

Mémoire présenté à La Faculté des Sciences-Meknès

Pour l'obtention du Diplôme de Master

Intelligence Artificielle et Analyse des Données (IAAD)

Titre :

Bio-informatique et IA :

**Classification des Éléments Transposables CII
dans les Séquences ADN : transposons à TIR**

Réalisée par :

Assia BENGHDAIF

Encadré par :

M. Ali BEKRI

Soutenu le : 10/09/2024 devant le jury composé de :

Pr. Hamid BOURRAY

Professeur à FS de Meknès

Pr. Ali OUBELKACEM

Professeur à FS de Meknès

Pr. Ed-drissiya EL-ALLALY

Professeur à FS de Meknès

Pr. Ali BEKRI

Professeur à FS de Meknès

Remerciement

« *La reconnaissance est la mémoire du cœur.* »

[Hans Christian Andersen]

Je remercie également les membres du Jury qui ont accepté d'évaluer la pertinence de mon travail.

Quand les personnes ne ménagent ni temps ni connaissances pour nous informer et nous conseiller, les remercier ne relève pas seulement des règles de politesse, mais aussi de la reconnaissance et du devoir envers eux.

Mes remerciements vont particulièrement à mon encadrant **M. Ali BEKRI** pour ses orientations, ses conseils et son aide précieuse tout au long de ma période de ce stage.

Je tiens à exprimer ma profonde gratitude envers nos **collègues biologistes** pour leur collaboration précieuse dans le cadre de ce projet. Grâce à leur expertise et à leur engagement, nous avons pu mener à bien la livraison du dataset des éléments transposables, un composant essentiel de notre recherche.

Je souhaite aussi remercier l'équipe pédagogique et administrative de la faculté des sciences de Meknès pour leurs efforts dans le but de nous offrir une excellente formation.

Je remercie également les membres du Jury **M. BOURRAY, M. OUBELKACEM et Mme EL-ALLALY** qui ont accepté d'évaluer la pertinence de mon travail.

Finalement, je souhaite remercier toute personne ayant contribué de près ou de loin à la réalisation de ce travail

Résumé

Les éléments transposables, également appelés éléments mobiles ou transposons, sont des séquences d'ADN qui ont la capacité de se déplacer ou de se copier d'une région à une autre du génome. Ils constituent une part significative de l'ADN non codant présent dans les génomes des organismes, allant des bactéries aux humains.

Ce document est un rapport de stage de fin d'étude inscrit dans le programme de quatrième semestre du master intelligence artificielle et analyse de données MIAAD qui s'est déroulé à la faculté des sciences de Moulay Ismail de Meknès. Le but derrière de ce projet est d'améliorer le processus de la détection des éléments transposables de type Transposons a TIR (class 2 : La Sous-classe 1) TCI-Mariner, PiggyBac, P élément, hAT, CACTA, Mutator et Merlin en réduisant la complexité et en développant un programme bio-informatique basé sur deux algorithmes machine learning/ deep learning : LSTM pour la classification binaire des élément transposable et non transposable et modèle hybride CNN-GRU pour la classification multiclass plus précise de superfamilles de ces éléments au sein des génomes et finalement la détection des TIRs de ces superfamilles.

Et à partir de ce rapport en va voir les tous ses phases pas par pas.

Mots clés : élément transposable, élément non transposable, TCI-Mariner, PiggyBac, P élément, hAT, CACTA, Mutator, Merlin ADN, génome, machine learning, deep learning, TIR
...

Abstract

Transposable elements, also known as mobile elements or transposons, are DNA sequences that have the ability to move or copy themselves from one region of the genome to another. They constitute a significant portion of the non-coding DNA present in the genomes of organisms, ranging from bacteria to humans.

This document is a final internship report as part of the fourth semester program of the Master's in Artificial Intelligence and Data Analysis (MIAAD) at the Faculty of Sciences of Moulay Ismail University in Meknès. The goal of this project is to improve the process of detecting transposable elements of the TIR-type transposons (Class 2: Subclass 1) such as Tc1-Mariner, PiggyBac, P element, hAT, CACTA, Mutator, and Merlin by reducing complexity and developing a bioinformatics program based on two machine learning/deep learning algorithms: LSTM for binary classification of transposable and non-transposable elements and a hybrid CNN-GRU model for more precise multi-class classification of these elements' superfamilies within genomes, and finally, the detection of TIRs in these superfamilies.

This report will detail all these phases step by step.

Keywords: transposable element, non-transposable element, Tc1-Mariner, PiggyBac, P element, hAT, CACTA, Mutator, Merlin, DNA, genome, machine learning, deep learning, TIR...

Table des matières

Remerciement	
Résumé	
Abstract	
Table des matières	
Liste des figues	
Liste des Tableaux	
Nomenclature	
Introduction générale.....	13
Chapitre I : Etat de l'art.....	14
1. IA en Bio-informatique	14
2. Historique des ETs [1].....	14
3. Définition	15
4. Classification et variations structurales.....	15
5. Les superfamilles.....	16
a) Les rétrotransposons à LTR	18
b) Les rétrotransposons non-LTR	18
c) Les transposons à TIR : La Sous-classe 1 [3][4].....	19
6. Travaux connexes.....	20
7. Cadre général du projet	20
a. Cahier de charge.....	20
b. Problématique.....	21
8. Conclusion.....	21
Chapitre II : Les superfamilles de la classe II	22
1. Introduction	22
2. Définitions et structure	22

a)	La superfamille Tc1-Mariner [9] (ANNEXE A).....	22
b)	La superfamille hAT [10] (ANNEXE A).....	24
c)	La superfamille P élément [11] (ANNEXE A)	25
d)	La superfamille Mutator/ MuDR [12] (ANNEXE A).....	26
e)	La superfamille Merlin [14] (ANNEXE A)	27
f)	La superfamille CACTA [15] (ANNEXE A)	27
g)	La superfamille PiggyBac (ANNEXE A)	27
3.	Conclusion.....	28
	Chapitre III : Etude fonctionnelle.....	29
a.	Introduction	29
b.	IA [16]	29
c.	ML.....	30
a.	Types d'Apprentissage.....	30
a.1.	Apprentissage Supervisé	31
a.1.	Apprentissage Non Supervisé	31
a.1.	Apprentissage semi-supervisé	31
a.1.	Apprentissage par Renforcement	32
d.	DL.....	32
e.	NLP [17].....	32
5.1.	NLU	33
5.2.	NLG	33
5.3.	Applications	33
5.3.1.	Reconnaissance vocale	33
5.3.2.	Reconnaissance des entités nommées (NER)	33
5.3.3.	Chat GPT	33
f.	Généralité sur les modèles ML/DL utilises.....	34
6.1.	Modèles ML.....	34

a.	Random Forest	34
b.	Extra Trees [22][23]	34
c.	Naive Bayes.....	35
d.	SVM [24].....	36
6.2.	Modèles DL	36
a.	LSTM	36
a.1.	Architecture [25].....	36
b.	CNN	37
b.1.	CNN et GRU [26]	38
b.2.	CNN et BiLSTM.....	39
c.	DNAAlbert [27][28]	40
6.3.	K-means	41
d.	Conclusion.....	42
	Chapitre IV : Etude technique.....	43
1.	Introduction	43
2.	Architecture du projet.....	43
2.1.	Partie 1 : classification binaire	43
2.2.	Partie 2 : classification multi-classes	44
2.3.	Partie 3 : Localisation des TIRs	45
3.	Collecte de données.....	46
4.	Prétraitement des données	50
5.	Tokenisation des séquences	51
6.	Vectorisation des séquences.....	52
6.1.	TF-IDF	52
6.2.	CountVectorizer	52
6.3.	One Hot Encoding.....	52
7.	Ensemble des données après le prétraitement	53

8.	Knowledge discovery : choix des modèles ML/DL	53
a.	K-means	53
b.	Classification binaire.....	55
b.1.	ML	55
b.1.1.	Random Forest	55
b.1.2.	Extra Trees	60
b.1.3.	Naïve Bayes	64
b.1.4.	SVM.....	68
b.2.	DL : LSTM.....	72
b.3.	Discutions des résultats	76
b.4.	Résultat final	76
c.	Classification multi-classe.....	79
c.1.	ML	79
c.1.1.	Random Forest	80
c.1.2.	Extra Trees	86
c.1.3.	Naïve Bayes	91
c.2.	DL.....	93
c.2.1.	Bert.....	93
c.2.1.	CNN	94
c.2.1.	CNN et BiLSTM.....	96
c.2.2.	CNN et GRU.....	98
c.3.	Discutions des résultats	98
c.4.	Résultat final	100
9.	Conclusion.....	100
	Chapitre IV : Réalisation du projet	101
1.	Introduction	101
2.	Choix techniques	101

2.1.	Environnement de développement.....	101
a.	Kaggle	101
b.	Google Colab.....	101
c.	IntelliJ IDEA	101
d.	Visual Studio Code.....	101
2.2.	Outils utilisés	102
i.	Langages utilisés.....	102
a.	Python.....	102
b.	Java pour l'Android.....	102
ii.	Framework et bibliothèques utilisées.....	102
a)	Numpy	102
b)	Matplotlib	102
c)	Pandas	102
d)	SeqIO	102
3.	Architecture	103
4.	Présentation des interfaces	104
a.	Login / signup.....	105
b.	Activité Profile	105
c.	Activité de traitement	106
d.	Activité historique	109
5.	Conclusion.....	109
	Conclusion et perspectives	110
	Bibliographie et Webographie	111
	Annexes	114
	Annexe 1	114
	Annexe 2 : définitions	115

Liste des figures

<i>Figure 1 : La rétrotranscription des ET de classe I</i>	16
<i>Figure 2 : La transcription des ET de classe II</i>	16
<i>Figure 3 : Classification des éléments transposables (d'après Wicker et al., 2007)</i>	17
<i>Figure 4 : Classification des éléments transposables</i>	17
<i>Figure 5 : Les rétrotransposons à LTR</i>	18
<i>Figure 6 : Les rétrotransposons non-LTR</i>	18
<i>Figure 7 : les transposons à TIR</i>	19
<i>Figure 8 : Schéma général d'un transposon.</i>	22
<i>Figure 9 : la structure de la superfamille tc1-mariner</i>	23
<i>Figure 10 : la structure de la superfamille hAT</i>	25
<i>Figure 11 : la structure de la superfamille P elemet</i>	26
<i>Figure 12 : la structure de la superfamille mudr</i>	26
<i>Figure 13 : la structure de la superfamille PiggyBac</i>	28
<i>Figure 14 : l'écosystème de l'Intelligence Artificielle</i>	30
<i>Figure 15 : types d'apprentissage automatique</i>	30
<i>Figure 16 : Un exemple une fonction discriminante apprise de manière semi supervisée sur des points exprimés dans un espace à 2 dimensions</i>	31
<i>Figure 17 : Apprentissage par Renforcement</i>	32
<i>Figure 18 : mécanisme d'algorithme Random Forest</i>	34
<i>Figure 19 : mécanisme d'algorithme Extra Trees</i>	35
<i>Figure 20 : Principe de l'algorithme SVM (deux classes)</i>	36
<i>Figure 21 : L'architecture d'une unité LSTM</i>	37
<i>Figure 22 : Reseau neuronal CNN</i>	38
<i>Figure 23 : architecture du modèle GRU</i>	38
<i>Figure 24 : architecture du modèle BiLSTM</i>	40
<i>Figure 25 : Principe de l'algorithme K-means</i>	41
<i>Figure 26 : architecture de la 1ère partie du projet : classification binaire</i>	43
<i>Figure 27 : architecture de la 2eme partie du projet : classification multi-classes</i>	45
<i>Figure 28 : architecture de la 3eme partie du projet : localisation des TIRs</i>	45
<i>Figure 29 : dataset EnT</i>	48
<i>Figure 30 : dataset ET</i>	48
<i>Figure 31 : dataset classification binaire : les ET/EnT</i>	49
<i>Figure 32 : pourcentage des ETs et EnTs dans l'ensemble de données</i>	49
<i>Figure 33 : dataset classification multiclass</i>	50
<i>Figure 34 : quantité de chaque superfamille (avant le pré-traitement)</i>	50
<i>Figure 35 : exemple de 4-mers</i>	51
<i>Figure 36 : quantité de chaque superfamille (après le pré-traitement)</i>	53

<i>Figure 37 : La methode Elbow</i>	54
<i>Figure 38 : Les centroïdes de chaque cluster</i>	54
<i>Figure 39 : logigramme pour la CB : partie ML</i>	55
<i>Figure 40 : architecture modèle choisi pour CB : LSTM / 5-mers</i>	77
<i>Figure 41 : plot Accuracy/Loss du modèle choisi pour CB : LSTM / 5-mers</i>	77
<i>Figure 42 : matrice de confusion du modèle choisi pour CB : LSTM / 5-mers</i>	78
<i>Figure 43 : logigramme pour la CM : partie ML</i>	79
<i>Figure 44 : exemple de graphe PCA pour 7-mers CV</i>	80
<i>Figure 45 : CNN& BiLSTM architecture</i>	96
<i>Figure 46 : architecture front-end / back-end</i>	103
<i>Figure 47 : diagramme de classe Front-End</i>	104
<i>Figure 48 : activités login/signup</i>	105
<i>Figure 49 : activité profile</i>	106
<i>Figure 50 : activité des classes ET/EnT</i>	107
<i>Figure 51 : activité ET SuperFamily</i>	108
<i>Figure 52 : TIRs Localisation : email envoyé</i>	108
<i>Figure 53 : activité historique</i>	109

Liste des Tableaux

<i>Tableau 1 : travaux connexes</i>	20
<i>Tableau 2 : les datasets utilisées : ETs</i>	47
<i>Tableau 3 : les datasets utilisées : EnTs</i>	47
<i>Tableau 4 : résultat CB/ML : modèle Random Forest</i>	56
<i>Tableau 5 : résultat CB/ML : modèle Extra Trees</i>	60
<i>Tableau 6 : résultat CB/ML : modèle Naive Bayes</i>	64
<i>Tableau 7 : résultat CB/ML : modèle SVM</i>	68
<i>Tableau 8 : résultat CB/DL : modèle LSTM</i>	72
<i>Tableau 9 : résultat CM/ML : modèle Random Forest</i>	81
<i>Tableau 10 : résultat CM/ML : modèle Extra Trees</i>	86
<i>Tableau 11 : résultat CM/ML : modèle Naive Bayes</i>	91
<i>Tableau 12 : résultat CM/DL : modèle Bert</i>	93
<i>Tableau 13 : résultat CM/DL : modèle CNN</i>	94
<i>Tableau 14 : résultat CM/DL : modèle CNN & BiLSTM</i>	96
<i>Tableau 15 : résultat CM/DL : modèle CNN & GRU</i>	98

Nomenclature

ET - Element transposable

EnT – Elément non transposable

ADN - Acide Désoxyribonucléique

ARN - Acide ribonucléique

MLEs - Mariner-Like Elements

MITE - Miniatures Inverse Transposable Elements

TSD - Terminal Sequence Duplicate

TIR - Terminal Inverted Repeat

LTR - Long Terminal Repeat

LINE - Long Interspersed Nuclear Elements

IA - Intelligence Artificielle

ML – Machine Learning / apprentissage automatique

DL – Deep Learning / apprentissage profond

NLP - Natural Language Processing / Traitement du Langage Naturel

NLU - Natural Language Understanding / Compréhension du Langage Naturel

NLG - Natural Language Generation / Génération du Langage Naturel

SVM - Support Vector Machine / Machine à vecteurs de support

CNN - Convolutional Neural Network / Réseau neuronal convolutif

GRU - Gated Recurrent Unit / Unité récurrente fermée

LSTM - Long Short Term Memory / Les réseaux de longue mémoire à court terme

BiLSTM - Bidirectional Long Short-Term Memory /

TFiDF - term frequency-inverse document frequency

CV - CountVectorizer

CB - Classification binaire

Introduction générale

Un élément transposable (ET) est traditionnellement décrit comme étant un fragment d'ADN possédant la particularité d'être mobile. Un tel élément est donc capable de s'insérer en d'autres endroits du génome, lui permettant ainsi de se multiplier. Ces caractéristiques en font des composants atypiques des génomes. Leur découverte a été à l'origine de fructueuses réflexions théoriques en biologie, de la régulation de l'expression des gènes jusqu'au niveau auquel la sélection naturelle est supposée agir. Pour autant, leur dynamique et leurs impacts sur les génomes ne sont toujours pas bien compris.

ET, appelé parfois **transposon**, est capable de se déplacer de manière autonome dans un génome, par un mécanisme appelé transposition. Cette transposition est rendue possible sous l'effet d'une enzyme, la transposase. Les éléments mobiles, c'est à dire les fragments d'ADN insérer dans le génome d'un organisme hôte propriété remarquable de se déplacer d'un point à un autre génome, "de transposer" à l'intérieur de la même cellule.

C'est dans cette perspective que s'inscrit ce projet bio-informatique qui a pour objectif de mener une étude et une analyse permettant d'améliorer le service de détecter les Ets et les classifier dans leurs superfamilles aussi la localisation des TIRs au sein du génome.

Ainsi, le présent rapport représente une étude détaillée, faite dans le cadre du Projet bio-informatique et qui s'articule sur quatre chapitres.

Le premier chapitre est consacré à la présentation de l'état de l'art à savoir une étude générale sur les ETs.

Puis, le deuxième chapitre comprendra une brève présentation des superfamilles qui je les travaillée avec.

Ensuite, le troisième chapitre a pour avoir une description générale des modèles ML/DL utilisés dans le projet afin de choisir les meilleurs modèles pour chaque partie du projet.

Puis, le quatrième chapitre sera dédié à l'étude technique à savoir les différentes étapes de réalisation du projet : les résultats des modèles....

Enfin, dans le dernier chapitre, nous allons entamer la partie réalisation : les choix techniques et la présentation des interfaces à partir des captures d'écran commenter.

Chapitre I : Etat de l'art

1. IA en Bio-informatique

La bio-informatique est la science de l'analyse et de l'interprétation des données biologiques à l'aide d'outils et de méthodes informatiques. Il s'agit d'un domaine en pleine croissance et en évolution qui offre de nombreuses opportunités aux bioingénieurs qui souhaitent appliquer leurs compétences et leurs connaissances pour résoudre des problèmes du monde réel.

L'IA est un terme général qui englobe diverses techniques et technologies qui permettent aux machines d'effectuer des tâches qui nécessitent normalement l'intelligence humaine, telles que l'apprentissage, le raisonnement et la prise de décision. L'IA peut être utilisée en bio-informatique pour l'exploration et l'analyse de données, l'apprentissage automatique et l'apprentissage profond, le traitement du langage naturel et l'exploration de texte, ainsi que la vision par ordinateur et l'analyse d'images.

Dans notre cas l'IA, en particulier les techniques d'apprentissage automatique et de traitement du langage naturel, permet d'analyser de grandes quantités de données génomiques pour identifier et classifier ces éléments. Les modèles IA peuvent détecter des motifs complexes dans les séquences d'ADN, facilitant ainsi la compréhension des mécanismes de mobilité des transposons et leur impact sur les génomes.

2. Historique des ETs [1]

Le génome a longtemps été considéré comme une succession de gènes, chacun associé à un locus invariant sur le chromosome. Cette vision statique a été bouleversée dans les années 1940, par Barbara Mc Clintock lorsqu'elle observa des phénotypes mutants instables chez le maïs (coloration des grains de l'épi) induits par des éléments génétiques appelés alors « éléments de contrôles ». Plus tard, McClintock démontra que ces mutations réversibles entraînent l'activation ou l'inactivation des gènes responsables de la coloration des grains de l'épi de maïs (McClintock, 1953). Elle émit, dès lors, l'hypothèse que ces mutations pouvaient être induites en réponse à des chocs / événements affectant le génome (McClintock, 1984). Cette découverte a révolutionné la notion de génome qui de statique est devenu dynamique. Longtemps considérés comme de l'ADN égoïste car ils utilisent la machinerie de l'hôte pour leur prolifération (Doolittle et Sapienza, 1980), ces « éléments de contrôle » sont aujourd'hui appelés éléments transposables (ET) et sont reconnus comme une composante quantitative des génomes (Capy et al., 1997 ; Aziz et al., 2010). De nombreuses études ont démontré leurs implications dans l'évolution et notamment les processus adaptatifs des génomes (Shapiro,

1999 ; McDonald, 1995 ; Capy et al., 2000 ; Kidwell et Lisch, 2001 ; Schmidt et Anderson, 2006 ; Böhne et al., 2008).

3. Définition

Les ET sont des courtes séquences d'ADN ayant la capacité de se déplacer dans un chromosome ou même d'un chromosome à l'autre, au sein du génome qui les héberge. Ils possèdent dans leur séquence le ou les gènes codant les enzymes nécessaires à leur déplacement encore appelé transposition. On distingue les ET autonomes qui sont mobiles de leur seul fait car ils possèdent des gènes codant des protéines fonctionnelles de transposition et les ET non-autonomes qui contiennent des gènes de transposition mutés ou délétés. Toutefois, ces-derniers peuvent être mobilisés par des éléments autonomes, ce processus est appelé mobilisation en trans.

4. Classification et variations structurales

Les ET présentent une grande diversité de structures et de mécanismes de transposition, ce qui rend leur classification complexe. Plusieurs classifications se sont succédées, la première reposant sur le mécanisme de transposition a été proposée par Finnegan (1989). Ceci a donné lieu à la répartition des ET en deux grandes classes selon que la transposition nécessitait un intermédiaire ARN (classe I ou rétrotransposon) ou non (classe II ou transposon). [2]

Les ET de classe I ou rétrotransposons regroupent tous les éléments qui transposent selon un mécanisme qualifié de « copier-coller » (Figure 1 La transposition s'effectue au moyen de la production d'un ARN rétro transcript en ADN complémentaire (ADNc). L'ADNc correspond donc à une copie qui sera ultérieurement intégrée dans un locus accepteur du génome.

Les éléments de classe II ou transposons transposent directement, sans intermédiaire, selon un mécanisme qualifié de « couper-coller » (Figure 2 Ainsi l'élément présent au sein du site donneur est excisé et réinséré dans le génome au niveau d'un site accepteur.

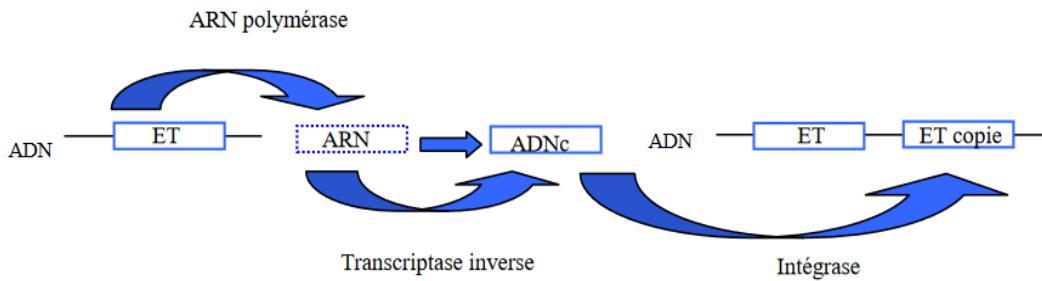


Figure 1 : La rétrotranscription des ET de classe I

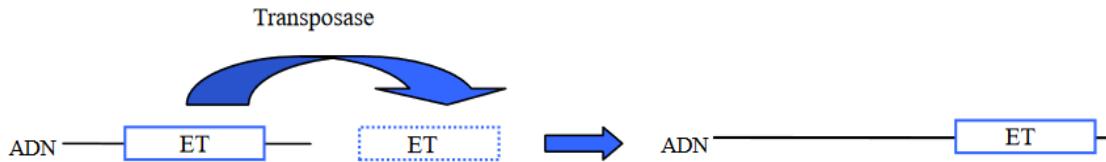
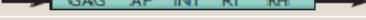
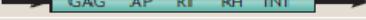
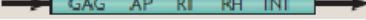
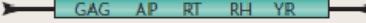
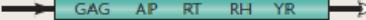
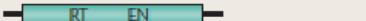
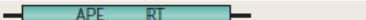
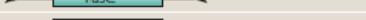
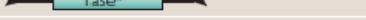
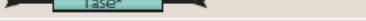
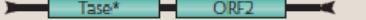
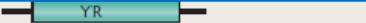


Figure 2 : La transcription des ET de classe II

5. Les superfamilles

Chaque classe est subdivisée en Ordres, Superfamilles et Familles qui peuvent coexister dans le même génome. La classe I englobe cinq ordres repartis sur dix-sept superfamilles et la classe II est divisée en deux Sous classes selon leur mode de transposition, les ETs de la sous classe I sont groupés en deux ordres et dix superfamilles et les ETs de la sous classe II sont repartis en deux ordres et deux superfamilles.

