIVERSITE IBN KHALDOUN - TIARET, Tiaret, 2021.

[54] "10 sujets de mémoire sur le machine learning," (Consulté le 2024-05-16 à 11 :02 :31). [Online]. Available : https://www.pimido.com/blog/vie-etudiant/sujets-memoire-machine-learning-16-02-2023.html

[55] G. Fenza, M. Gallo, V. Loia, F. Orciuoli, and E. Herrera-Viedma, "Data set quality in Machine Learning : Consistency measure based on Group Decision Making," *Applied Soft Computing*, vol. 106, p. 107366, Jul. 2021. [Online]. Available : https://www.sciencedirect.com/science/article/pii/S1568494621002891

[56] P. Kim, "Machine Learning," in *MATLAB Deep Learning : With Machine Learning, Neural Networks and Artificial Intelligence*, P. Kim, Ed. Berkeley, CA : Apress, 2017, pp. 1–18. [Online]. Available : https://doi.org/10.1007/978-1-4842-2845-6_1

[57] M. Labiadh, C. Obrecht, C. Ferreira da Silva, and P. Ghodous, "On the suitability of Data Selection for Cross-building Knowledge Transfer," in *2019 International Conference on High Performance Computing & Simulation (HPCS)*, Jul. 2019, pp. 818–824. [Online]. Available : https://ieeexplore.ieee.org/document/9188132

[58] A. I. Khan and S. Al-Habsi, "Machine Learning in Computer Vision," *Procedia Computer Science*, vol. 167, pp. 1444–1451, 2020. [Online]. Available : https://linkinghub.elsevier.com/retrieve/pii/S1877050920308218

[59] "Recherche tabou," page Version ID : 215752081. [Online]. Available : https://fr.wikipedia.org/w/index.php?title=Recherche_tabou&oldid=215752081

[60] S. Ru, "Vehicle logistics intermodal route optimization based on tabu search algorithm," vol. 14, no. 1, p. 11859. [Online]. Available : https://www.nature.com/articles/s41598-024-60361-7

[61] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization : Overview and conceptual comparison," vol. 35, no. 3, pp. 268–308. [Online]. Available : https://dl.acm.org/doi/10.1145/937503.937505

[62] A. C. Fierro, R. Thuret, K. Engelen, G. Bernot, K. Marchal, and N. Pollet, "Evaluation of time profile reconstruction from complex two-color microarray designs," vol. 9, no. 1, p. 1. [Online]. Available : https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-9-1

[63] "Answer to "How do I reference the Python programming language in a thesis or a paper ?"," Nov. 2012. [Online]. Available : https://academia.stackexchange.com/a/5484

[64] "Python : définition et utilisation de ce langage informatique." [Online]. Available : https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1445304-python-definition-et-utilisation-de-ce-langage-informatique/#

[65] T. N. Prabhu, "Mastering the features of Google Colaboratory ! ! !" Dec. 2020. [Online]. Available : https://towardsdatascience.com/mastering-the-features-of-google-colaboratory-92850e75701

[66] H. Michel, "Google Colab : Le guide Ultime," May 2019, section : Data Science. [Online]. Available : https://ledatascientist.com/google-colab-le-guide-ultime/

[67] "Scikit Learn Tutorial." [Online]. Available : https://www.tutorialspoint.com/scikit_learn/index.htm

[68] "Matplotlib — Visualization with Python." [Online]. Available : https://matplotlib.org/

[69] "NumPy -." [Online]. Available : https://numpy.org/

[70] "pandas - Python Data Analysis Library." [Online]. Available : https://pandas.pydata.org/

[71] Z. Hussain, M. Sheng, and W. E. Zhang, "Different approaches for human activity recognition : A survey," *arXiv preprint arXiv :1906.05074*, 2019.

[72] J. Aïzan, "Modélisation et reconnaissance d'activités quotidiennes au sein d'une maison intelligente : application à la surveillance des personnes âgées," Theses, Université du Littoral Côte d'Opale ; Université d'Abomey-Calavi (Bénin), Oct. 2020. [Online]. Available : https://theses.hal.science/tel-03052115

[73] Parkinson's disease : Causes, symptoms, and treatments. [Online]. Available : https://www.nia.nih.gov/health/parkinsons-disease/parkinsons-disease-causes-symptoms-and-treatments

[74] G. S. C. Sakar, "Parkinson's disease classification." [Online]. Available : https://archive.ics.uci.edu/dataset/470

[75] R. Guha, A. H. Khan, P. K. Singh, R. Sarkar, and D. Bhattacharjee, "Cga : A new feature selection model for visual human action recognition," *Neural Computing and Applications*, vol. 33, pp. 5267–5286, 2021.

[76] C. O. Sakar, G. Serbes, A. Gunduz, H. C. Tunc, H. Nizam, B. E. Sakar, M. Tutuncu, T. Aydin, M. E. Isenkul, and H. Apaydin, "A comparative analysis of speech signal processing algorithms for parkinson's disease classification and the use of the tunable q-factor wavelet transform," *Applied Soft Computing*, vol. 74, pp. 255–263, 2019.

[77] H. Gunduz, "Deep learning-based parkinson's disease classification using vocal feature sets," *Ieee access*, vol. 7, pp. 115 540–115 551, 2019.

Résumé

Ce projet aborde le défi de la sélection de caractéristiques dans le machine learning, avec pour objectif d'améliorer la précision des modèles tout en réduisant leur complexité. Nous avons développé une approche hybride combinant la Recherche Taboue itérative et la Recursive Feature Elimination (RFE) pour exploiter leurs avantages respectifs.
Notre objectif principal est de maximiser la précision des modèles tout en minimisant le nombre de caractéristiques, répondant ainsi aux défis de l'optimisation multi-objective dans le domaine de l'apprentissage automatique.
Notre approche a été testée sur les datasets de Reconnaissance d'Activités Humaines (HAR) et de Classification de la Maladie de Parkinson, démontrant des améliorations significatives et validant son efficacité. Les résultats montrent que notre méthode hybride de sélection de caractéristiques conduit à une précision de classification supérieure, offrant une solution plus efficace et robuste.

Mots-clés : Sélection de Caractéristiques, Apprentissage Automatique, Recherche Taboue, Recursive Feature Elimination, Reconnaissance d'Activités Humaines, Classification de la Maladie de Parkinson, Optimisation de Modèle.

Abstract

This project addresses the challenge of feature selection in machine learning, with the objective of improving model accuracy while reducing complexity. We have developed a hybrid approach combining Iterative Tabu Search and Recursive Feature Elimination (RFE) to leverage their respective advantages.
Our primary objective is to maximize model accuracy while minimizing the number of features, thus addressing the challenges of multi-objective optimization in the field of machine learning.
Our approach was tested on the Human Activity Recognition (HAR) and Parkinson's Disease Classification datasets, demonstrating significant improvements and validating its effectiveness. The results show that our hybrid feature selection method leads to superior classification accuracy, offering a more efficient and robust solution.

Keywords : Feature Selection, Machine Learning, Tabu Search, Recursive Feature Elimination, Human Activity Recognition, Parkinson's Disease Classification, Model Optimization.