Caractérisation d'éléments transposables : les transposons à TIR

Classification		Structure	TSD
Order	Superfamily		
Class I (retrotransposons)			
LTR	Copia		4-6
	Gypsy		4-6
	Bel-Pao		4-6
	Retrovirus		4-6
	ERV		4-6
DIRS	DIRS		0
	Ngaro		0
	VIPER		0
PLE	Penelope		Variable
LINE	R2		Variable
	RTE		Variable
	Jockey		Variable
	L1		Variable
	I		Variable
SINE	tRNA		Variable
	7SL		Variable
	5S		Variable
Class II (DNA transposons) - Subclass 1			
TIR	Tc1-Mariner		TA
	hAT		8
	Mutator		9-11
	Merlin		8-9
	Transib		5
	P		8
	PiggyBac		TTAA
	PIF-Harbinger		3
	CACTA		2-3
Crypton	Crypton		0
Class II (DNA transposons) - Subclass 2			
Helitron	Helitron		0
Maverick	Maverick		6

Caractéristiques structurales

— — répétitions terminales inversées ————— région non codante —■— région codante —/— région pouvant contenir une ou plusieurs ORF

Figure 3 : Classification des éléments transposables (d'après Wicker et al. 2007)

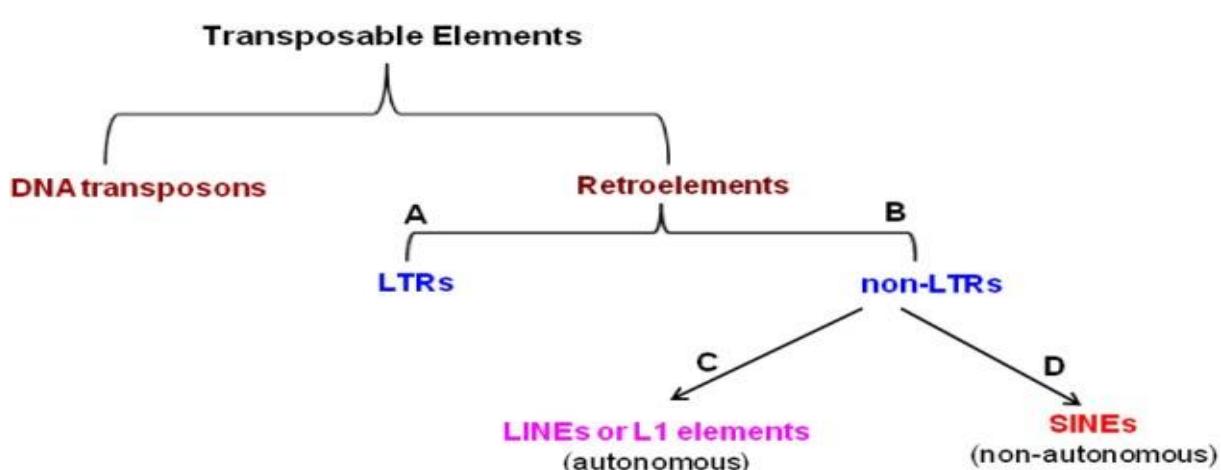


Figure 4 : Classification des éléments transposables

a) Les rétrotransposons à LTR

Les rétrotransposons à LTR peuvent représenter une part importante des génomes le plus souvent chez les plantes (Bennetzen, et al., 2005). San Miguel et Bennetzen (1998) indiquent que l'activité des rétrotransposons serait responsable du doublement voire du quadruplement du génome du maïs. Chez le coton la composante principale du génome est constituée d'été (> 79 %), parmi ceux-ci près de 62 % sont des éléments de classe I à LTR (Charles et al., 2008).

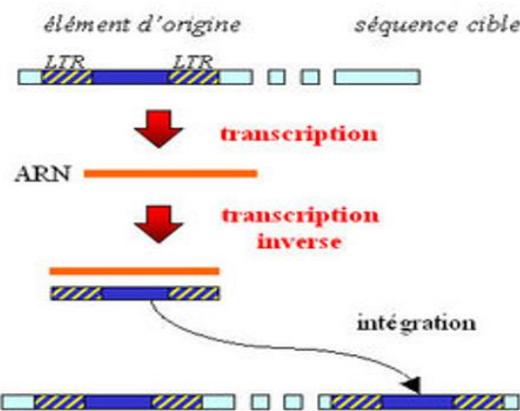


Figure 5 : Les rétrotransposons à LTR

b) Les rétrotransposons non-LTR

Les rétrotransposons non-LTR sont également nommés reproposons et sont des éléments autonomes pour la grande majorité.

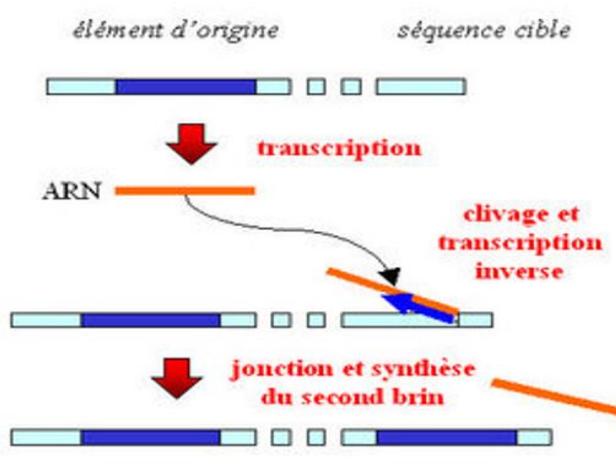


Figure 6 : Les rétrotransposons non-LTR

c) Les transposons à TIR : La Sous-classe 1 [3][4]

Les transposons à TIR sont des éléments génétiques mobiles qui appartiennent à la classe des transposons ADN. Ils sont caractérisés par la présence de séquences répétées inversées aux extrémités de l'élément, connues sous le nom de répétitions terminales inversées (TIR), ce groupe réunit tous les éléments dont la structure est composée d'une région centrale comportant une ORF codant la transposase. Les extrémités du transposon sont composées par des régions terminales inversées et répétées ou TIR. Ces séquences jouent un rôle crucial dans le processus de transposition, où l'élément est excisé de sa position actuelle et réintégré ailleurs dans le génome par un mécanisme de "couper-coller". Les transposons à TIR contiennent généralement un gène codant pour une enzyme appelée transposase, qui est responsable de l'excision et de la réintroduction de l'élément.

Les transposons à TIR sont également connus pour générer des duplications du site cible lors de leur insertion, ce qui ajoute une complexité supplémentaire au génome. Ils sont présents dans une grande diversité d'organismes, y compris les plantes, les champignons, et les animaux, et jouent un rôle important dans l'évolution du génome.

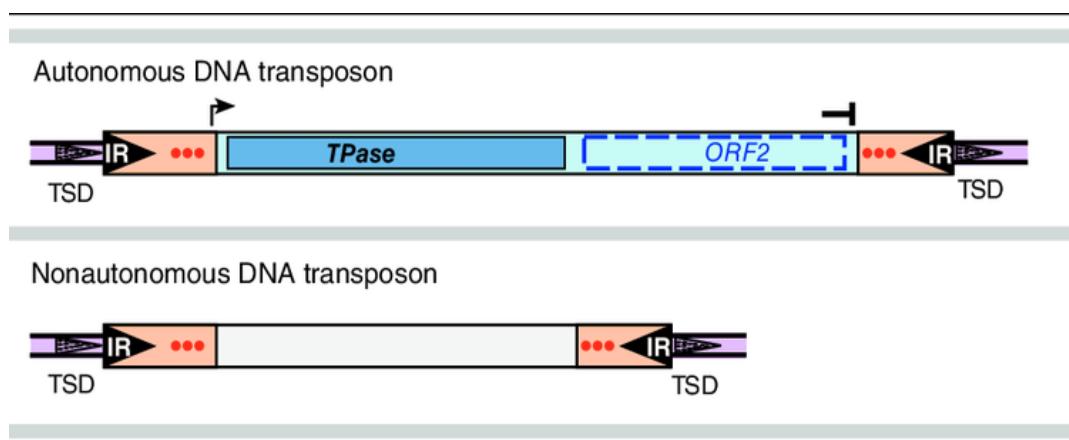


Figure 7 : les transposons à TIR

6. Travaux connexes

Tableau 1 : travaux connexes

Réf.	Auteurs	Titre	Méthodes	Année
(Orozco-Arias S, 2019) [5]	<ul style="list-style-type: none"> • Orozco-Arias S • Isaza G • Guyot R • Tabares-Soto R. 	A systematic review of the application of machine learning in the detection and classification of transposable elements	Random Forest ; Decision Trees; Support Vector Machine ...	2019 Dec 18
(Panta, 2021) [6]	<ul style="list-style-type: none"> • Panta Manisha • Mishra Avdesh • Hoque Md Tamjidul • Atallah, Joel 	ClassifyTE: a stacking-based prediction of hierarchical classification of transposable elements	stacking-based machine learning approach: combination de plusieurs modèles ML (KNN, SVM, Random Forest...)	03 March 2021
(Yan, 2020) [7]	<ul style="list-style-type: none"> • Yan Haidong • Bombarely, Aureliano • Li Song 	DeepTE: a computational method for de novo classification of transposons with convolutional neural network	convolutional neural networks (CNNs)	16 May 2020
(da Cruz & Bugatti, 2020) [8]	<ul style="list-style-type: none"> • Murilo Horacio Pereira da Cruz • Douglas Silva Domingues • Priscila Tiemi Maeda Saito • Alexandre Rossi Paschoal • Pedro Henrique Bugatti 	TERL: classification of transposable elements by convolutional neural networks	convolutional neural networks	08 September 2020

7. Cadre général du projet

a. Cahier de charge

Le projet vise à développer un système d'intelligence artificielle pour la classification et la détection des éléments transposables dans les séquences ADN. Les éléments transposables sont des segments d'ADN qui peuvent se déplacer d'un endroit à un autre au sein du génome, et leur détection et classification sont essentielles pour la compréhension de diverses fonctions biologiques et évolutives. Pour ce faire, plusieurs modèles de machine learning et de deep

learning seront explorés afin d'identifier ceux offrant les meilleures performances pour la classification des éléments transposables.

Le projet se déroule en trois grandes étapes :

1. **Classification binaire (annexe2)** : Distinguer les séquences ADN qui contiennent des éléments transposables de celles qui n'en contiennent pas.
2. **Classification multiclasse (annexe2)** : Classer les séquences contenant des éléments transposables en différents types spécifiques (par exemple, TC1 MARINER, HAT, PiggyBac, etc.).
3. **Localisation des TIR** : Identifier avec précision les positions des TIR dans les séquences d'éléments transposables.

b. Problématique

La problématique du projet réside dans la **complexité de la classification et de la détection des éléments transposables** dans les séquences ADN. Les séquences transposables peuvent présenter des motifs variés et complexes, ce qui rend leur classification et localisation un défi majeur. Le projet s'efforce donc de développer des modèles capables de traiter ces complexités, avec un objectif de précision élevée tant dans la classification binaire que multiclasse, ainsi que dans la localisation précise des TIR, ce qui est crucial pour des analyses biologiques fiables et approfondies.

8. Conclusion

Dans ce chapitre nous avons présenté des généralités sur les ETs, ainsi la présentation du cadre générale du projet.

Le prochain chapitre sera dédié à la présentation des superfamilles des ETs la classe II transposons à TIR.

Chapitre II : Les superfamilles de la classe II

1. Introduction

Selon la classification publiée par Wicker et al. en 2007, les éléments de classe II sont divisés en deux sous-classes selon la spécificité de leur mécanisme de transposition.

La sous-classe 1 est composée des éléments ayant un mécanisme de transposition classique de type « couper-coller ». La sous-classe 2 comprend les éléments Hélitrons au mécanisme de cercle roulant (Kapitonov et Jurka, 2001) et Maverick dont le processus de transposition n'est pas connu (Pritham et al., 2007).



Figure 8 : Schéma général d'un transposon.

Les flèches représentent les régions terminales inversées et la séquence centrale se compose d'un ORF codant la transcriptase.

2. Définitions et structure

a) La superfamille Tc1-Mariner [9] (ANNEXE A)

C'est la superfamille la plus répandue chez les organismes vivants, elle a été mise en évidence des protozoaires jusqu'à l'homme (pour revue : Plasterk et al., 1999 ; Shao et Tu, 2001). Elle est souvent associée aux Séquences d'Insertions (IS) bactériennes IS630 formant alors la superfamille ITm pour IS630-Tc1-Mariner (Shao et Tu, 2001). Wicker et al., 2007 propose de la dénommer superfamille DTT (pour DNA transposon, TIRgroup, Tc1-mariner). Une description plus détaillée de la superfamille Tc1-mariner fera l'objet de la partie III de ce chapitre, car les ET étudiés au cours du présent travail appartiennent à cette superfamille.

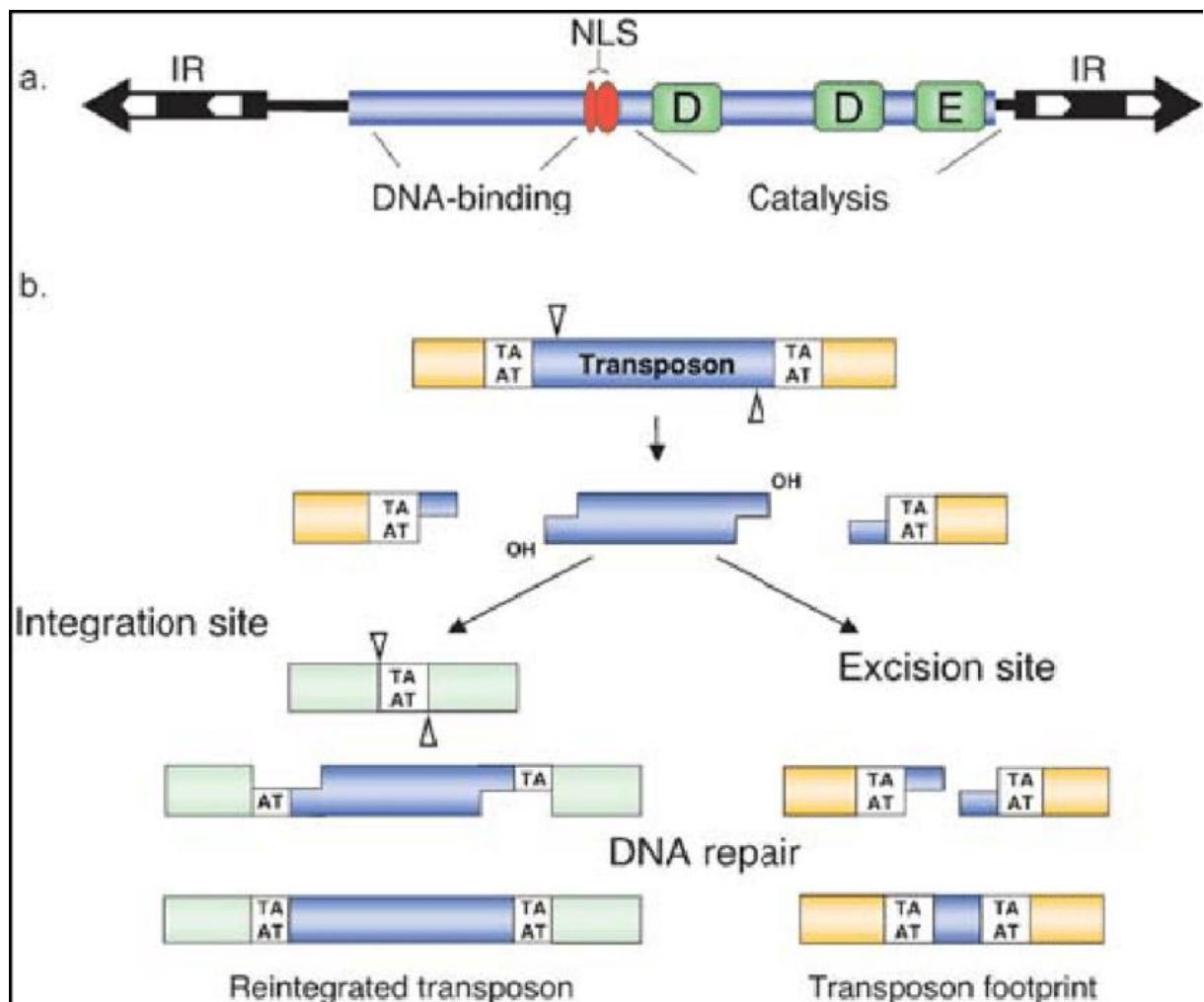


Figure 9 : la structure de la superfamille tc1-mariner

Structure et mécanisme de transposition des éléments Tc1/mariner. **(a)** Représentation schématique d'un transposon Tc1/mariner. Les répétitions inversées terminales (IR, flèches noires) contiennent un ou deux sites de liaison pour la transposase (flèches blanches). L'élément contient un seul gène codant pour la transposase (encadré bleu). La partie Nterminale de la transposase contient un domaine de liaison à l'ADN, suivi d'un signal de localisation nucléaire (NLS). La partie C-terminale de la protéine est responsable de la catalyse, y compris des réactions de clivage et de réintégration de l'ADN. La triade d'acides aminés DDE est une signature caractéristique des transposases de type Tc1 ; les marins ont DDD. **(b)** Mécanisme de transposition couper-coller. La transposase initie l'excision du transposon avec des coupes échelonnées et le réintègre au niveau d'un dinucléotide cible TA. Les lacunes simple brin au niveau du site d'intégration ainsi que les cassures de l'ADN double brin dans l'ADN du donneur sont réparées par la machinerie de réparation de l'ADN hôte. Après réparation, le TA cible est dupliqué sur le site d'intégration et une petite empreinte est laissée sur le lieu d'excision.

Cette superfamille englobe deux familles majeures :

- **Mariner-Like Éléments**, le premier élément mariner a été découvert chez la drosophile *Drosophila mauritiana* par l'analyse d'une mutation instable dans l'allèle du gène white qui se traduit par un changement de coloration au niveau de l'œil. Cet élément appelé Dmmar (pour *Drosophila mauritiana mariner*) est plus connu sous le nom de Mos1 (Jacobson et al., 1986). De- puis, les éléments apparentes à l'élément mariner ont été appelés mariner-like éléments (MLE). Les MLE ont été caractérisées chez un grand nombre d'espèces animales : insectes, nématodes, poissons, mammifères, crustacés (Plasterk et al., 1999 ; Robertson, 2002 ; Casse et al., 2006) ou encore chez le protozoaire *Trichomonas vaginalis* (Silva et al., 2005).
- **Tc1-Like éléments**, décrits pour la première fois chez *C.elegans* (Emmons et al., 1983).

b) La superfamille hAT [10] (ANNEXE A)

Le nom de cette superfamille dérive des trois familles d'éléments : hobo chez la drosophile *Drosophila melanogaster* (McGinnis et al., 1983), Activator-Dissociation (Ac/Ds) chez le maïs, *Zea mays* (McClintock, 1953), et Tam3 chez le muflier *Antirrhinum majus* (Coen et al., 1986). L'origine évolutive commune de ces trois familles a été mise en évidence par l'analyse de leurs séquences (Calvi et al., 1991). Les éléments de la superfamille hAT possèdent un gène codant une transposase de 500 à 800 acides aminés encadré par de courts TIR de 5 à 27 pb (Kempken et Windhofer, 2001). Les éléments actifs Hermes chez *Musca domestica* (Warren et al., 1994), Tol2 chez le poisson *Oryzias latipes* (Koga et al., 1996) appartiennent à cette superfamille. Les études portant sur l'activité de ces deux éléments ont montré qu'ils pouvaient être utilisés comme outil de mutagénèse en système hétérologue et en cellules eucaryotes (Evertts et al., 2007 ; Kawakami, 2007).

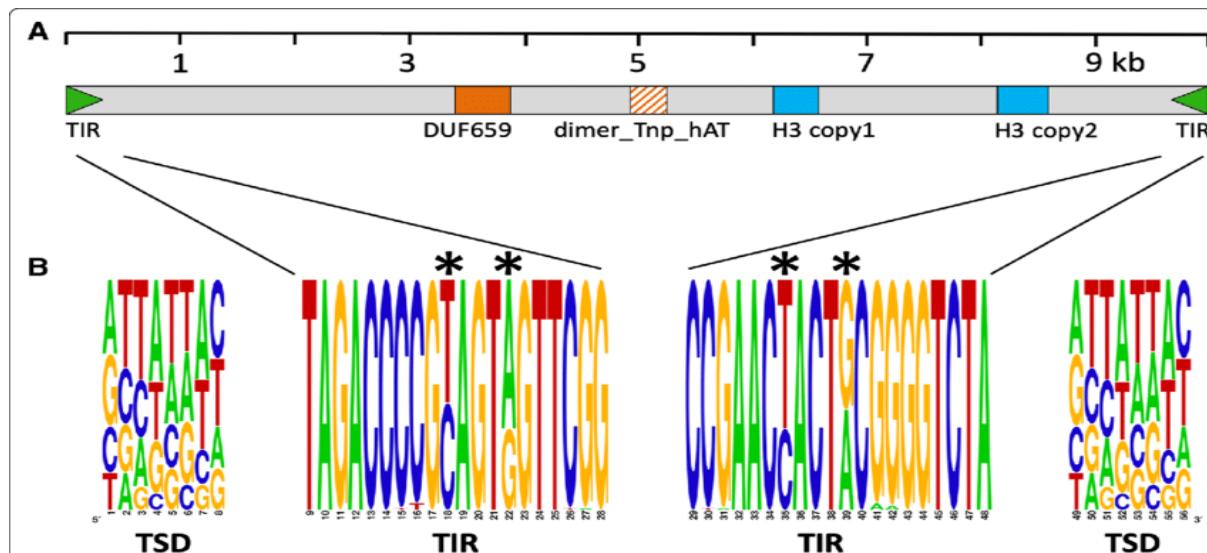


Figure 10 : la structure de la superfamille hAT

Organisation structurelle de la famille hAT d'éléments répétitifs dans le génome de *Pseudocercospora fijiensis* qui contiennent les copies de type histone H3. L'organisation relative des répétitions inversées terminales (TIR, non dessinées à l'échelle), des domaines hAT et des domaines de type histone H3 dans un élément hAT représentatif. Chaque élément contenait une à deux copies de la séquence de type H3, indiquée par les cases bleues. L'élément hAT contient le domaine DUF659, représenté par un cadre orange plein, qui se trouve dans les transposases hAT, mais le domaine « dimer_Tnp_hAT », représenté par un cadre orange avec des rayures diagonales vers le haut, n'a pas pu être identifié. La règle en haut indique la longueur de l'élément pleine longueur. Une répétition inversée terminale (TIR) de 20 pb à la fin de chaque élément complet était flanquée d'une duplication du site cible (TSD) de 8 pb dans l'ADN de l'hôte. Ces séquences consensus ont été dérivées de 99 éléments hAT complets répondant à quatre critères : longueur d'élément de ~ 9,5 kb ; présence d'un domaine hAT ; des TIR intacts ; et des TSD identiques ou presque identiques. Deux positions au sein des TIR présentaient les transitions T vers C et G vers A (indiquées par un astérisque) caractéristiques du RIP.

c) La superfamille P élément [11] (ANNEXE A)

L'élément P a été décrit la première fois chez *Drosophila melanogaster* où sa mobilité est à l'origine du phénomène de dysgénèse dans la lignée germinale des mouches hybrides (Bingham et al., 1982). D'une longueur totale de 2,9 Kpb, les éléments P sont composés de TIR de 31 pb et d'un gène codant une transposase de 87 kDa (O'Hare et Rubin, 1983). Des éléments de la superfamille des éléments P ont été identifiés chez des métazoaires (Hammer et al., 2005) et chez l'algue unicellulaire *Chlamidomonas reinhardtii* (Jurka et al., 2005). L'élément P est actif

Caractérisation d'éléments transposables : les transposons à TIR

et largement utilisé par les biologistes en tant qu'outil d'étude génomique et génétique de la drosophile (Rubin et Spradling, 1982 ; Ryder et Russell, 2003).

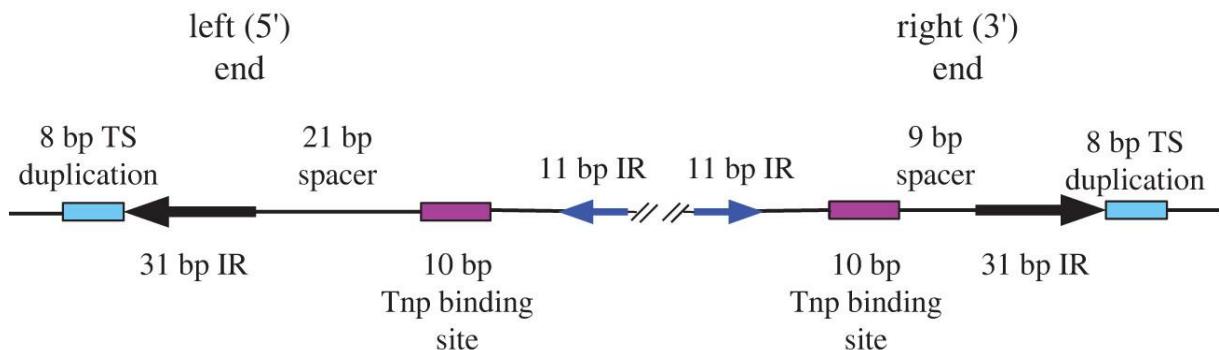


Figure 11 : la structure de la superfamille P elemet

d) La superfamille Mutator/ MuDR [12] (ANNEXE A)

Les éléments transposables de la famille Mutator, également appelés MULEs (Mutator-like elements), sont des transposons ADN caractérisés par de longues répétitions terminales inversées (TIR) et une activité de transposition élevée. Ils sont connus pour leur capacité à insérer des fragments d'ADN hôte dans leur structure, ce qui en fait des outils génétiques importants. Les MULEs jouent un rôle significatif dans l'évolution des génomes en raison de leur capacité à provoquer des mutations et à faciliter le transfert horizontal de gènes.

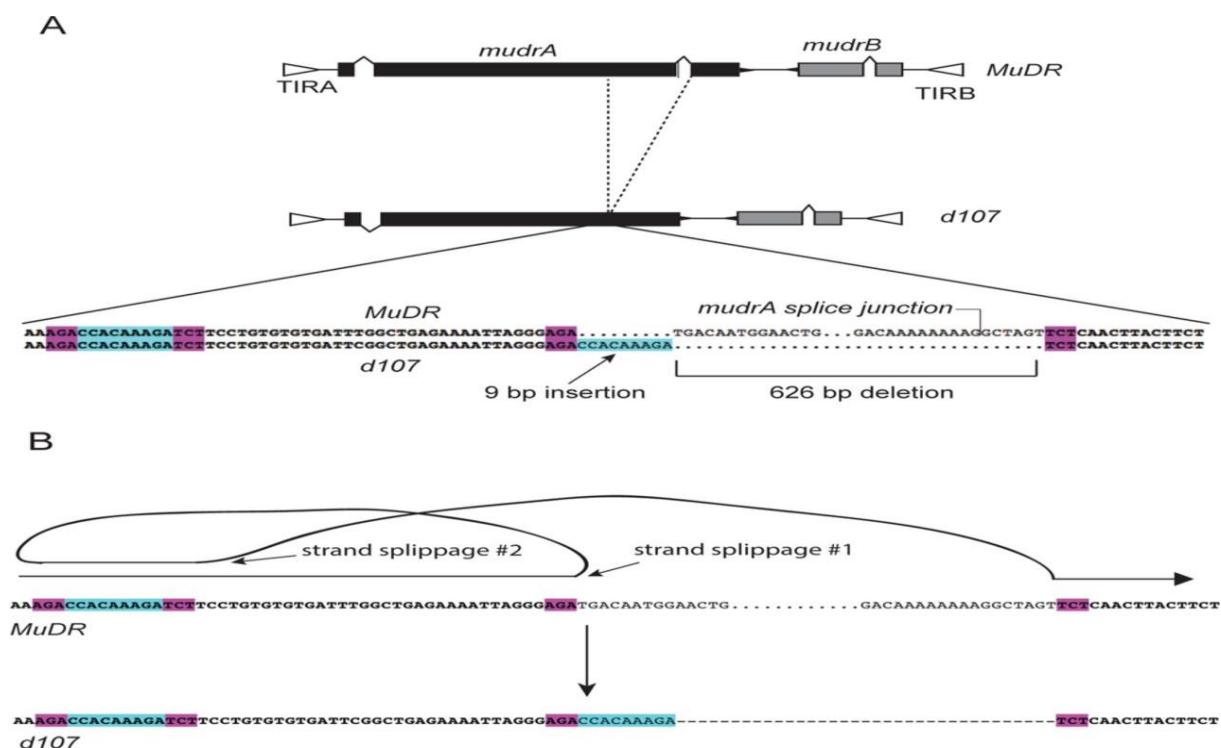


Figure 12 : la structure de la superfamille mudr

A) Structure de la séquence progénitrice (MuDR) et celle du dérivé par délétion. B) Mécanisme hypothétique de délétion. On pense que la réplication de MuDR implique la réparation des lacunes en utilisant la chromatide sœur comme modèle (Li et al. 2008). Pour générer la délétion à d107, nous émettons l'hypothèse que la réplication s'est poursuivie jusqu'à un triplet AGA, moment auquel il est supposé que le brin répliqué est passé à un deuxième triplet AGA situé 47 pb en amont. La réplication s'est ensuite poursuivie jusqu'à ce qu'elle atteigne un triplet TCT, moment auquel la réplication a basculé vers un deuxième triplet TCT situé 660 pb en aval. Le résultat net a été une délétion de 626 pb et l'insertion d'une courte séquence en amont (9 pb) à la fin de la délétion.[13]

e) La superfamille Merlin [14] (ANNEXE A)

Plusieurs nouvelles familles de transposons d'ADN ont été identifiées dans diverses espèces animales, y compris les nématodes, les vers plats, les moustiques, les ascidies, les poissons-zèbres et les humains. Ces transposons, appelés Merlin/IS1016, partagent des caractéristiques telles que des répétitions inversées terminales et des duplications du site cible de 8 ou 9 pb. Ils appartiennent à une nouvelle superfamille de transposons d'ADN, active récemment dans plusieurs espèces animales, notamment le parasite **Schistosoma mansoni**, où Merlin est la première famille de transposons d'ADN décrite.

f) La superfamille CACTA [15] (ANNEXE A)

Les éléments transposables CACTA (TEs) font partie des superfamilles les plus abondantes des transposons de Classe 2 (cut-and-paste). Bien que ces éléments aient été largement identifiés dans les plantes, les champignons, et les animaux, ils n'ont été suffisamment étudiés que dans quelques espèces modèles. Cependant, l'étude des génomes d'espèces non-modèles, comme celles du genre Chenopodium (Amaranthaceae, Caryophyllales), peut apporter des informations précieuses. Caryophyllales est une branche distincte des Angiospermes, et la diversité des éléments CACTA y était jusqu'à récemment inconnue.

g) La superfamille PiggyBac (ANNEXE A)

Le transposon PiggyBac (PB) est un élément génétique mobile qui se transpose efficacement entre vecteurs et chromosomes via un mécanisme « copier-coller ». Pendant la transposition, la transposase PB reconnaît les séquences répétées inversées (ITR) spécifiques du transposon situé aux deux extrémités du vecteur transposon et déplace efficacement le contenu des sites d'origine et les intègre dans les sites chromosomiques TTAA. La puissante activité du système de transposon PiggyBac permet aux gènes d'intérêt entre les deux ITR du vecteur PB

d'être facilement mobilisés dans les génomes cibles. Le transposon piggyBac spécifique à TTAA devient rapidement un transposon très utile pour le génie génétique d'une grande variété d'espèces, en particulier les insectes. Ils ont été découverts en 1989 par Malcolm Fraser de l'Université de Notre Dame.

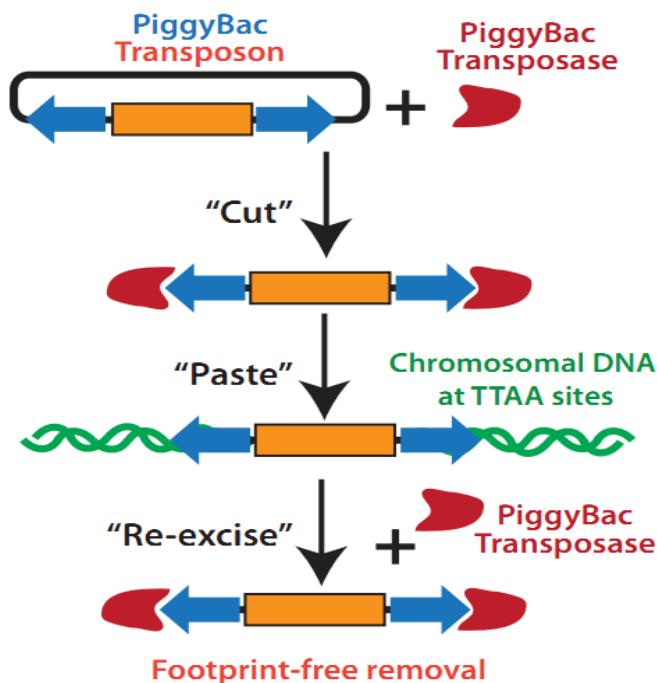


Figure 13 : la structure de la superfamille PiggyBac

3. Conclusion

Après une étude générale sur les superfamilles Tc1-Mariner, hAT, P élément, PiggyBac, CACTA, Mutator et Merlin et un bref aperçu des informations sur les éléments transposables. Le prochain chapitre sera dédié à une des matériaux et des méthodes utilisées dans l'implémentation de notre projet.

Chapitre III : Etude fonctionnelle

a. Introduction

Dans ce chapitre nous plongeons dans le monde de l'intelligence artificielle, de l'apprentissage automatique, l'apprentissage profond. Nous explorons leur évolution historique, les différences fondamentales entre l'IA, le ML et le DL, et les divers types d'apprentissage. Ce chapitre établit les bases essentielles pour notre exploration de IA dans la bio-informatique pour mettre l'existence à notre projet.

b. IA [16]

L'intelligence artificielle (IA) est un domaine de recherche qui remonte aux années 1950. Le terme "intelligence artificielle" a été introduit en 1956 lors de la conférence de Dartmouth, considérée comme le point de départ officiel de ce domaine. Les premières recherches se concentraient sur des systèmes capables de simuler des aspects de l'intelligence humaine, tels que le raisonnement, la résolution de problèmes et l'apprentissage. Les progrès en IA ont évolué au fil des décennies, passant des systèmes basés sur des règles simples aux réseaux neuronaux complexes et aux algorithmes d'apprentissage profond (Deep Learning), qui ont révolutionné le domaine.

Aujourd'hui, l'IA est omniprésente dans de nombreux secteurs, y compris la bio-informatique, où elle joue un rôle crucial dans l'analyse de grandes quantités de données génomiques, l'identification de motifs dans les séquences d'ADN, et la prédiction des fonctions biologiques. Ces avancées ont permis de mieux comprendre des phénomènes complexes, tels que l'évolution des éléments transposables dans les génomes.

Le diagramme illustre les intersections et les relations entre les domaines de la **Linguistique (annexe 2)**, de l'**Intelligence Artificielle (IA)** aussi représente visuellement les chevauchements et les distinctions entre domaines et leurs sous-domaines, mettant en évidence leurs interconnexions et les endroits où ils divergent.

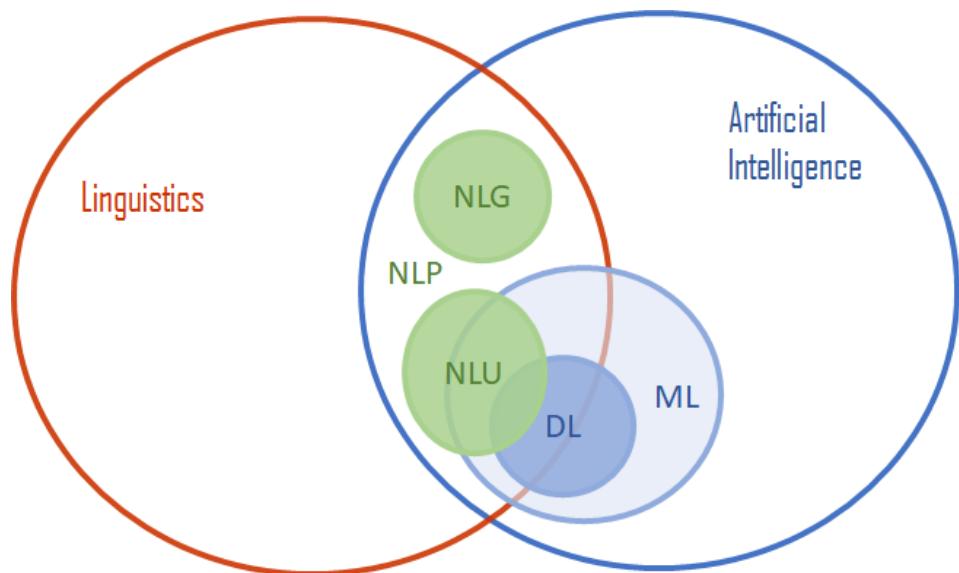


Figure 14 : l'écosystème de l'Intelligence Artificielle

c. ML

L'apprentissage automatique (Machine Learning - ML) est une sous-discipline de l'intelligence artificielle qui remonte aux années 1950. L'idée fondatrice est que les machines peuvent apprendre à partir de données plutôt que d'être explicitement programmées. Arthur Samuel, en 1959, a été l'un des pionniers en définissant le ML comme un domaine permettant aux ordinateurs d'apprendre par eux-mêmes. Au fil des décennies, le ML a évolué, passant d'algorithmes simples comme les perceptrons à des modèles plus complexes tels que les réseaux neuronaux profonds, transformant de nombreux secteurs.

a. Types d'Apprentissage

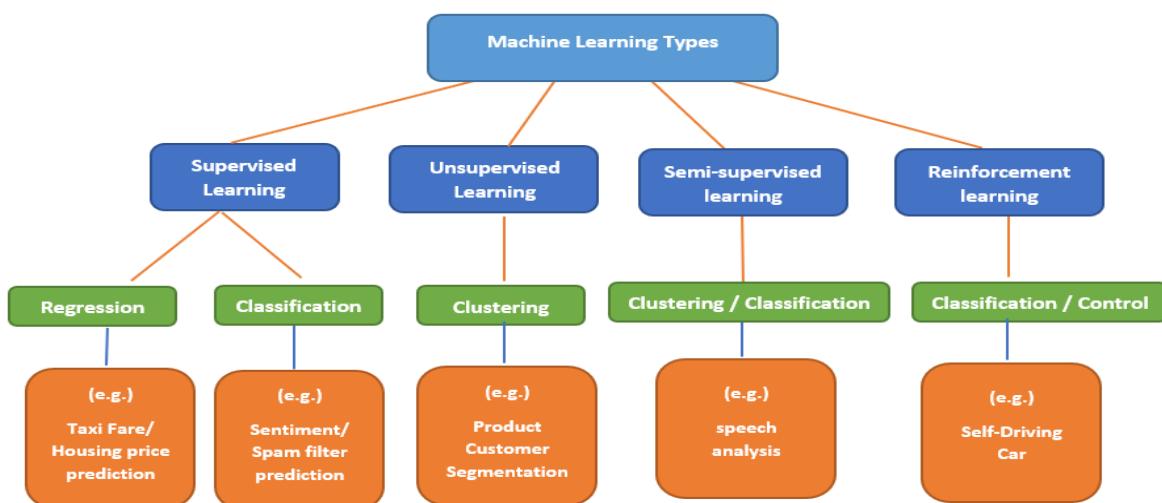


Figure 15 : types d'apprentissage automatique

a.1. Apprentissage Supervisé

L'apprentissage supervisé obtient les entrées étiquetées et les sorties souhaitées. Le but est d'apprendre une règle générale pour mapper les entrées à la sortie.

Vous allez dicter à la machine que faire afin de dégager des outputs à travers d'inputs

a.1. Apprentissage Non Supervisé

La machine obtient des entrées (inputs) sans sorties (outputs) souhaitées, le but est de trouver une structure ou des modèles pour des informations utiles dans les entrées. Extrêmement utile lorsque nous avons une très grande quantité de données sans étiquette.

a.1. Apprentissage semi-supervisé

L'apprentissage semi-supervisé consiste plutôt à fournir une information qualificative (des étiquettes de classification) ou une information quantitative (valeur réelle à des fins de régression) pour seulement un sous-ensemble des points d'entraînement soumis à l'algorithme d'apprentissage. L'algorithme peut quand même extraire de l'information sur la nature dit ou des phénomène(s) observé(s) à partir des vecteurs qui sont fournis sans information de qualification ou de quantification.

Dans la figure 16 on peut voir une surface de décision apprise à des fins de classification étant donné certains points étiquetés comme appartenant à la première ou à la deuxième classe et certains points dont l'étiquette n'est pas révélée.

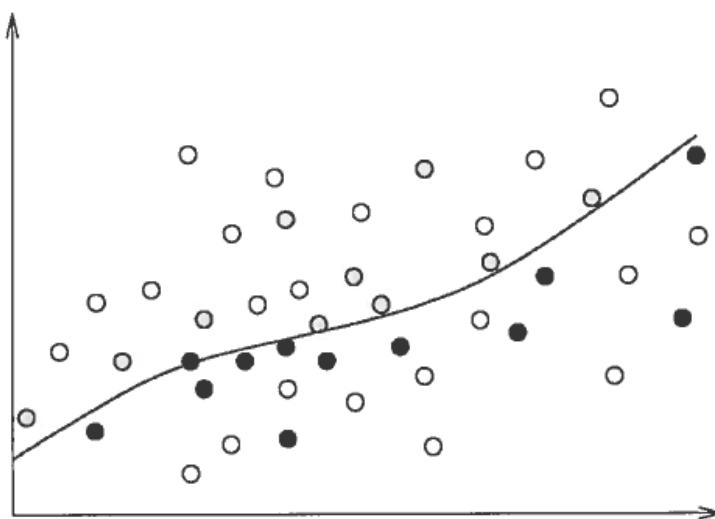


Figure 16 : Un exemple une fonction discriminante apprise de manière semi supervisée sur des points exprimés dans un espace à 2 dimensions

a.1. Apprentissage par Renforcement

Dans cet algorithme, votre agent interagit avec un environnement dynamique et il doit atteindre un certain objectif sans guide ni enseignant en échange de "reward/récompense" dans le cas de bon résultat.

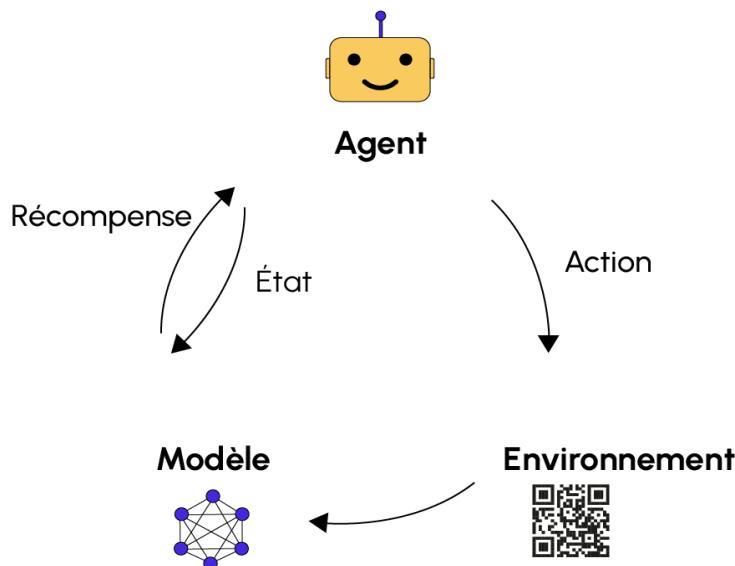


Figure 17 : Apprentissage par Renforcement

d. DL

L'apprentissage profond (Deep Learning - DL) est une branche de l'apprentissage automatique qui a pris son essor au début des années 2010. Il repose sur l'utilisation de réseaux neuronaux profonds, composés de plusieurs couches de neurones artificiels. Bien que les concepts fondamentaux des réseaux neuronaux remontent aux années 1980, ce n'est que grâce aux avancées en puissance de calcul et à la disponibilité de grandes quantités de données que le DL a connu un développement spectaculaire. Aujourd'hui, le DL est utilisé pour des tâches complexes telles que la reconnaissance d'images, le traitement du langage naturel, et l'analyse de données biologiques.

e. NLP [17]

Le Traitement du Langage Naturel (NLP) se concentre sur la manière de permettre aux ordinateurs d'interpréter, de comprendre et de manipuler les langues humaines. Traditionnellement, l'interaction entre les humains et les ordinateurs se fait par le biais d'un langage de programmation. Cependant, lorsqu'il s'agit d'interaction entre le langage humain et la machine, cette tâche devient très difficile, car le langage humain est extrêmement ambigu, contient des expressions familières avec des significations inhabituelles et intègre des contextes

sociaux. La tâche devient encore plus complexe lorsque l'accent est pris en compte, car les personnes de différentes régions ont des accents différents.[9]

Le NLP comprend deux tâches principales : l'analyse syntaxique et l'analyse sémantique.

5.1. NLU

NLU est un sous-ensemble du Traitement du Langage Naturel (NLP) qui se concentre sur la compréhension du sens d'une phrase en utilisant l'analyse syntaxique et sémantique du texte. Comprendre la syntaxe fait référence à la structure grammaticale de la phrase, tandis que la sémantique se concentre sur la compréhension du sens réel de chaque mot. [9][10][11]

5.2. NLG

NLG, comme son nom l'indique, permet aux systèmes informatiques de rédiger, en générant du texte. Elle se concentre sur la génération d'une réponse textuelle en langue humaine basée sur certaines données d'entrée. À l'origine, les systèmes de NLG utilisaient plusieurs modèles préétablis pour générer du texte. Sur la base de certaines données d'entrée ou requêtes, les systèmes de NLG généraient le texte. Cependant, le texte généré suivait un modèle typique. Néanmoins, avec l'augmentation de la puissance de calcul, des données textuelles disponibles et l'émergence de nouvelles technologies d'apprentissage profond, ces modèles de NLG sont devenus très puissants. [18][19][20]

5.3. Applications

5.3.1. Reconnaissance vocale

C'est le processus de conversion des données vocales en données textuelles. La partie difficile de la reconnaissance vocale réside dans le fait que chaque personne a une manière différente de parler, avec des accents et des argots différents.

5.3.2. Reconnaissance des entités nommées (NER)

C'est le processus de marquage des mots utiles comme entités. Par exemple : des mots tels que Inde, États-Unis sont étiquetés comme des lieux, et des mots comme Fred sont étiquetés comme des noms.

5.3.3. Chat GPT

L'outil de traitement du langage naturel le plus connu est GPT-3, de OpenAI, qui utilise l'intelligence artificielle et des statistiques pour prédire le mot suivant dans une phrase en se basant sur les mots précédents. Les praticiens du NLP appellent ces outils des « modèles de langage », et ils peuvent être utilisés pour des tâches d'analyse simples, comme la classification

de documents et l'analyse du sentiment dans des blocs de texte, ainsi que pour des tâches plus avancées, comme répondre à des questions et résumer des rapports. Les modèles de langage transforment déjà l'analyse traditionnelle des textes, mais GPT-3 a été un modèle de langage particulièrement crucial car, étant 10 fois plus grand que tout modèle précédent lors de sa sortie, il a été le premier grand modèle de langage, ce qui lui a permis de réaliser des tâches encore plus avancées comme la programmation et la résolution de problèmes mathématiques de niveau lycée. La dernière version, appelée InstructGPT, a été affinée par des humains pour générer des réponses beaucoup mieux alignées avec les valeurs humaines et les intentions des utilisateurs, et le dernier modèle de Google montre des avancées impressionnantes supplémentaires en matière de langage et de raisonnement.

f. Généralité sur les modèles ML/DL utilisés

6.1. Modèles ML

a. Random Forest

L'algorithme de la forêt aléatoire est une extension de la méthode de l'ensachage (bagging), car il utilise à la fois l'ensachage et la randomisation des caractéristiques pour créer une forêt non corrélée d'arbres de décision. La randomisation des caractéristiques, également appelée ensachage des caractéristiques ou "méthode des sous-espaces aléatoires", génère un sous-ensemble aléatoire de caractéristiques, ce qui assure une faible corrélation entre les arbres de décision. C'est une différence clé entre les arbres de décision et les forêts aléatoires. Alors que les arbres de décision considèrent toutes les divisions possibles des caractéristiques, les forêts aléatoires ne sélectionnent qu'un sous-ensemble de ces caractéristiques.[21]

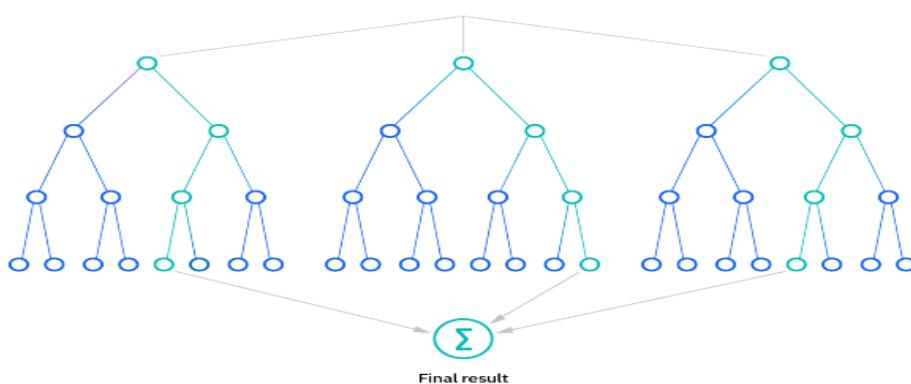


Figure 18 : mécanisme d'algorithme Random Forest

b. Extra Trees [22][23]

Similaire aux forêts aléatoires (Random Forests), ExtraTrees est une approche d'apprentissage automatique ensembliste qui entraîne de nombreux arbres de décision et agrège les résultats du

groupe d'arbres de décision pour produire une prédiction. Cependant, il existe quelques différences entre Extra Trees et Random Forest.

Random Forest utilise le bagging pour sélectionner différentes variations des données d'entraînement afin de garantir que les arbres de décision sont suffisamment différents. En revanche, Extra Trees utilise l'ensemble du jeu de données pour entraîner les arbres de décision. Pour assurer des différences suffisantes entre les arbres de décision individuels, Extra Trees SÉLECTIONNE ALÉATOIUREMENT les valeurs auxquelles diviser une caractéristique et créer des nœuds enfants. En revanche, dans une forêt aléatoire, un algorithme est utilisé pour effectuer une recherche gloutonne et sélectionner la valeur à laquelle diviser une caractéristique. Mis à part ces deux différences, Random Forest et Extra Trees sont en grande partie identiques.

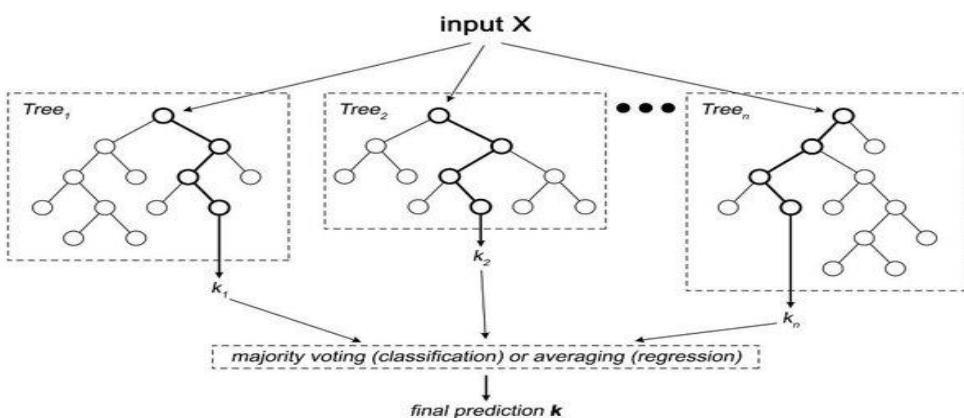


Figure 19 : mécanisme d'algorithme Extra Trees

c. Naïve Bayes

Le classificateur Bayes naïf est également considéré comme un classificateur probabiliste, car il est basé sur le théorème de Bayes. Il serait difficile d'expliquer cet algorithme sans expliquer les bases de la statistique bayésienne. Ce théorème, également connu sous le nom de règle de Bayes, nous permet « d'inverser » les probabilités conditionnelles. Pour rappel, les probabilités conditionnelles représentent la probabilité qu'un événement se produise sachant qu'un autre événement a eu lieu, ce qui est représenté par la formule suivante :

$$P(Y|X) = \frac{P(X \text{ and } Y)}{P(X)}$$

d. SVM [24]

Les SVMs sont une famille d'algorithmes d'apprentissage automatique qui permettent de résoudre des problèmes tant de classification que de régression ou de détection d'anomalie.

Le principe des SVM consiste à ramener un problème de classification ou de discrimination à un **hyperplan** (*feature space*) dans lequel les données sont **séparées en plusieurs classes** dont la frontière est la plus éloignée possible des points de données (ou *marge maximale*).

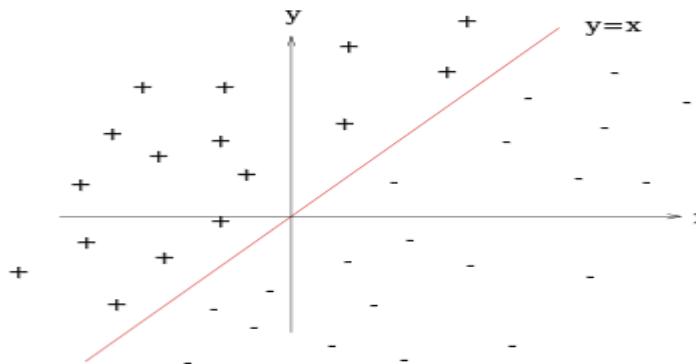


Figure 20 : Principe de l'algorithme SVM (deux classes)

6.2. Modèles DL

a. LSTM

Le Long Short-Term Memory (LSTM) est une version améliorée des réseaux de neurones récurrents, conçue par Hochreiter & Schmidhuber.

Dans un RNN traditionnel, un seul état caché est transmis au fil du temps, ce qui peut rendre difficile l'apprentissage des dépendances à long terme par le réseau. Les architectures LSTM résolvent ce problème en introduisant une cellule mémoire, qui est un conteneur capable de conserver des informations sur une période prolongée.

a.1. Architecture [25]

L'architecture des LSTM implique une cellule mémoire qui est contrôlée par trois portes : la porte d'entrée, la porte d'oubli et la porte de sortie. Ces portes déterminent quelles informations ajouter, supprimer et sortir de la cellule mémoire.

- **Cellule mémoire** : Le cœur de l'unité LSTM, où les informations peuvent être stockées sur de longues périodes. La cellule mémoire peut ajouter ou supprimer des informations en fonction des besoins

- **La porte d'entrée/ input gate** contrôle quelles informations sont ajoutées à la cellule mémoire.
- **La porte d'oubli/ forget gate** contrôle quelles informations sont supprimées de la cellule mémoire.
- **La porte de sortie/ output gate** contrôle quelles informations sont sorties de la cellule mémoire.

Cela permet aux réseaux LSTM de retenir ou de supprimer sélectivement des informations au fur et à mesure qu'elles traversent le réseau, ce qui leur permet d'apprendre des dépendances à long terme.

Le LSTM maintient un état caché, qui agit comme la mémoire à court terme du réseau. Cet état caché est mis à jour en fonction de l'entrée, de l'état caché précédent et de l'état actuel de la cellule mémoire.

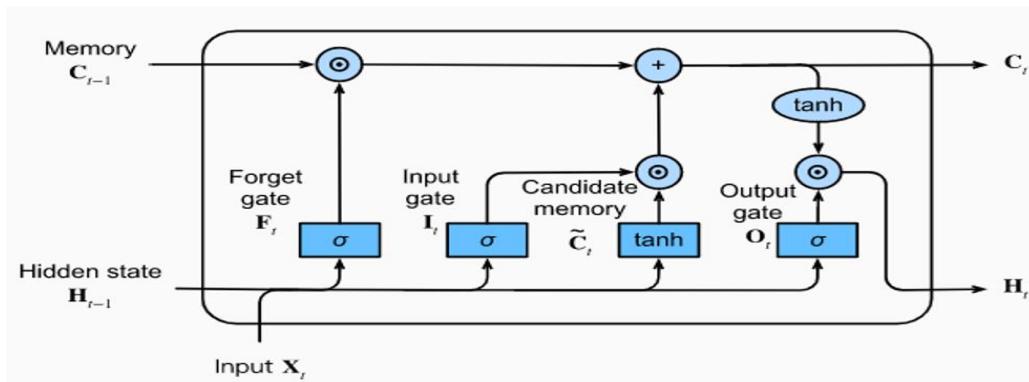


Figure 21 : L'architecture d'une unité LSTM

b. CNN

Les réseaux de neurones convolutifs ont une méthodologie similaire à celle des méthodes traditionnelles d'apprentissage supervisé : ils reçoivent des images en entrée, détectent les *features* de chacune d'entre elles, puis entraînent un classifieur dessus.

Techniquement, l'apprentissage profond des modèles CNN pour former et tester, chaque image d'entrée la fera passer à travers une série de couches de convolution avec filtres (Kernels), Pooling, couches entièrement connectées (FC) et appliquera la fonction Softmax pour classer un objet avec des valeurs probabilistes comprises entre 0 et 1. La figure ci-dessous est un flux complet de CNN pour traiter une image d'entrée et classe les objets en fonction des valeurs.

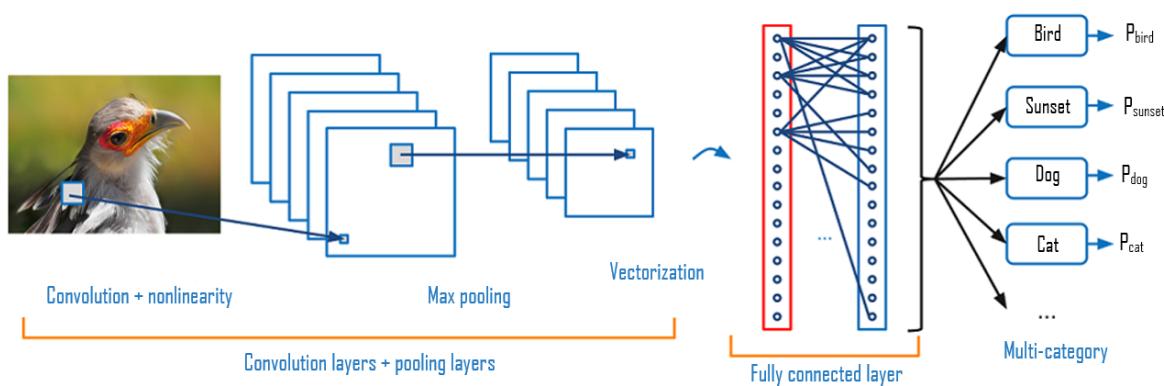


Figure 22 : Réseau neuronal CNN

La figure suivante montre le graphique du réseau neuronal de CNN pour bien comprendre l'architecture de notre modèle.

b.1. CNN et GRU [26]

Comme le LSTM, le GRU est conçu pour modéliser les données séquentielles.

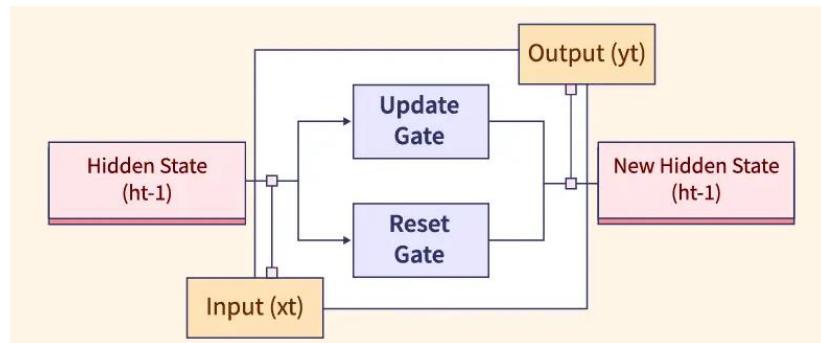


Figure 23 : architecture du modèle GRU

L'architecture GRU se compose des éléments suivants :

- **Couche d'entrée :** La couche d'entrée reçoit des données séquentielles, telles qu'une séquence de mots ou une série temporelle de valeurs, et les transmet au GRU.
- **Couche cachée :** La couche cachée est le lieu où se produit le calcul récurrent. À chaque étape temporelle, l'état caché est mis à jour en fonction de l'entrée actuelle et de l'état caché précédent. L'état caché est un vecteur de nombres qui représente la « mémoire » du réseau des entrées précédentes.
- **Porte de réinitialisation :** La porte de réinitialisation détermine la quantité de l'état caché précédent à oublier. Elle prend en entrée l'état caché précédent et l'entrée actuelle, et produit un vecteur de nombres compris entre 0 et 1, qui contrôle le degré auquel l'état caché précédent est « réinitialisé » à l'étape temporelle actuelle.

- **Porte de mise à jour** : La porte de mise à jour détermine la quantité du vecteur d'activation candidat à incorporer dans le nouvel état caché. Elle prend en entrée l'état caché précédent et l'entrée actuelle, et produit un vecteur de nombres compris entre 0 et 1, qui contrôle le degré auquel le vecteur d'activation candidat est incorporé dans le nouvel état caché.
- **Vecteur d'activation candidat** : Le vecteur d'activation candidat est une version modifiée de l'état caché précédent qui est « réinitialisé » par la porte de réinitialisation et combiné avec l'entrée actuelle. Il est calculé à l'aide d'une fonction d'activation tanh qui écrase sa sortie entre -1 et 1.
- **Couche de sortie** : La couche de sortie prend l'état caché final en entrée et produit la sortie du réseau. Cela peut être un nombre unique, une séquence de nombres, ou une distribution de probabilité sur des classes, selon la tâche à accomplir.

La principale différence entre le GRU et le LSTM réside dans la manière dont ils gèrent l'état de la mémoire. Dans le LSTM, l'état de la mémoire est maintenu séparément de l'état caché et est mis à jour à l'aide de trois portes : la porte d'entrée, la porte de sortie et la porte d'oubli. Dans le GRU, l'état de la mémoire est remplacé par un « vecteur d'activation candidat », qui est mis à jour à l'aide de deux portes : la porte de réinitialisation et la porte de mise à jour.

CNN et GRU est une combinaison puissante utilisée pour traiter des données séquentielles tel que la CNN est utilisée pour extraire des caractéristiques locales ou spatiales à partir des données d'entrée, en appliquant des filtres de convolution qui capturent les motifs locaux. Ces caractéristiques extraites sont ensuite réduites en dimension via des couches de pooling, ce qui permet de résumer l'information tout en conservant les caractéristiques essentielles.

Après l'extraction des caractéristiques par la CNN, les sorties sont généralement organisées en une séquence, qui est ensuite passée à un GRU. Le GRU, conçu pour traiter les données séquentielles, prend cette séquence de caractéristiques et modélise les dépendances temporelles ou séquentielles entre les différentes étapes de la séquence.

b.2. CNN et BiLSTM

Un Bidirectional Long Short-Term Memory (BiLSTM) est une extension des LSTM qui traite les séquences de données dans les deux directions, c'est-à-dire de gauche à droite et de droite à gauche. Cela permet au modèle de capturer des informations contextuelles des deux côtés de chaque point de la séquence.

En résumé, le BiLSTM ajoute une couche LSTM supplémentaire qui inverse la direction du flux d'information. Concrètement, cela signifie que la séquence d'entrée est traitée dans le sens inverse dans cette couche LSTM supplémentaire. Ensuite, nous combinons les sorties des deux couches LSTM de plusieurs manières, telles que la moyenne, la somme, la multiplication, ou la concaténation.

Pour illustrer, le BiLSTM déroulé est présenté dans la figure ci-dessous :

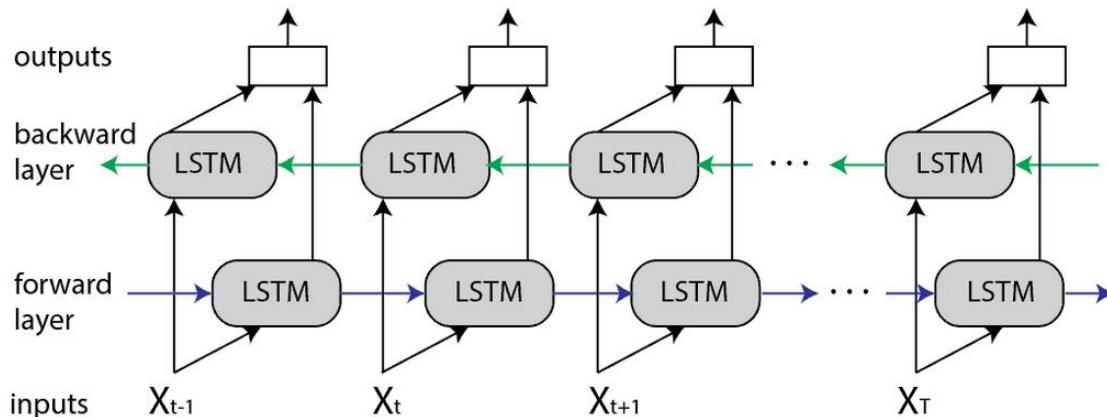


Figure 24 : architecture du modèle BiLSTM

Dans notre cas nous avons combiné CNN avec BiLSTM, pour que la CNN est utilisée en première étape pour extraire des caractéristiques locales ou spatiales des données d'entrée.

Une fois que la CNN a extrait les caractéristiques pertinentes, celles-ci sont structurées en une séquence de vecteurs, qui sont ensuite passés à un BiLSTM. Le BiLSTM est une variante du LSTM classique qui traite les données séquentielles dans les deux directions (avant et arrière). Cette capacité bidirectionnelle permet au BiLSTM de capturer les dépendances temporelles passées et futures dans les données, offrant ainsi une compréhension plus complète des relations séquentielles.

c. DNABert [27][28]

BERT est un modèle de représentation de langage contextualisé basé sur les transformateurs qui a atteint des performances surhumaines dans de nombreuses tâches de traitement du langage naturel (NLP). Il introduit un paradigme de pré-entraînement et de fine-tuning, qui développe d'abord des compréhensions à usage général à partir d'une grande quantité de données non étiquetées, puis résout diverses applications avec des données spécifiques à la tâche avec une modification architecturale minimale. DNABERT suit le même processus d'entraînement que BERT. Plus de détails sont inclus dans le matériel supplémentaire.

DNABERT commence par prendre un ensemble de séquences représentées sous forme de tokens k-mer en entrée (Figure 35 Chaque séquence est représentée sous forme de matrice en intégrant chaque token dans un vecteur numérique.

DNABERT, similaire à BERT, utilise un schéma de pré-entraînement et de fine-tuning, mais avec des modifications adaptées aux séquences ADN. Le processus de pré-entraînement de DNABERT a été ajusté en supprimant la prédiction de la phrase suivante et en forçant le modèle à prédire des séquences de tokens contigus adaptées à l'ADN. Le modèle apprend les bases de la syntaxe et de la sémantique de l'ADN à partir de séquences du génome humain, puis est affiné pour des tâches spécifiques.

6.3. K-means

K-means est un algorithme de clustering non supervisé conçu pour partitionner des données non étiquetées en un certain nombre (c'est-à-dire le « K ») de groupes distincts. En d'autres termes, k-means trouve des observations qui partagent des caractéristiques importantes et les classe ensemble en groupes. Une bonne solution de clustering est celle qui trouve des clusters tels que les observations au sein de chaque cluster sont plus similaires que les clusters eux-mêmes.

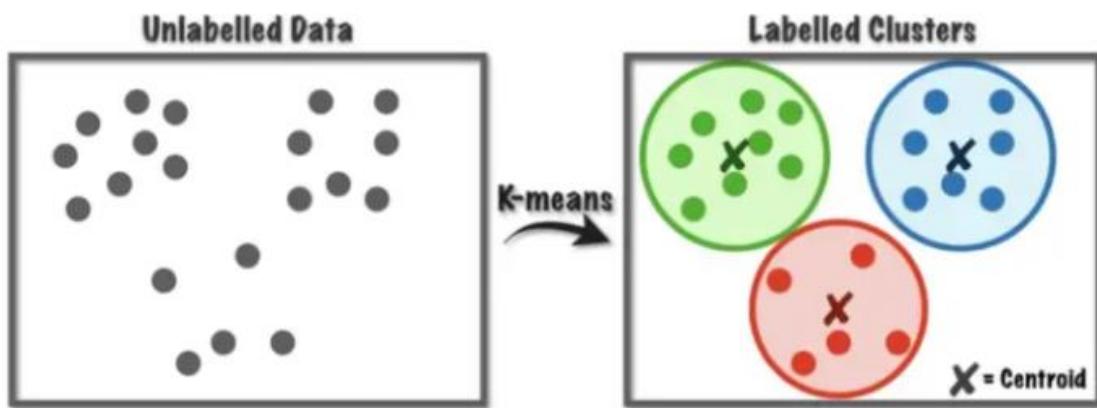


Figure 25 : Principe de l'algorithme K-means

L'intuition derrière l'algorithme est en fait assez simple. Pour commencer, nous choisissons une valeur pour k (le nombre de clusters) et choisissons au hasard un centroïde initial (coordonnées centrales) pour chaque cluster. Nous appliquons ensuite un processus en deux étapes :

- 1- Etape d'affectation — Assignez chaque observation au centre le plus proche.

2- Etape de mise à jour — Mettez à jour les centroïdes comme étant le centre de leur observation respective.

Nous répétons ces deux étapes encore et encore jusqu'à ce qu'il n'y ait plus de changement dans les clusters. A ce stade, l'algorithme a convergé et nous pouvons récupérer nos clusters finaux.

d. Conclusion

Dans ce chapitre nous avons présenté l'ensembles des modèles ML/DL utilisés.

Dans le chapitre qui suit, nous allons présenter les opérations qui seront effectuées lors de la réalisation du projet.

Chapitre IV : Etude technique

1. Introduction

La bio-informatique et les disciplines liées à la biologie ne sont pas en reste dans la révolution. Avant l'émergence de l'apprentissage automatique, ces disciplines étaient confrontées au problème de l'extraction d'informations précieuses à partir de grands ensembles de données biologiques. Mais à partir d'aujourd'hui, les techniques de ML telles que l'apprentissage en profondeur peuvent apprendre les fonctionnalités d'ensembles de données complexes et les présenter d'une manière facile à comprendre. Dans ce chapitre, nous visons à expliquer l'approche adoptée pour la détection et la classification des éléments transposable et sa superfamille en citant les différentes étapes suivies lors de la réalisation de ce projet.

2. Architecture du projet

Ce projet vise à développer un système d'intelligence artificielle capable de classifier les séquences ADN en éléments transposables et non transposables Classification Binaire (Annexe2), puis de déterminer le type d'élément transposable parmi plusieurs classes spécifiques Classification Multi-classes (Annexe2), et enfin de localiser précisément les TIRs dans la séquence d'entrée.

2.1. Partie 1 : classification binaire

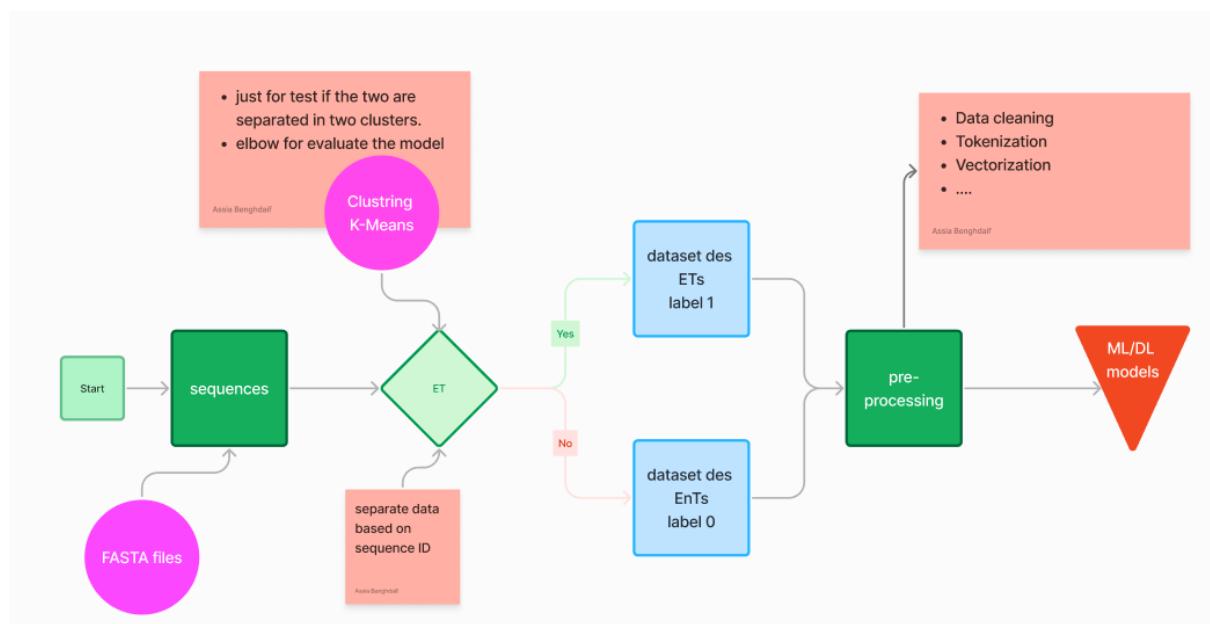


Figure 26 : architecture de la 1ère partie du projet : classification binaire

L'image représente un schéma de flux de travail pour la détection des éléments transposables (ET) dans un génome à partir de fichiers FASTA. Voici une description des différentes étapes illustrées dans le schéma :

1. **Début (Start)** : Le processus commence par l'acquisition des séquences génomiques à partir de fichiers FASTA.
2. **Séquences** : Les séquences génomiques extraites sont ensuite analysées pour identifier la présence d'éléments transposables (ET).
3. **Clustering K-Means** : Une étape de clustering K-Means est utilisée pour tester si les séquences peuvent être séparées en deux clusters distincts.
4. **Séparation des données (ET)** : Si les séquences identifiées contiennent des ET, elles sont séparées en deux ensembles de données distincts :
 - **Dataset des ETs (label 1)** : Les séquences contenant des éléments transposables sont étiquetées avec le label 1.
 - **Dataset des EnTs (label 0)** : Les séquences ne contenant pas d'éléments transposables sont étiquetées avec le label 0.
5. **Pré-traitement (Pre-processing)** : Les ensembles de données sont ensuite pré-traités, ce qui inclut des étapes telles que le nettoyage des données, la tokenization, et la vectorisation.
6. **Modèles ML/DL** : Les données prétraitées sont finalement utilisées pour entraîner des modèles de machine learning (ML) ou de deep learning (DL) pour la détection et la classification des éléments transposables.

Ce flux de travail illustre un processus complet de traitement des données génomiques pour la détection des éléments transposables, intégrant à la fois des techniques de clustering et des modèles d'apprentissage.

2.2. Partie 2 : classification multi-classes

Caractérisation d'éléments transposables : les transposons à TIR

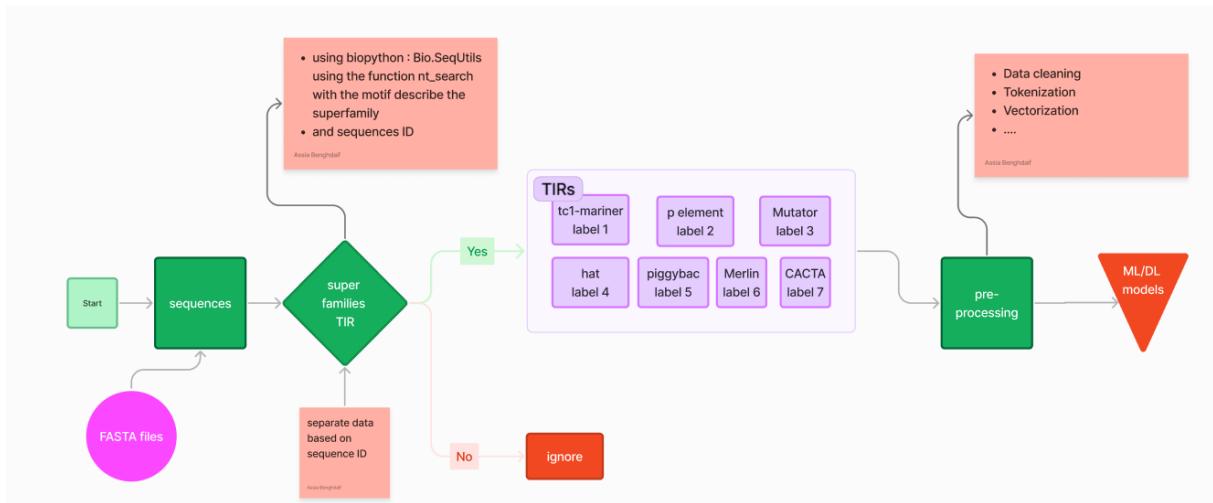


Figure 27 : architecture de la 2eme partie du projet : classification multi-classes

Ce schéma illustre un processus détaillé pour identifier et classer les superfamilles transposons à TIR dans les séquences génomiques, en intégrant des techniques bio-informatiques et des modèles d'IA pour une analyse avancée.

2.3. Partie 3 : Localisation des TIRs

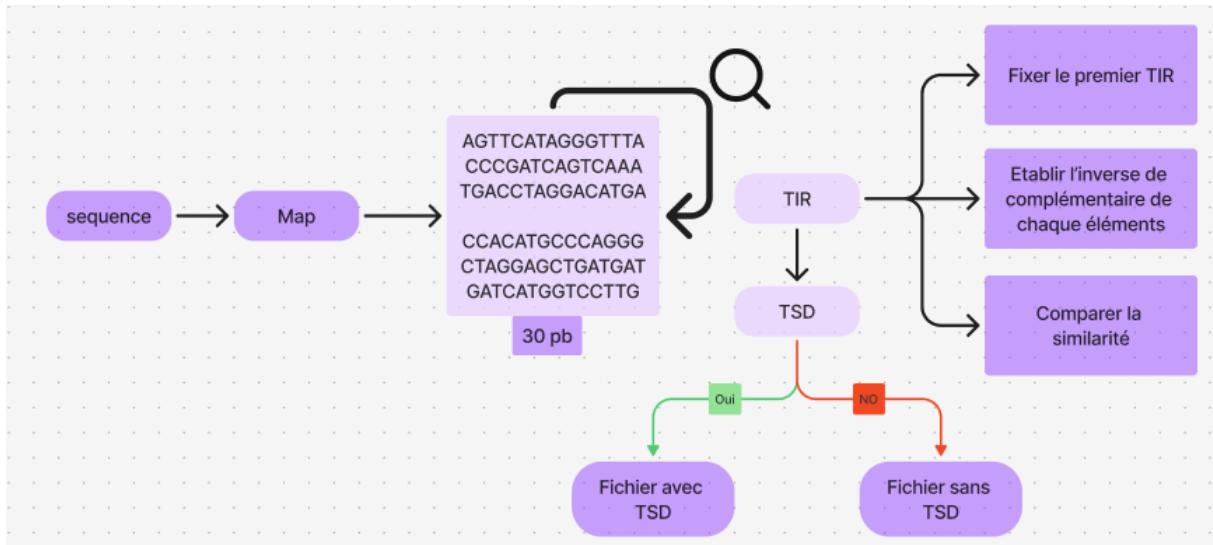


Figure 28 : architecture de la 3eme partie du projet : localisation des TIRs

Il était nécessaire d'éliminer les descriptions de chaque scaffold avant de passer à la phase de détection des TIRs supposés.

Ensuite, nous avons découpé le génome en portions de 30 nucléotides (TIR), en avançant d'un seul pas (nucléotide). Le résultat a été stocké dans une map clé-valeur, où chaque clé indique la position du premier nucléotide du TIR dans le génome et la valeur représente la séquence du TIR récupérée. Après avoir récupéré la map des TIRs, nous avons vérifié pour chaque TIR s'il

existe l'inverse de son complémentaire dans la map, avec un seuil de similarité spécifié. Cependant, étant donné que le nombre de mutations dans la famille TC1-mariner peut aller de 2 à 28, il est difficile de les générer toutes, donc ce seuil de similarité est plus optimal.

Cette similarité consiste à vérifier si la première extrémité du TIR et son inverse complémentaire présentent une ressemblance. Si cette ressemblance existe, nous passons à une deuxième phase de vérification qui teste s'il y a la même séquence de TSD à l'extrémité de chaque couple de TIR. Si la condition est bien vérifiée, nous stockons l'élément et sa description dans un fichier ; sinon, nous éliminons l'élément en le déplaçant vers un autre fichier contenant des éléments transposables supposés sans TSD.

3. Collecte de données

La collecte de données est une phase cruciale dans toute recherche ou étude empirique. Elle consiste à rassembler, mesurer et analyser des informations précises à l'aide de techniques standardisées et validées. Cette procédure permet de recueillir des informations qui seront ensuite analysées pour confirmer ou réfuter des hypothèses. Les données peuvent être collectées à l'aide de diverses techniques et aident le chercheur à comprendre le phénomène, le fait ou le sujet qu'il étudie. La pertinence des données collectées et leur interprétation appropriée lors de l'analyse et de la conclusion fournissent des réponses au phénomène étudié. En somme, la collecte de données est un processus essentiel qui permet de prendre des décisions éclairées et d'améliorer les produits et services.

Pour la réalisation de cette étude, les données des éléments transposables ont été collectées à partir de plusieurs sources fiables et diversifiées. Tout d'abord, une partie des données provient de trois bases de données en ligne réputées : **UniProt [29]**, **DFAM [30]** et **NCBI [31]**, qui offrent une richesse d'informations sur les séquences génomiques et les éléments transposables. En complément de ces sources, un ensemble de données supplémentaires a été obtenu grâce à un partenariat étroit avec des biologistes, permettant ainsi d'enrichir et de diversifier notre base de données avec des informations spécifiques et adaptées aux besoins de notre recherche. Cette approche hybride, combinant des ressources numériques et des contributions directes d'experts, garantit la robustesse et la fiabilité des données utilisées dans cette étude.

Tableau 2 : les datasets utilisées : ETs

Source	Taille
UniProt et NCBI	13 334 séquences
DFAM	116 415 séquences
Données livrer	9 597 séquences
Totale	139 346 séquences

Les données des EnTs sont collecté à partir des sites **UniProt** et **NCBI**, en me basant initialement sur les séquences protéiques correspondantes. Ces séquences protéiques ont ensuite été converties en séquences d'ADN à l'aide d'un script Python que nous avons développé. Cette méthode m'a permis de générer des séquences d'ADN précises et pertinentes pour l'analyse des éléments transposables dans notre projet.

Tableau 3 : les datasets utilisées : EnTs

Source	Taille
NCBI	3 080 séquences
UniProt	601 792 séquences
Totale	604 872 séquences

Dans le cadre de cette étude, nous avons utilisé un ensemble de donnée distincts de type de fichier FASTA/TXT pour analyser les éléments transposables et non transposables dans le génome qui contient deux colonnes :

- **Sequence_ID** : Cette colonne contient les identifiants uniques des séquences génomiques. Ces identifiants indiquent souvent le type d'élément transposable et, dans certains cas, l'espèce d'origine ou un nom de catégorie.
- **Sequence** : Cette colonne affiche la séquence nucléotidique associée à chaque identifiant. Il s'agit de chaînes de lettres représentant les bases nucléotidiq (A, T, C, G) de l'ADN.

Des fichiers qui contient :

- Ensemble de données sur les éléments non transposables fichier : Cet ensemble de données contient des séquences ADN qui restent à une position fixe dans le génome. Il sert de base de référence pour comprendre la structure et la fonction normales du génome.

Caractérisation d'éléments transposables : les transposons à TIR

	Sequence_ID	Sequence
0	np_001104815.1 doublesex isoform m [bombyx mori]	atggtagcatgggcagctggaagaggagggtgcccacgtcg...
1	np_001036871.1 doublesex isoform f [bombyx mori]	atggtagcatgggcagctggaagaggagggtgcccacgtcg...
2	np_001037349.1 prothoracicotropic hormone prep...	atgtacccaggccatcatctggtgatcttgctacgcac...
3	np_001037683.1 gloverin 2 precursor [bombyx mori]	atgaacagcaaccttacatctcgccaccaccctggtgtgc...
4	np_001037488.1 fibroin light chain precursor [...]	atgaagccatcttcgtgtctgtgtggccaccagcgccatc...

Figure 29 : dataset EnT

- Ensemble de données sur les éléments transposables : Cet ensemble de données contient des informations sur les éléments génétiques qui peuvent se déplacer dans le génome qui sont présentée cette forme des séquences d'ADN et leurs séquences Id.

	Sequence_ID	Sequence
0	tc1-mariner_mite-44-llong_diptera_lutzomyia lo...	tgggtcatttatattacgcagaatttaacctcaaaatccat...
1	tc1-mariner_mite-189-aaegy_diptera_aedes aegypti	gggggactggggtaattggccacgttaaggaaaacgcgtttt...
2	piggybac_mite-25-bterr_hymenoptera_bombus terr...	ccgttgagtcacaaggcgcgtcgctgttagtttttgt...
3	p_mite-13-ldece_coleoptera_leptinotarsa_deceml...	cagtgcgtccgttaaggtaacttgcgcattcggtactattcc...
4	p_mite-141-apisu_hemiptera_acyrthosiphon pisum	catagatatacacaactagatagccgtcttcattgtcatgg...

Figure 30 : dataset ET

En travaillant avec ces ensembles de données, vous pouvez obtenir une vue d'ensemble complète de la dynamique des éléments transposables dans le génome et de leur impact sur la structure et la fonction génomiques.

Après la lecture des datasets on va maintenant les préparer pour le prétraitement :

L'image ci-dessous montre une table de données contenant des informations sur des séquences génomiques, avec plusieurs colonnes décrivant différents attributs des séquences.

- La première partie de la table contient des séquences non transposables (EnT) avec un label de 0.
- La seconde partie contient des séquences transposables (ET) avec un label de 1.
- Sequence Length : Cette colonne donne la longueur de chaque séquence en nombre de bases nucléotidiques, cette colonne est définie pour voir si la longueur du séquence satisfait les conditions définis (Annexe 1).

Caractérisation d'éléments transposables : les transposons à TIR

	Sequence_ID	Sequence	Label	Classname	Sequence Length
0	np_001104815.1 doublesex isoform m [bombyx mori]	atggtagcatggcgagcttgaagaggagggtgcggacgtcg...	0	EnT	798
1	np_001036871.1 doublesex isoform f [bombyx mori]	atggtagcatggcgagcttgaagaggagggtgcggacgtcg...	0	EnT	792
2	np_001037349.1 prothoracicotropic hormone prep...	atgatcaccaggccatcatcttgcgtatccgtgtacgc...	0	EnT	672
3	np_001037683.1 gloverin 2 precursor [bombyx mori]	atgaacagcaacctgttctacatcttcgcaccacccttgtgc...	0	EnT	519
4	np_001037488.1 fibroin light chain precursor [...]	atgaagccatcttcgtgtctgtggccaccaggcgccatcg...	0	EnT	786

	Sequence_ID	Sequence	Label	Classname	Sequence Length
0	tc1-mariner_mite-44-llong_diptera_lutzomyia lo...	ttgggtcatcttatatttacgcagaatttaacctaaaaatccat...	1	ET	101
1	tc1-mariner_mite-189-aaegy_diptera_aedes aegypti	gggggactggggtaatttccccacgttaaggaaaacagcgatttt...	1	ET	829
2	piggybac_mite-25-bterr_hymenoptera_bombus terr...	ccgtttgagtcccaagcgcgcgtcgctgttagattttgt...	1	ET	319
3	p_mite-13-ldece_coleoptera_leptinotarsa_deceml...	cagtgcgtccgttaaggtaacttattgtaccgcattcggtactattcc...	1	ET	791
4	p_mite-141-apisu_hemiptera_acyrthosiphon pisum	catagatataacaactagatagccgtctccgtcatgg...	1	ET	245

Figure 31 : dataset classification binaire : les ET/EnT

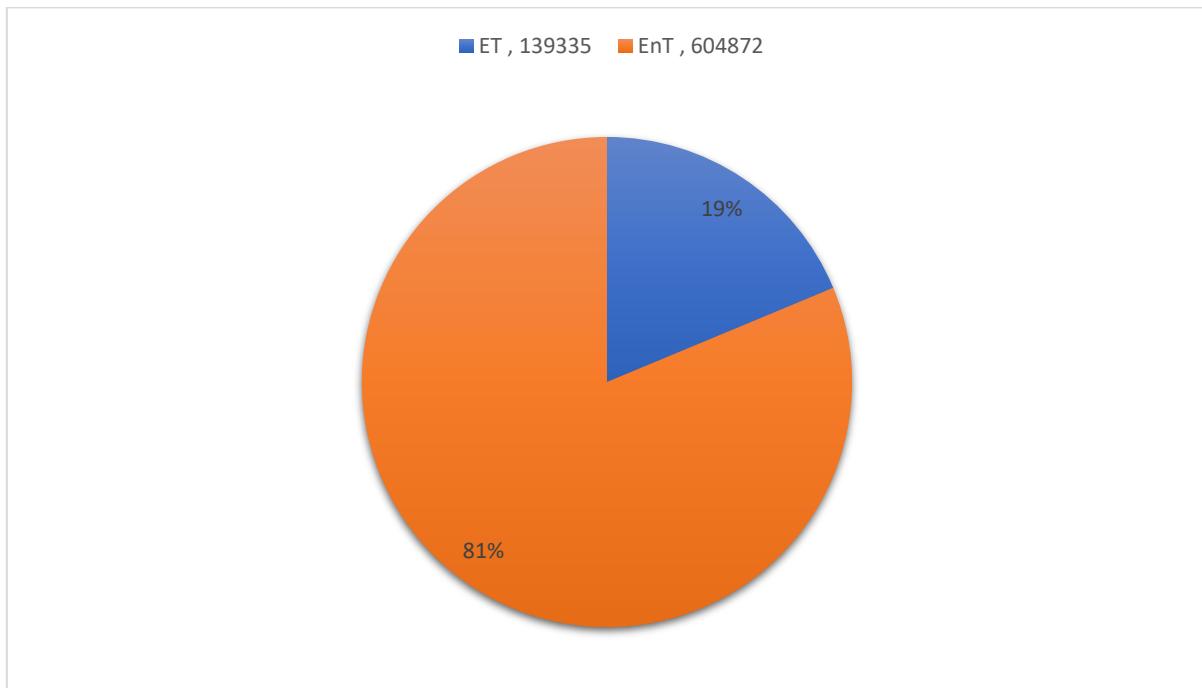


Figure 32 : pourcentage des ETs et EnTs dans l'ensemble de données

D'après l'ensemble de données des ETs (figure 30) on va les étiqueter on 7 classes

L'image montre une table de données qui contient des informations spécifiques sur des séquences génomiques appartenant à des éléments transposables (ET) pour la classification multiclasse.

- Category : Cette colonne identifie la superfamille ou le type spécifique d'élément transposable auquel chaque séquence appartient. Les catégories visibles incluent "Merlin", "HAT", "MuDR", "PiggyBac", "P_Element", "CACTA" et "TC1_Mariner".

Caractérisation d'éléments transposables : les transposons à TIR

	Sequence_ID	Sequence	Classname	Sequence Length	Category	Label
0	kw=dna/merlin.	attattcttgntaacgtcagttaaaaatacaactgttcatgtta...	ET	503	Merlin	6
1	kw=dna/hat-ac.	aaaaaaattaataaaaataacataaggcagagggaggaatttat...	ET	6911	HAT	4
2	MuDR-18_SBi\tMuDR\tSorghum bicolor\n	cgcgaatctccggccgaaacctcgccggacggggccgtccgtccgt...	ET	29154	MuDR	3
3	piggyBac-N2B_DR\tpiggyBac\tDanio rerio\n	cccttaactgccacaccaagaaaaagcaatagaaaaaaatttgc...	ET	1094	PiggyBac	5
4	kw=dna/piggybac.	tagcatgtcgcccnaaaacgtcatagtagatagcatgtcgctcaa...	ET	19125	P_Element	2

Figure 33 : dataset classification multiclasses

Voici la quantité de chaque superfamille dans le totale des données :

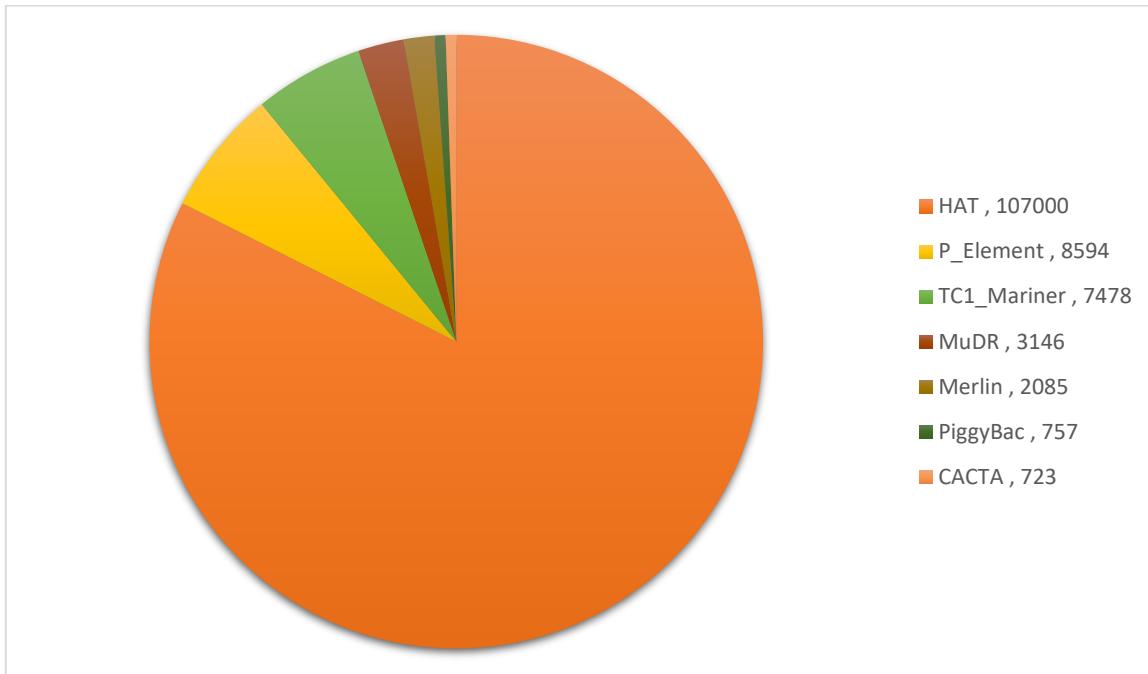


Figure 34 : quantité de chaque superfamille (avant le pré-traitement)

4. Prétraitement des données

C'est le processus de nettoyage et de transformation des données brutes avant le traitement et l'analyse. Il s'agit d'une étape importante avant le traitement et implique souvent le reformatage des données, la correction des données et la combinaison d'ensembles de données pour enrichir les données.

Le pré-traitement de nos données consiste à éliminer les descriptions des scaffolds (les séquences Id), transformer tous les nucléotides en minuscule, supprimer les séquences dupliquées puis filtrage des données, pour chaque catégorie, les données qui satisfont les conditions définies (**Annexe 1**), le processus élimine automatiquement les séquences qui ne répondent pas aux critères de longueur minimale définis pour chaque catégorie d'éléments transposables. Cela signifie que pour chaque catégorie, seules les séquences ayant une longueur égale ou supérieure au seuil spécifié (par exemple, 1300 nucléotides pour **TC1 Mariner** ou

2900 nucléotides pour **P Element**) sont conservées. Les séquences plus courtes que ces seuils sont automatiquement exclues du DataFrame final, garantissant ainsi que seules les séquences pertinentes et suffisantes en termes de longueur sont retenues pour l'analyse.

Enfin de vérifier si l'une des séquences comprend un nucléotide qui diffère de (A, C, G, T) et de le mettrez comme (z).

5. Tokenisation des séquences

La tokenisation des séquences avec des k-mers est une technique couramment utilisée en bio-informatique pour l'analyse des séquences génomiques. Un k-mer est une sous-chaîne de longueur k d'une chaîne de caractères plus grande, dans ce cas, une séquence d'ADN. La tokenisation avec des k-mers implique la division de la séquence d'ADN en sous-séquences de longueur k. Ces k-mers sont ensuite utilisés pour diverses applications, comme l'assemblage et l'alignement de séquences. Par exemple, lors de l'assemblage de séquences, le fractionnement en k-mers aide à résoudre le problème des différentes longueurs de lectures initiales. Ainsi, la tokenisation des séquences avec des k-mers est un outil puissant pour l'analyse des séquences génomiques.

Considérons la séquence d'ADN "ACGAGGTACGA" qui se compose de 11 nucléotides. Essayons d'obtenir tous les 4-mers (sous-chaînes de longueur 4) dans cette séquence d'ADN.

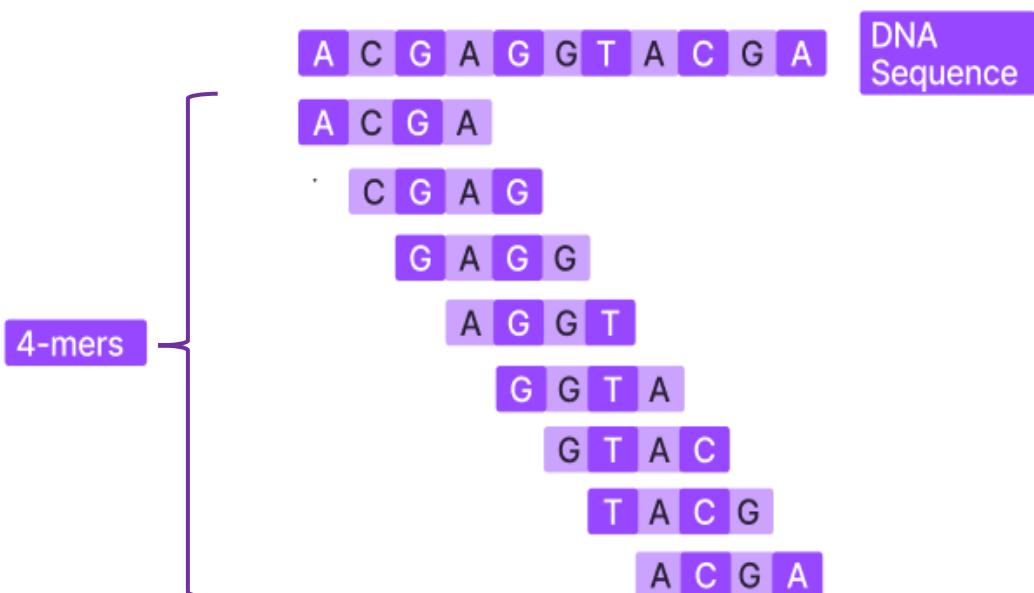


Figure 35 : exemple de 4-mers

Dans notre cas, nous avons essayé de tokeniser les séquences avec k allant de 3 à 6 pour choisir le bon k pour la classification binaire et multi-classe.

6. Vectorisation des séquences

Nous avons utilisé trois méthodes de vectorisation pour représenter les séquences d'ADN sous forme numérique : TF-IDF Vectorizer, CountVectorizer et One Hot Encoding :

6.1. TF-IDF

La vectorisation avec TF-IDF (Term Frequency-Inverse Document Frequency) est une technique couramment utilisée pour transformer des séquences textuelles en vecteurs numériques. Après avoir divisé les séquences en k-mers, chaque k-mer est traité comme un “mot” ou un “token”. Le TF-IDF est ensuite utilisé pour quantifier l’importance de chaque k-mer dans la séquence. Le score TF-IDF pour un k-mer donné est le produit de sa fréquence dans la séquence (TF) et l’inverse de sa fréquence dans l’ensemble des séquences (IDF). Cette méthode permet de réduire l’importance des k-mers communs et d’augmenter l’importance des k-mers rares, ce qui peut être utile pour identifier des caractéristiques uniques dans les séquences.

6.2. CountVectorizer

La vectorisation avec CountVectorizer pour les séquences d'ADN est une technique qui permet de transformer des séquences d'ADN en vecteurs numériques exploitables par des algorithmes de ML/DL. Cette méthode repose sur la conversion des k-mers présents dans les séquences d'ADN en vecteurs de comptage, où chaque dimension du vecteur représente un k-mer spécifique, et la valeur correspondante indique le nombre de fois que ce k-mer apparaît dans la séquence.

6.3. One Hot Encoding

L'encodage one-hot est une méthode simple et efficace pour transformer des données catégorielles en un format numérique compréhensible par les modèles d'apprentissage automatique. Lorsqu'on travaille avec des séquences ADN, qui sont composées de quatre nucléotides (A, C, G, T), l'encodage one-hot est couramment utilisé pour représenter chaque nucléotide comme un vecteur binaire :

Voici comment cela se fait :

- **A** : [1, 0, 0, 0]
- **T** : [0, 1, 0, 0]

- **C** : [0, 0, 1, 0]
- **G** : [0, 0, 0, 1]

7. Ensemble des données après le prétraitement

Après le prétraitement des données on visualise les résultats suivants :

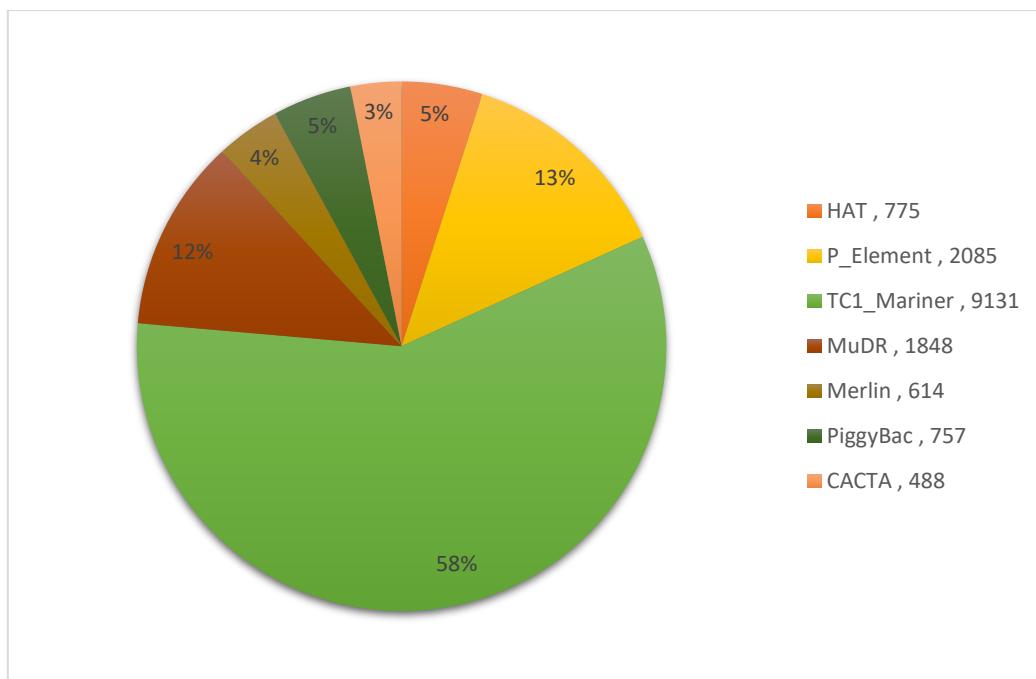


Figure 36 : quantité de chaque superfamille (après le pré-traitement)

8. Knowledge discovery : choix des modèles ML/DL

Dans cette partie on va voir les résultats des modèles ML/DL utilisés pour choisir la bonne méthode de vectorisation et le meilleur k pour k-mer.

a. K-means

Pour évaluer le modèle de K-means nous avons utilisé la méthode Elbow qui est une méthode heuristique utilisée pour déterminer le nombre de clusters dans le clustering kmeans. C'est ce qu'on appelle la méthode du "Elbow" car le nombre de clusters est choisi au point d'inflexion (où le "Elbow" se produit) dans le tracé de la somme intra-cluster des distances au carré par rapport au nombre de clusters.

La figure suivante montre que le nombre optimal de cluster dans notre cas est deux (k=2)

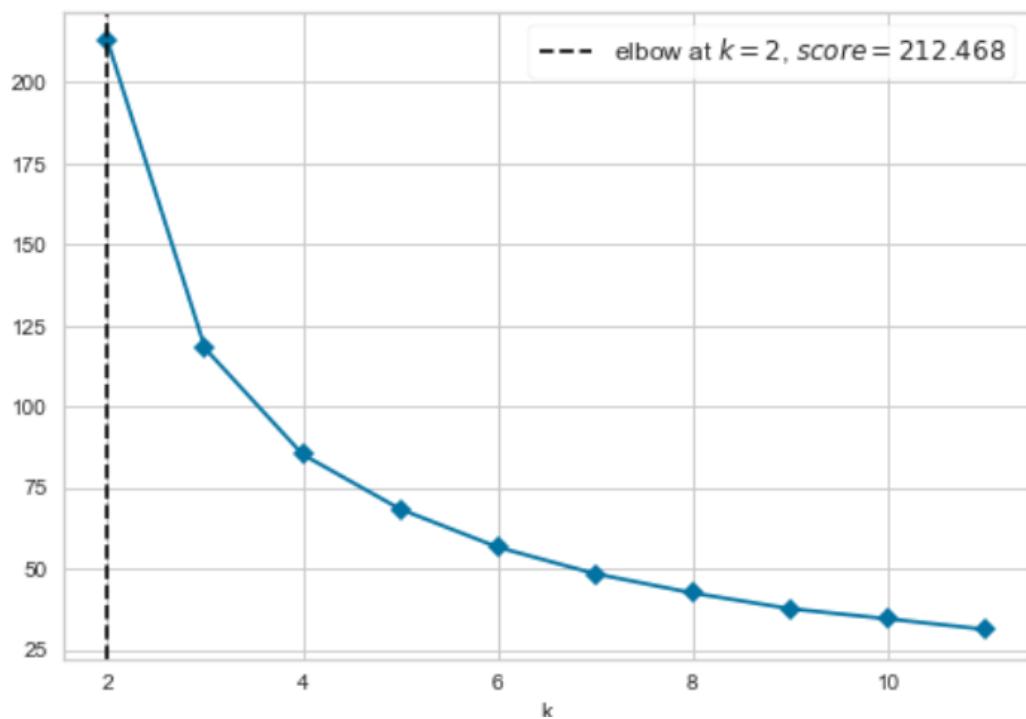


Figure 37 : La méthode Elbow

Après qu'on a trouvé le nombre de cluster on lance l'entraînement de modèle puis on récupère les centroïdes de chaque cluster et on les affiche par le graphe suivant.

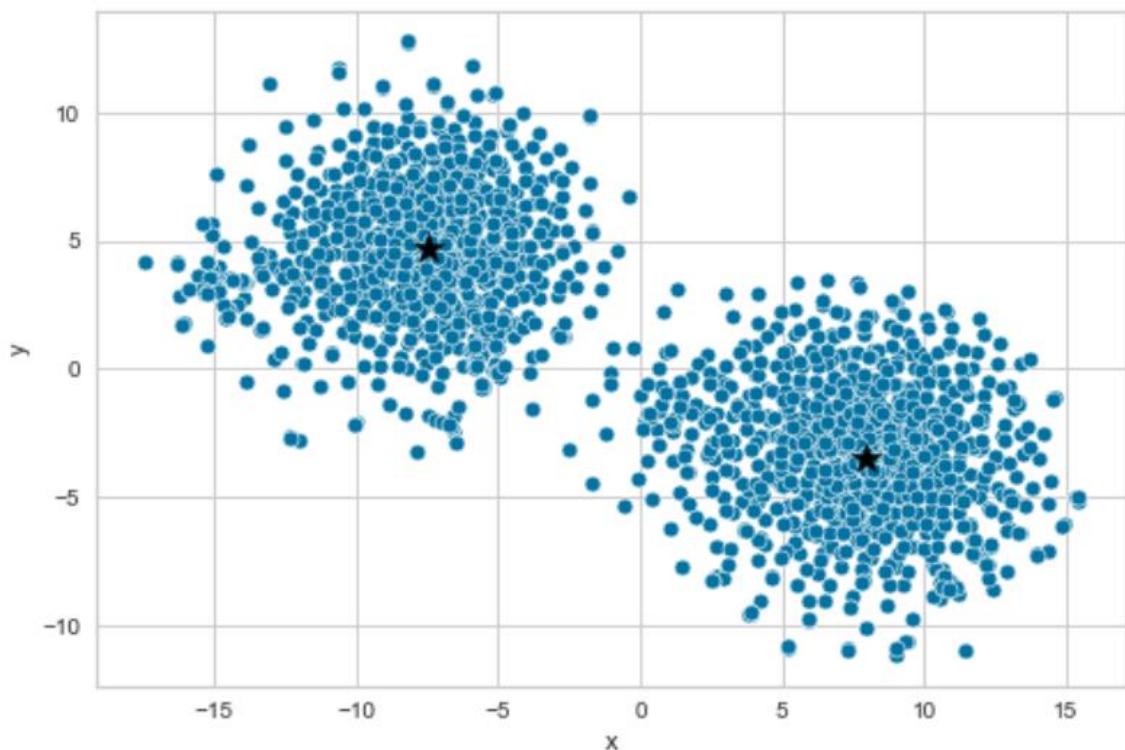


Figure 38 : Les centroïdes de chaque cluster

Ce modèle est juste pour avoir si les données peuvent être séparées linéairement en deux groupes distincts et c'est le cas pour la classification binaire.

b. Classification binaire

b.1. ML

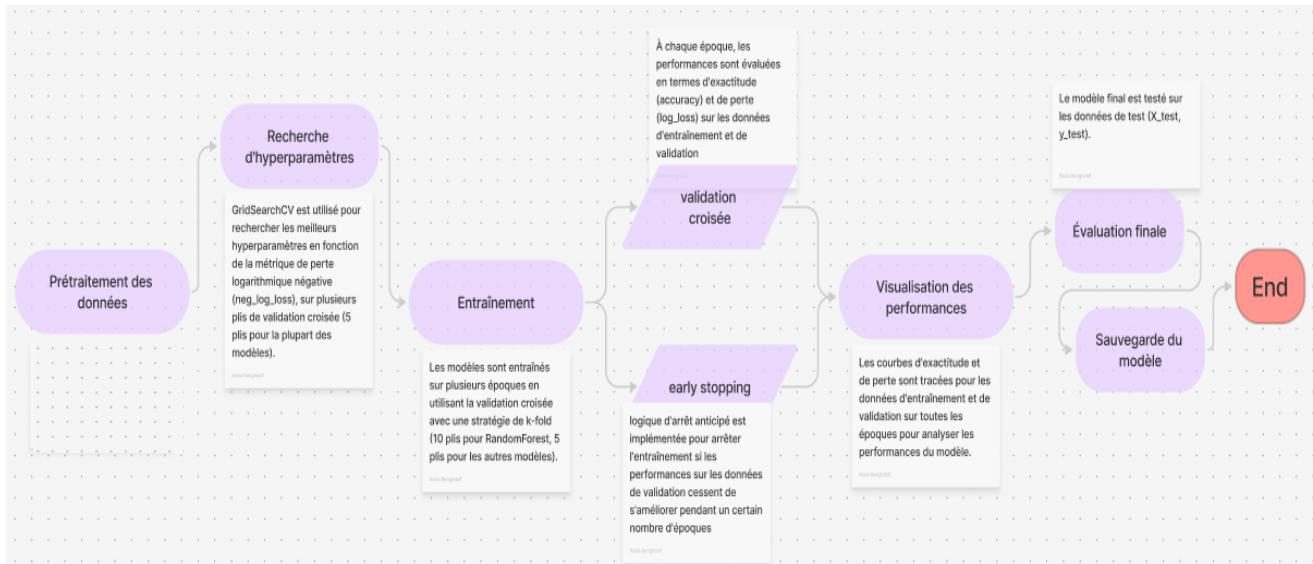


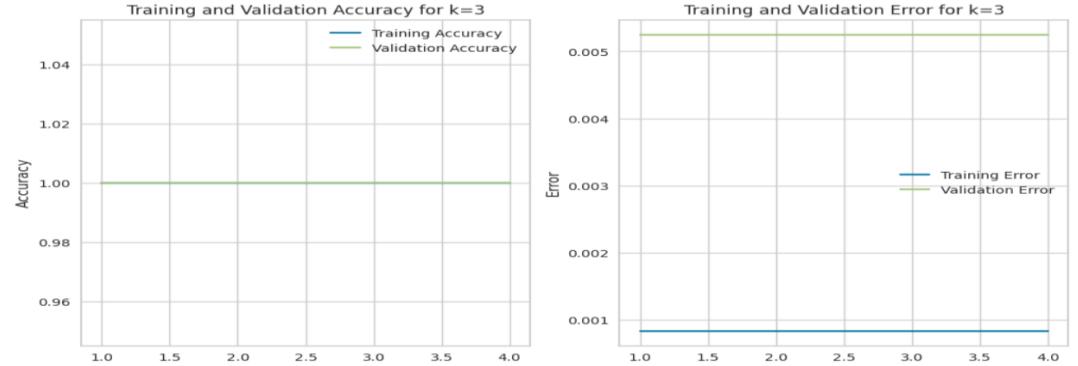
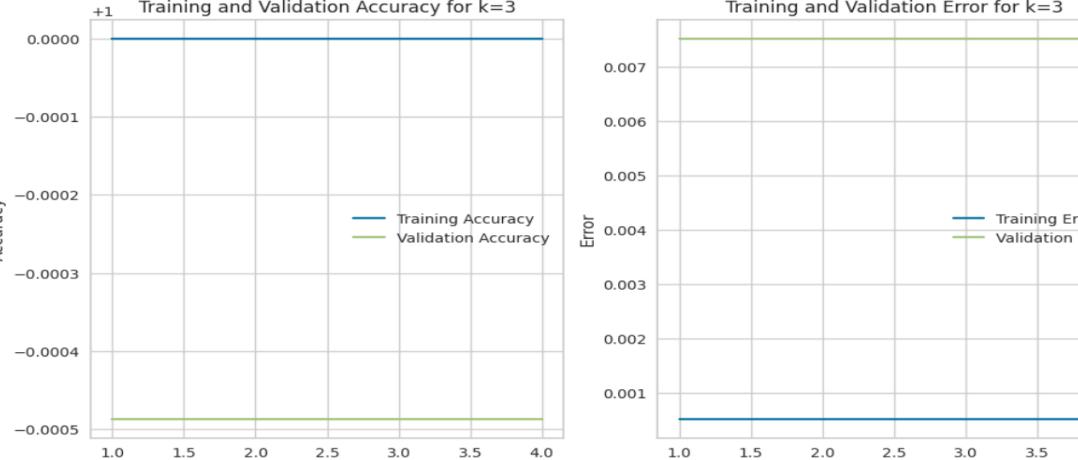
Figure 39 : logigramme pour la CB : partie ML

Ce logigramme est bien structuré pour évaluer et optimiser les modèles de classification ML pour la tâche de classification binaire à l'aide de séquences de k-mers, avec des techniques avancées de validation et de visualisation des performances.

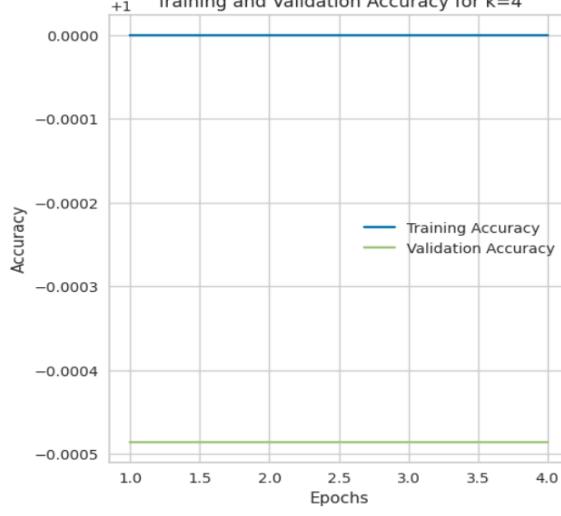
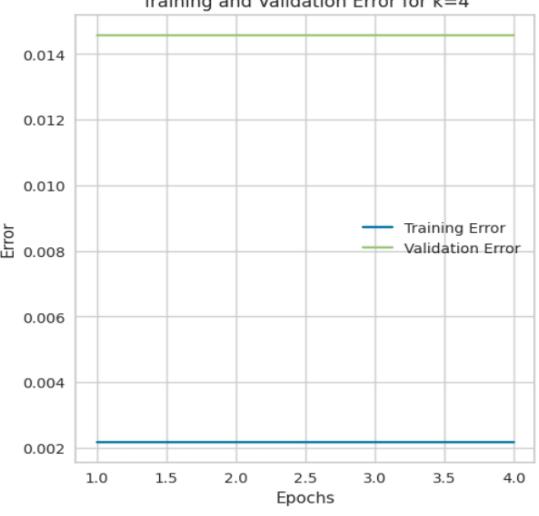
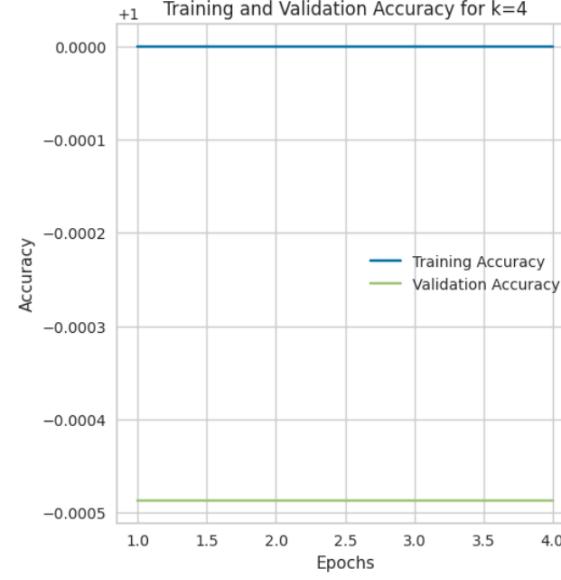
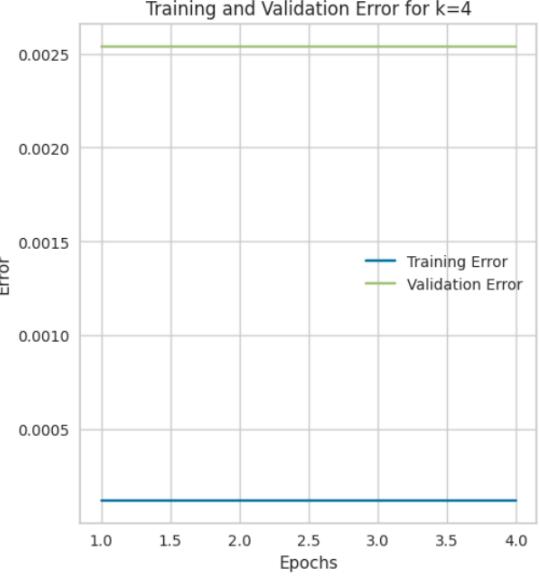
Dans la partie de prétraitement des données, nous avons utilisé la matrice de corrélation pour identifier les paires de caractéristiques fortement corrélées dans notre ensemble de données d'entraînement. En définissant un seuil de corrélation de 0,8, nous avons pu repérer les paires de caractéristiques dont la corrélation absolue était supérieure à ce seuil. Ensuite, pour chaque paire fortement corrélée, nous avons éliminé l'une des deux caractéristiques, en conservant celle avec l'indice le plus faible ou en utilisant d'autres critères basés sur mes connaissances du domaine. Finalement, nous avons filtré notre ensemble de données d'entraînement pour ne conserver que les caractéristiques sélectionnées, afin de réduire la redondance dans les données et d'améliorer l'efficacité du modèle.

b.1.1. Random Forest

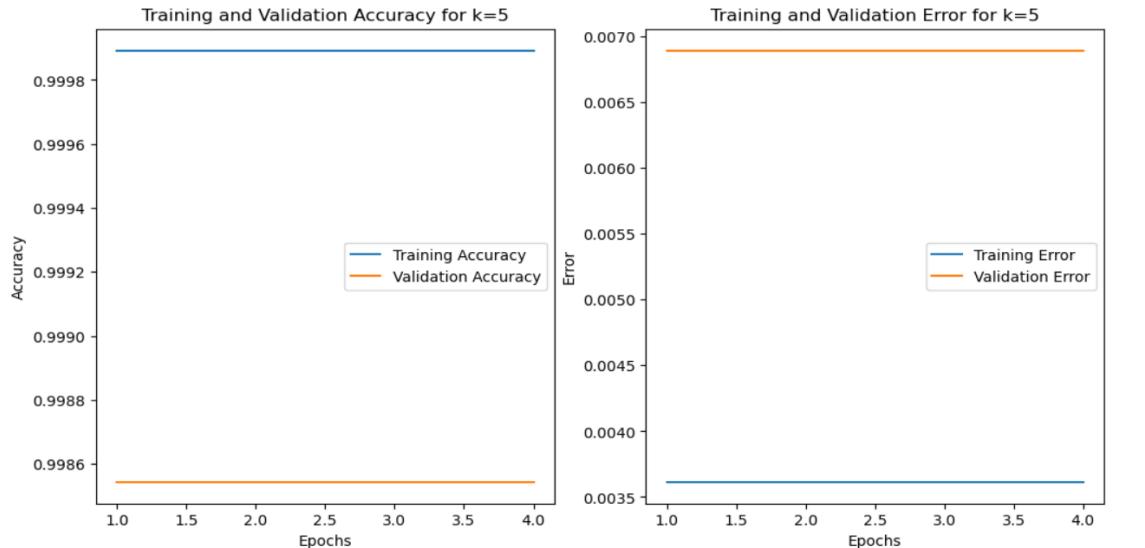
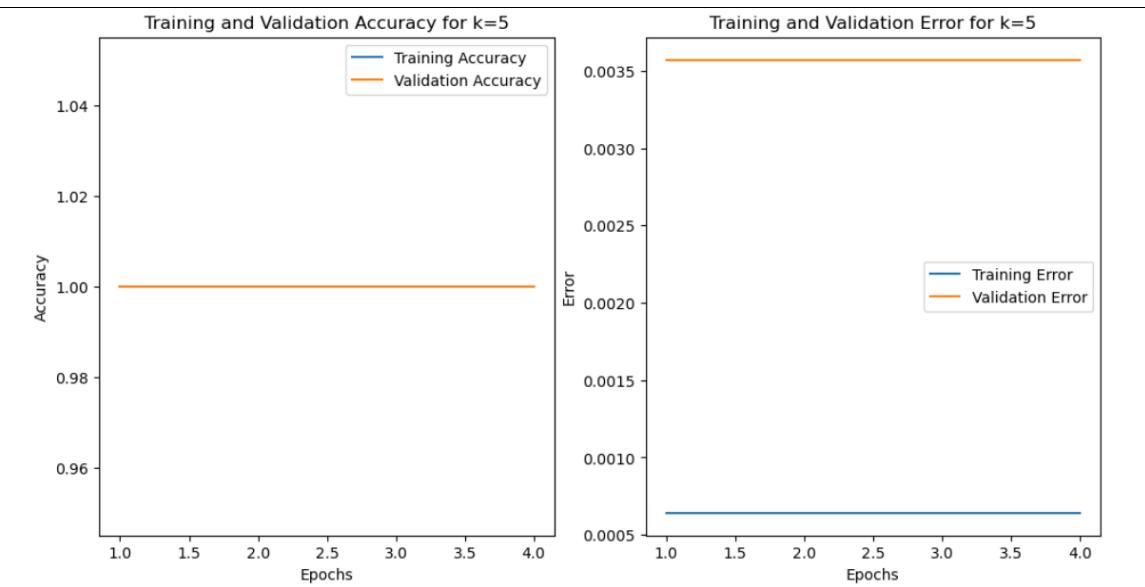
Tableau 4 : résultat CB/ML : modèle Random Forest

Feature Extraction	Grid Search	Precision	Recall	F1-Score	plot
CV k=3	Best parameters found: {'bootstrap': False, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 4, 'n_estimators': 300}	100%	100%	100%	
TFiDF k=3	Best parameters found: {'bootstrap': False, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 4, 'n_estimators': 300}	100%	100%	100%	

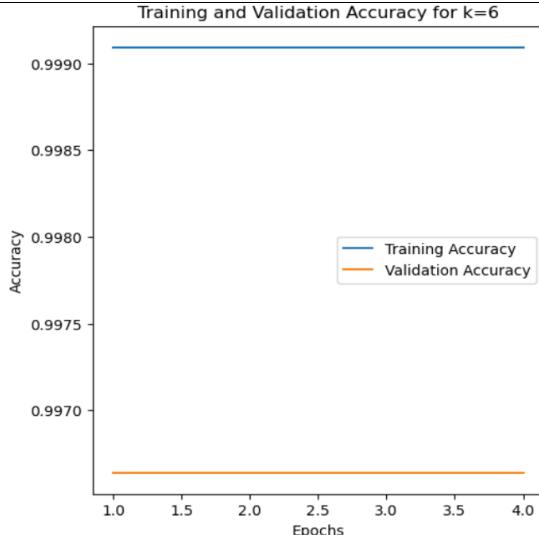
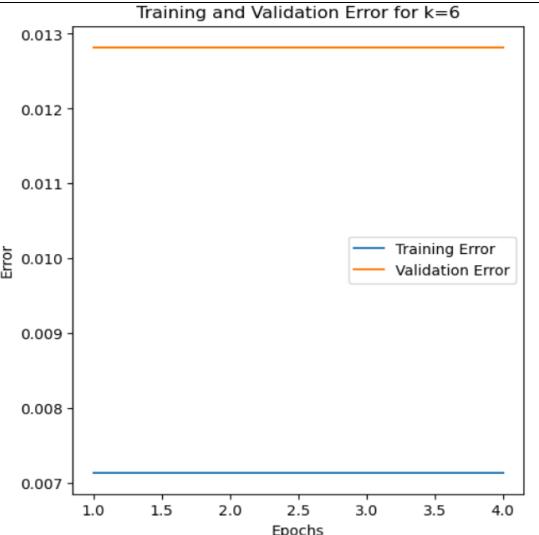
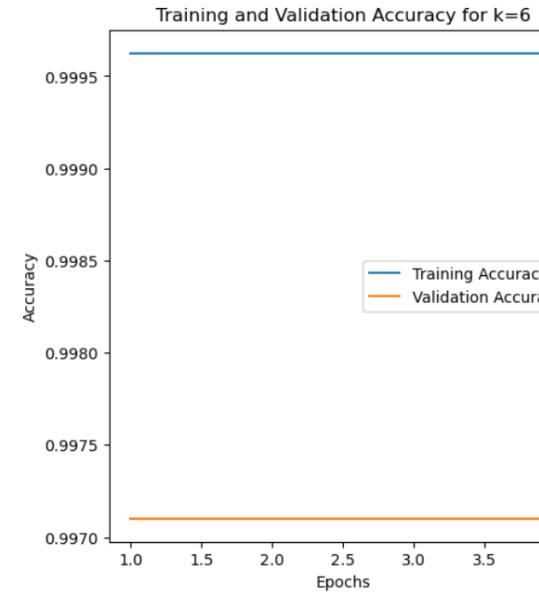
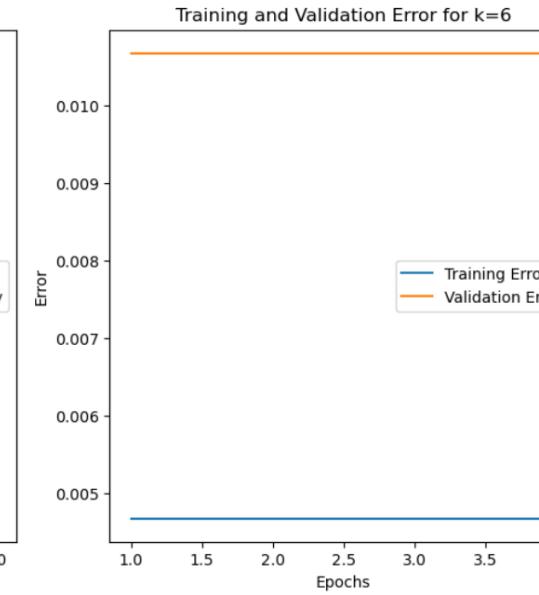
Caractérisation d'éléments transposables : les transposons à TIR

CV k=4	Best parameters found: { 'bootstrap': False, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 4, 'n_estimators': 100}	100%	100%	100%	 <p>Training and Validation Accuracy for k=4</p> <table border="1"> <thead> <tr> <th>Epochs</th> <th>Training Accuracy</th> <th>Validation Accuracy</th> </tr> </thead> <tbody> <tr><td>1.0</td><td>0.0000</td><td>-0.0005</td></tr> <tr><td>1.5</td><td>0.0000</td><td>-0.0005</td></tr> <tr><td>2.0</td><td>0.0000</td><td>-0.0005</td></tr> <tr><td>2.5</td><td>0.0000</td><td>-0.0005</td></tr> <tr><td>3.0</td><td>0.0000</td><td>-0.0005</td></tr> <tr><td>3.5</td><td>0.0000</td><td>-0.0005</td></tr> <tr><td>4.0</td><td>0.0000</td><td>-0.0005</td></tr> </tbody> </table>	Epochs	Training Accuracy	Validation Accuracy	1.0	0.0000	-0.0005	1.5	0.0000	-0.0005	2.0	0.0000	-0.0005	2.5	0.0000	-0.0005	3.0	0.0000	-0.0005	3.5	0.0000	-0.0005	4.0	0.0000	-0.0005	 <p>Training and Validation Error for k=4</p> <table border="1"> <thead> <tr> <th>Epochs</th> <th>Training Error</th> <th>Validation Error</th> </tr> </thead> <tbody> <tr><td>1.0</td><td>0.002</td><td>0.015</td></tr> <tr><td>1.5</td><td>0.002</td><td>0.015</td></tr> <tr><td>2.0</td><td>0.002</td><td>0.015</td></tr> <tr><td>2.5</td><td>0.002</td><td>0.015</td></tr> <tr><td>3.0</td><td>0.002</td><td>0.015</td></tr> <tr><td>3.5</td><td>0.002</td><td>0.015</td></tr> <tr><td>4.0</td><td>0.002</td><td>0.015</td></tr> </tbody> </table>	Epochs	Training Error	Validation Error	1.0	0.002	0.015	1.5	0.002	0.015	2.0	0.002	0.015	2.5	0.002	0.015	3.0	0.002	0.015	3.5	0.002	0.015	4.0	0.002	0.015
Epochs	Training Accuracy	Validation Accuracy																																																				
1.0	0.0000	-0.0005																																																				
1.5	0.0000	-0.0005																																																				
2.0	0.0000	-0.0005																																																				
2.5	0.0000	-0.0005																																																				
3.0	0.0000	-0.0005																																																				
3.5	0.0000	-0.0005																																																				
4.0	0.0000	-0.0005																																																				
Epochs	Training Error	Validation Error																																																				
1.0	0.002	0.015																																																				
1.5	0.002	0.015																																																				
2.0	0.002	0.015																																																				
2.5	0.002	0.015																																																				
3.0	0.002	0.015																																																				
3.5	0.002	0.015																																																				
4.0	0.002	0.015																																																				
TFIDF k=4	Best parameters found: { 'bootstrap': False, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 4, 'n_estimators': 100}	100%	100%	100%	 <p>Training and Validation Accuracy for k=4</p> <table border="1"> <thead> <tr> <th>Epochs</th> <th>Training Accuracy</th> <th>Validation Accuracy</th> </tr> </thead> <tbody> <tr><td>1.0</td><td>0.0000</td><td>-0.0005</td></tr> <tr><td>1.5</td><td>0.0000</td><td>-0.0005</td></tr> <tr><td>2.0</td><td>0.0000</td><td>-0.0005</td></tr> <tr><td>2.5</td><td>0.0000</td><td>-0.0005</td></tr> <tr><td>3.0</td><td>0.0000</td><td>-0.0005</td></tr> <tr><td>3.5</td><td>0.0000</td><td>-0.0005</td></tr> <tr><td>4.0</td><td>0.0000</td><td>-0.0005</td></tr> </tbody> </table>	Epochs	Training Accuracy	Validation Accuracy	1.0	0.0000	-0.0005	1.5	0.0000	-0.0005	2.0	0.0000	-0.0005	2.5	0.0000	-0.0005	3.0	0.0000	-0.0005	3.5	0.0000	-0.0005	4.0	0.0000	-0.0005	 <p>Training and Validation Error for k=4</p> <table border="1"> <thead> <tr> <th>Epochs</th> <th>Training Error</th> <th>Validation Error</th> </tr> </thead> <tbody> <tr><td>1.0</td><td>0.0025</td><td>0.0025</td></tr> <tr><td>1.5</td><td>0.0025</td><td>0.0025</td></tr> <tr><td>2.0</td><td>0.0025</td><td>0.0025</td></tr> <tr><td>2.5</td><td>0.0025</td><td>0.0025</td></tr> <tr><td>3.0</td><td>0.0025</td><td>0.0025</td></tr> <tr><td>3.5</td><td>0.0025</td><td>0.0025</td></tr> <tr><td>4.0</td><td>0.0025</td><td>0.0025</td></tr> </tbody> </table>	Epochs	Training Error	Validation Error	1.0	0.0025	0.0025	1.5	0.0025	0.0025	2.0	0.0025	0.0025	2.5	0.0025	0.0025	3.0	0.0025	0.0025	3.5	0.0025	0.0025	4.0	0.0025	0.0025
Epochs	Training Accuracy	Validation Accuracy																																																				
1.0	0.0000	-0.0005																																																				
1.5	0.0000	-0.0005																																																				
2.0	0.0000	-0.0005																																																				
2.5	0.0000	-0.0005																																																				
3.0	0.0000	-0.0005																																																				
3.5	0.0000	-0.0005																																																				
4.0	0.0000	-0.0005																																																				
Epochs	Training Error	Validation Error																																																				
1.0	0.0025	0.0025																																																				
1.5	0.0025	0.0025																																																				
2.0	0.0025	0.0025																																																				
2.5	0.0025	0.0025																																																				
3.0	0.0025	0.0025																																																				
3.5	0.0025	0.0025																																																				
4.0	0.0025	0.0025																																																				

Caractérisation d'éléments transposables : les transposons à TIR

CV k=5	Best parameters found: <pre>{'bootstrap': False, 'max_depth': 10, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 6, 'n_estimators': 100}</pre>	100%	100%	100%	
TFIDF k=5	Best parameters found: <pre>{'bootstrap': False, 'max_depth': 10, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 6, 'n_estimators': 300}</pre>	100%	100%	100%	

Caractérisation d'éléments transposables : les transposons à TIR

CV k=6	Best parameters found: {'bootstrap': False, 'max_depth': 10, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 6, 'n_estimators': 100}	99,5%	99,5%	100%	 Training and Validation Accuracy for k=6 Accuracy Epochs	 Training and Validation Error for k=6 Error Epochs
TFIDF k=6	Best parameters found: {'bootstrap': False, 'max_depth': 10, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 8, 'n_estimators': 500}	100%	100%	100%	 Training and Validation Accuracy for k=6 Accuracy Epochs	 Training and Validation Error for k=6 Error Epochs

b.1.2. Extra Trees

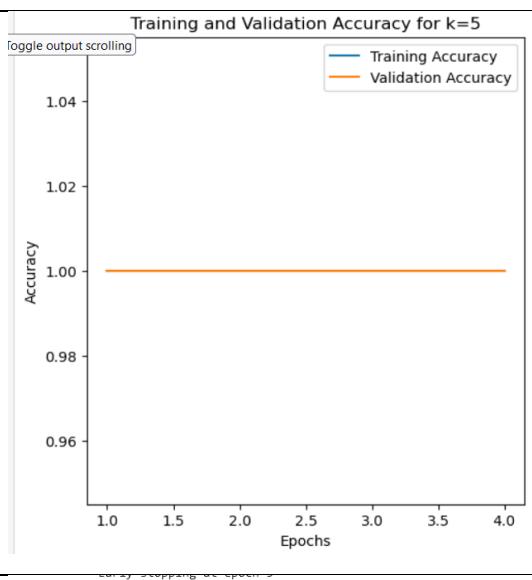
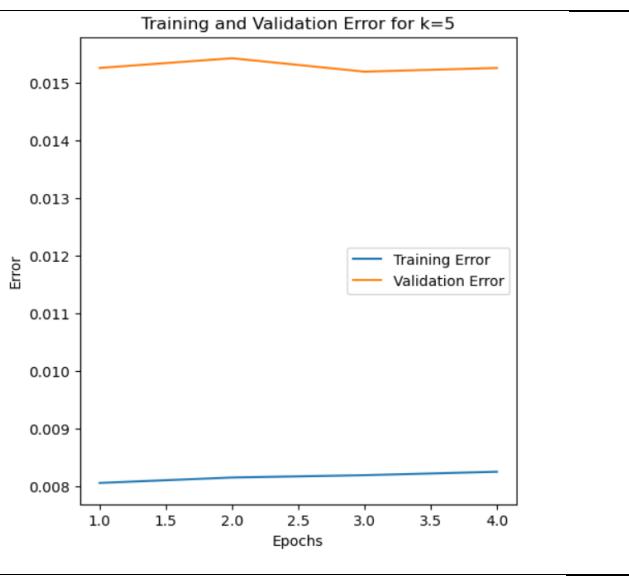
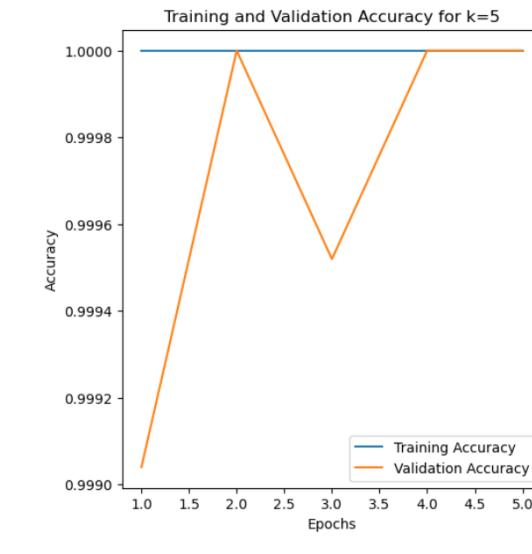
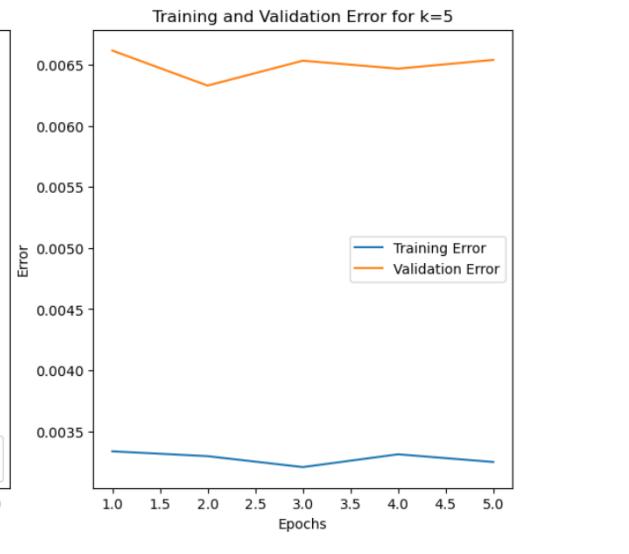
Tableau 5 : résultat CB/ML : modèle Extra Trees

Feature Extraction	Grid Search	Precision	Recall	F1-Score	plot
CV k=3	Best parameters found: { 'bootstrap': True, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 4, 'n_estimators': 100}	98%	98%	98%	
TFiDF k=3	Best parameters found: { 'bootstrap': False, 'max_features': 'auto', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100}	100%	100%	100%	

Caractérisation d'éléments transposables : les transposons à TIR

CV k=4	Best parameters found: {'bootstrap': True, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 8, 'n_estimators': 200}	100%	100%	100%	<p>Training and Validation Accuracy for k=4</p> <table border="1"> <thead> <tr> <th>Epochs</th> <th>Training Accuracy</th> <th>Validation Accuracy</th> </tr> </thead> <tbody> <tr><td>1</td><td>0.9993</td><td>0.9985</td></tr> <tr><td>2</td><td>0.9993</td><td>0.9989</td></tr> <tr><td>3</td><td>0.9993</td><td>0.9994</td></tr> <tr><td>4</td><td>0.9991</td><td>0.9982</td></tr> <tr><td>5</td><td>0.9993</td><td>0.9986</td></tr> <tr><td>6</td><td>0.9995</td><td>0.9990</td></tr> </tbody> </table>	Epochs	Training Accuracy	Validation Accuracy	1	0.9993	0.9985	2	0.9993	0.9989	3	0.9993	0.9994	4	0.9991	0.9982	5	0.9993	0.9986	6	0.9995	0.9990	<p>Training and Validation Error for k=4</p> <table border="1"> <thead> <tr> <th>Epochs</th> <th>Training Error</th> <th>Validation Error</th> </tr> </thead> <tbody> <tr><td>1</td><td>0.094</td><td>0.104</td></tr> <tr><td>2</td><td>0.093</td><td>0.105</td></tr> <tr><td>3</td><td>0.093</td><td>0.103</td></tr> <tr><td>4</td><td>0.094</td><td>0.103</td></tr> <tr><td>5</td><td>0.092</td><td>0.102</td></tr> <tr><td>6</td><td>0.092</td><td>0.103</td></tr> </tbody> </table>	Epochs	Training Error	Validation Error	1	0.094	0.104	2	0.093	0.105	3	0.093	0.103	4	0.094	0.103	5	0.092	0.102	6	0.092	0.103						
Epochs	Training Accuracy	Validation Accuracy																																																				
1	0.9993	0.9985																																																				
2	0.9993	0.9989																																																				
3	0.9993	0.9994																																																				
4	0.9991	0.9982																																																				
5	0.9993	0.9986																																																				
6	0.9995	0.9990																																																				
Epochs	Training Error	Validation Error																																																				
1	0.094	0.104																																																				
2	0.093	0.105																																																				
3	0.093	0.103																																																				
4	0.094	0.103																																																				
5	0.092	0.102																																																				
6	0.092	0.103																																																				
TFIDF k=4	Best parameters found: {'bootstrap': True, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 4, 'n_estimators': 100}	100%	100%	100%	<p>Training and Validation Accuracy for k=4</p> <table border="1"> <thead> <tr> <th>Epochs</th> <th>Training Accuracy</th> <th>Validation Accuracy</th> </tr> </thead> <tbody> <tr><td>1</td><td>1.000</td><td>0.9991</td></tr> <tr><td>2</td><td>1.000</td><td>0.9995</td></tr> <tr><td>3</td><td>1.000</td><td>0.9995</td></tr> <tr><td>4</td><td>1.000</td><td>1.000</td></tr> <tr><td>5</td><td>1.000</td><td>1.000</td></tr> <tr><td>6</td><td>1.000</td><td>0.9991</td></tr> <tr><td>7</td><td>1.000</td><td>1.000</td></tr> </tbody> </table>	Epochs	Training Accuracy	Validation Accuracy	1	1.000	0.9991	2	1.000	0.9995	3	1.000	0.9995	4	1.000	1.000	5	1.000	1.000	6	1.000	0.9991	7	1.000	1.000	<p>Training and Validation Error for k=4</p> <table border="1"> <thead> <tr> <th>Epochs</th> <th>Training Error</th> <th>Validation Error</th> </tr> </thead> <tbody> <tr><td>1</td><td>0.0068</td><td>0.0118</td></tr> <tr><td>2</td><td>0.0068</td><td>0.0116</td></tr> <tr><td>3</td><td>0.0066</td><td>0.0112</td></tr> <tr><td>4</td><td>0.0065</td><td>0.0118</td></tr> <tr><td>5</td><td>0.0065</td><td>0.0114</td></tr> <tr><td>6</td><td>0.0064</td><td>0.0114</td></tr> <tr><td>7</td><td>0.0062</td><td>0.0108</td></tr> </tbody> </table>	Epochs	Training Error	Validation Error	1	0.0068	0.0118	2	0.0068	0.0116	3	0.0066	0.0112	4	0.0065	0.0118	5	0.0065	0.0114	6	0.0064	0.0114	7	0.0062	0.0108
Epochs	Training Accuracy	Validation Accuracy																																																				
1	1.000	0.9991																																																				
2	1.000	0.9995																																																				
3	1.000	0.9995																																																				
4	1.000	1.000																																																				
5	1.000	1.000																																																				
6	1.000	0.9991																																																				
7	1.000	1.000																																																				
Epochs	Training Error	Validation Error																																																				
1	0.0068	0.0118																																																				
2	0.0068	0.0116																																																				
3	0.0066	0.0112																																																				
4	0.0065	0.0118																																																				
5	0.0065	0.0114																																																				
6	0.0064	0.0114																																																				
7	0.0062	0.0108																																																				

Caractérisation d'éléments transposables : les transposons à TIR

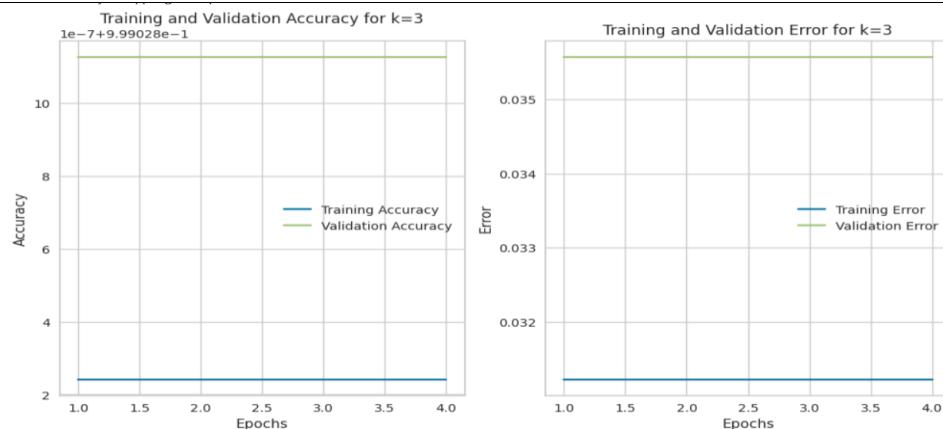
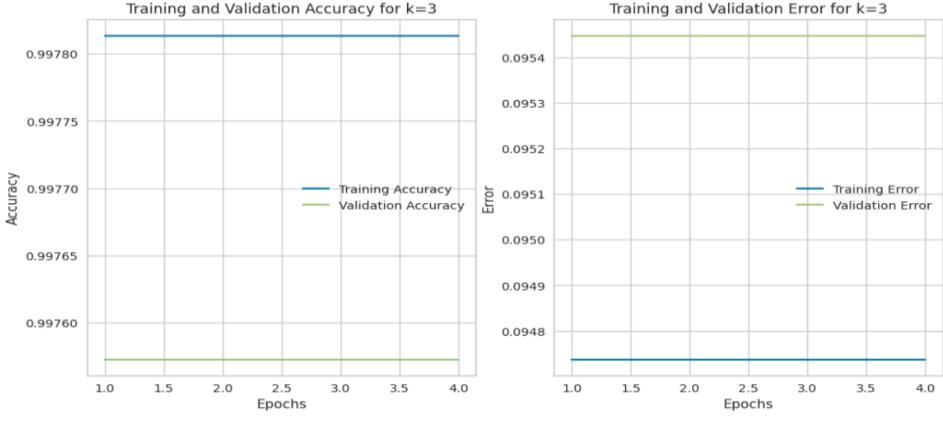
CV k=5	Best parameters found: { 'bootstrap': True, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 4, 'n_estimators': 200}	100%	100%	100%	 
TFIDF k=5	Best parameters found: { 'bootstrap': True, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 4, 'n_estimators': 200}	99,5%	99,5%	100%	 

Caractérisation d'éléments transposables : les transposons à TIR

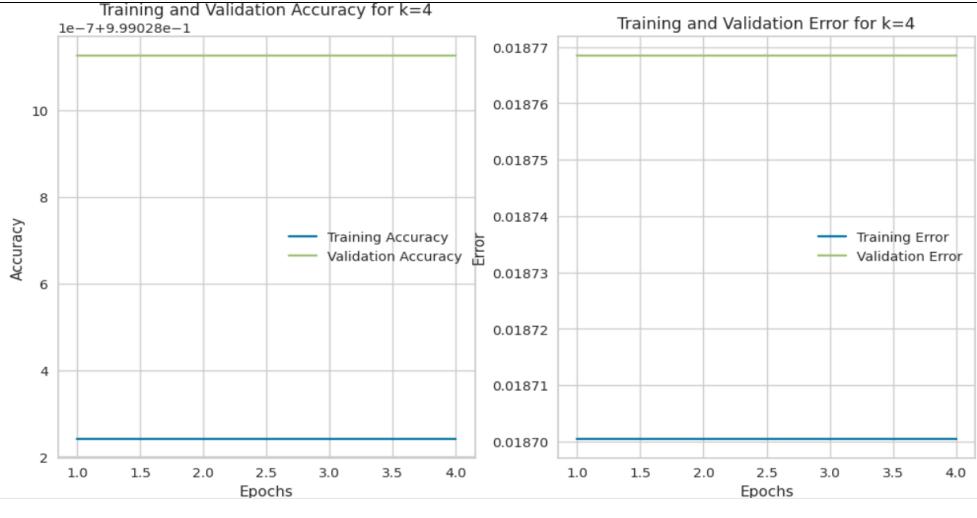
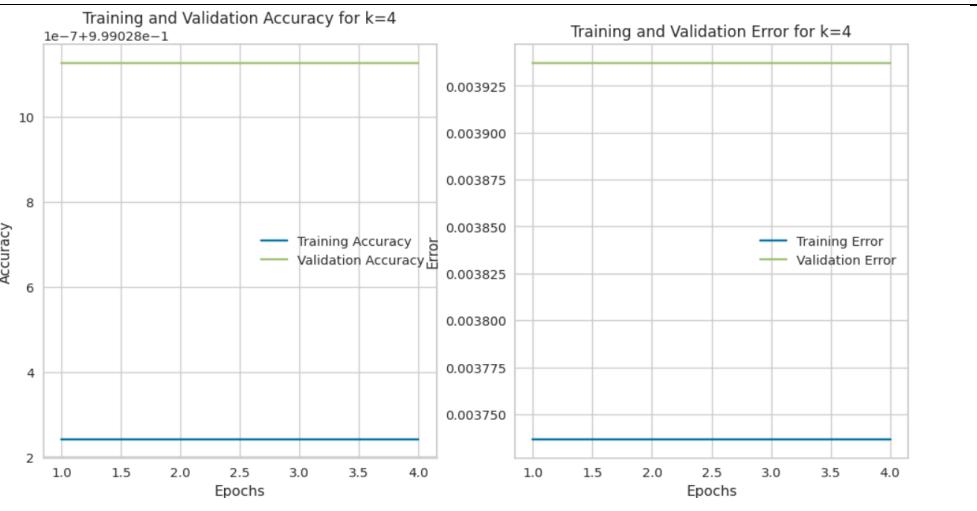
CV k=6	Best parameters found: { 'bootstrap': True, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 6, 'n_estimators': 100}	100%	100%	100%	
TFIDF k=6	Best parameters found: { 'bootstrap': True, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 8, 'n_estimators': 100}	99,5%	99,5%	100%	

b.1.3. Naïve Bayes

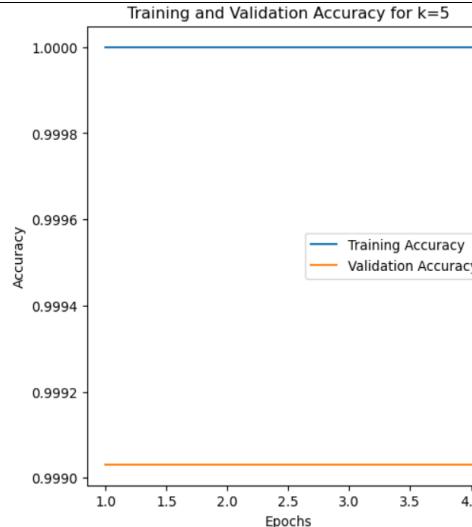
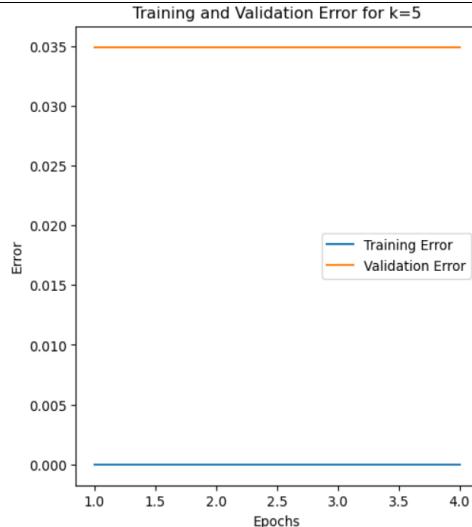
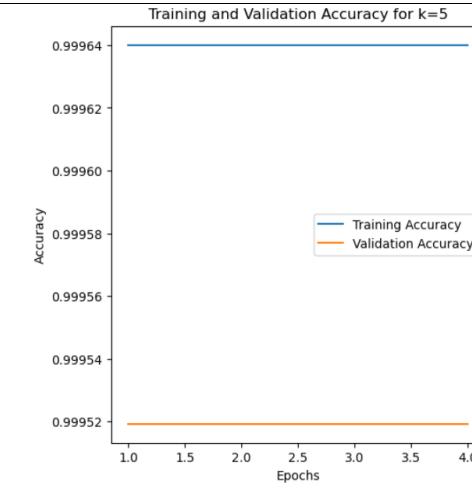
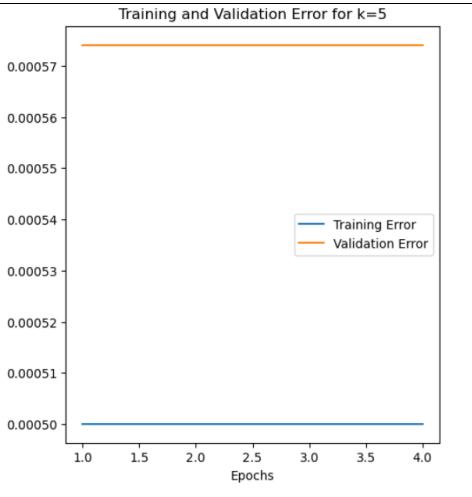
Tableau 6 : résultat CB/ML : modèle Naive Bayes

Feature Extraction	Grid Search	Precision	Recall	F1-Score	plot
CV k=3	Best parameters found: {'alpha': 0.001}	100%	100%	100%	
TFIDF k= 3	Best parameters found: {'alpha': 0.001}	100%	100%	100%	

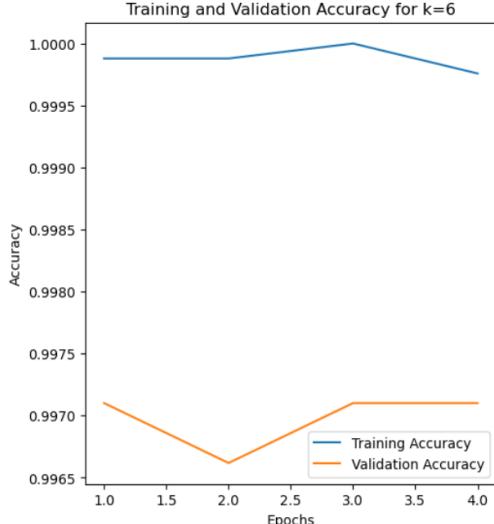
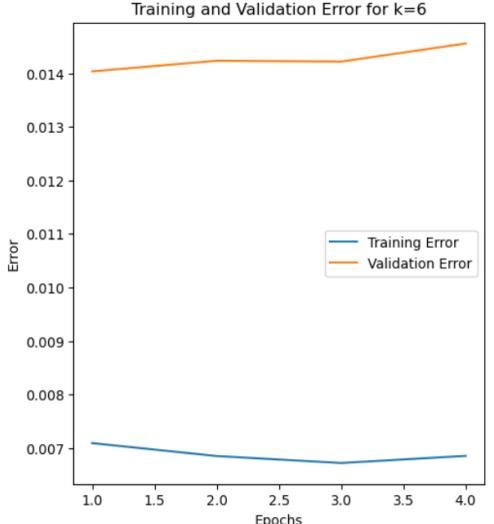
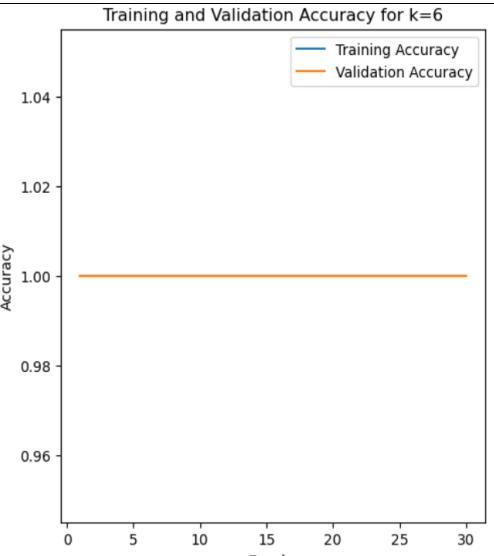
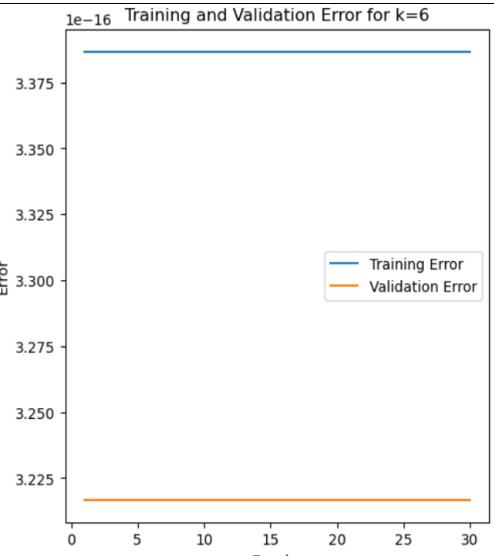
Caractérisation d'éléments transposables : les transposons à TIR

CV k=4	Best parameters found: {'alpha': 10.0}	100%	100%	100%	
TFIDF k=4	Best parameters found: {'alpha': 0.1}	100%	100%	100%	

Caractérisation d'éléments transposables : les transposons à TIR

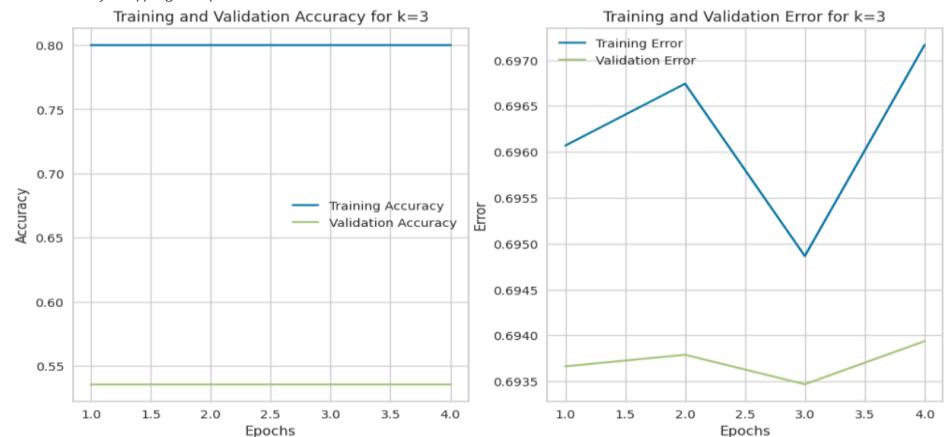
CV k=5	Best parameters found: {'alpha': 0.001}	100%	100%	100%		
TFIDF k=5	Best parameters found: {'alpha': 1.0}	100%	100%	100%		

Caractérisation d'éléments transposables : les transposons à TIR

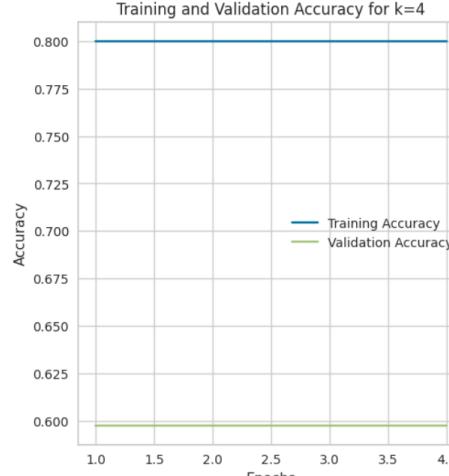
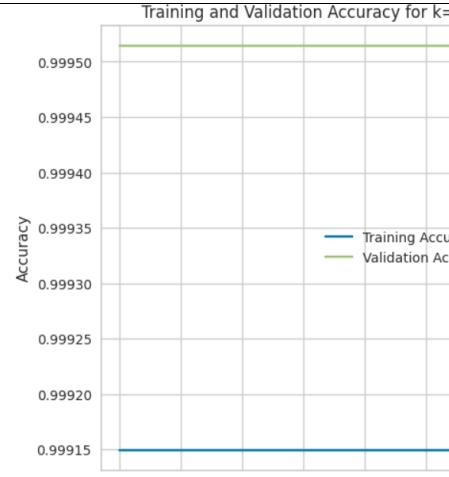
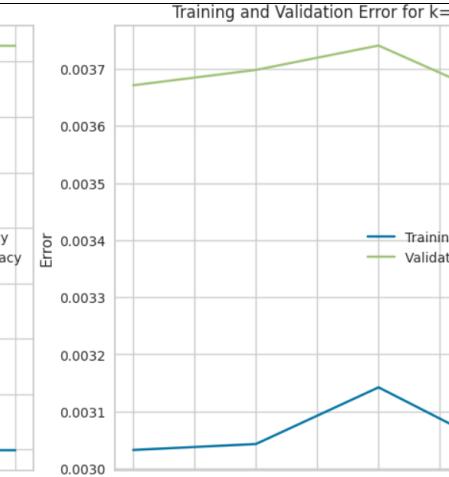
CV k=6	Best parameters found: {'alpha': 0.001}	99,5%	99,5%	100%	 <p>Training and Validation Accuracy for k=6</p> <table border="1"> <thead> <tr> <th>Epochs</th> <th>Training Accuracy</th> <th>Validation Accuracy</th> </tr> </thead> <tbody> <tr><td>1.0</td><td>0.9999</td><td>0.9970</td></tr> <tr><td>1.5</td><td>0.9999</td><td>0.9970</td></tr> <tr><td>2.0</td><td>0.9999</td><td>0.9966</td></tr> <tr><td>2.5</td><td>0.9999</td><td>0.9970</td></tr> <tr><td>3.0</td><td>1.0000</td><td>0.9972</td></tr> <tr><td>3.5</td><td>0.9999</td><td>0.9972</td></tr> <tr><td>4.0</td><td>0.9998</td><td>0.9972</td></tr> </tbody> </table>	Epochs	Training Accuracy	Validation Accuracy	1.0	0.9999	0.9970	1.5	0.9999	0.9970	2.0	0.9999	0.9966	2.5	0.9999	0.9970	3.0	1.0000	0.9972	3.5	0.9999	0.9972	4.0	0.9998	0.9972	 <p>Training and Validation Error for k=6</p> <table border="1"> <thead> <tr> <th>Epochs</th> <th>Training Error</th> <th>Validation Error</th> </tr> </thead> <tbody> <tr><td>1.0</td><td>0.0070</td><td>0.0140</td></tr> <tr><td>1.5</td><td>0.0068</td><td>0.0142</td></tr> <tr><td>2.0</td><td>0.0067</td><td>0.0143</td></tr> <tr><td>2.5</td><td>0.0066</td><td>0.0144</td></tr> <tr><td>3.0</td><td>0.0065</td><td>0.0145</td></tr> <tr><td>3.5</td><td>0.0065</td><td>0.0146</td></tr> <tr><td>4.0</td><td>0.0065</td><td>0.0147</td></tr> </tbody> </table>	Epochs	Training Error	Validation Error	1.0	0.0070	0.0140	1.5	0.0068	0.0142	2.0	0.0067	0.0143	2.5	0.0066	0.0144	3.0	0.0065	0.0145	3.5	0.0065	0.0146	4.0	0.0065	0.0147
Epochs	Training Accuracy	Validation Accuracy																																																				
1.0	0.9999	0.9970																																																				
1.5	0.9999	0.9970																																																				
2.0	0.9999	0.9966																																																				
2.5	0.9999	0.9970																																																				
3.0	1.0000	0.9972																																																				
3.5	0.9999	0.9972																																																				
4.0	0.9998	0.9972																																																				
Epochs	Training Error	Validation Error																																																				
1.0	0.0070	0.0140																																																				
1.5	0.0068	0.0142																																																				
2.0	0.0067	0.0143																																																				
2.5	0.0066	0.0144																																																				
3.0	0.0065	0.0145																																																				
3.5	0.0065	0.0146																																																				
4.0	0.0065	0.0147																																																				
TFIDF k=6	Best parameters found: {'alpha': 0.1}	100%	100%	100%	 <p>Training and Validation Accuracy for k=6</p> <table border="1"> <thead> <tr> <th>Epochs</th> <th>Training Accuracy</th> <th>Validation Accuracy</th> </tr> </thead> <tbody> <tr><td>0</td><td>1.0000</td><td>1.0000</td></tr> <tr><td>5</td><td>1.0000</td><td>1.0000</td></tr> <tr><td>10</td><td>1.0000</td><td>1.0000</td></tr> <tr><td>15</td><td>1.0000</td><td>1.0000</td></tr> <tr><td>20</td><td>1.0000</td><td>1.0000</td></tr> <tr><td>25</td><td>1.0000</td><td>1.0000</td></tr> <tr><td>30</td><td>1.0000</td><td>1.0000</td></tr> </tbody> </table>	Epochs	Training Accuracy	Validation Accuracy	0	1.0000	1.0000	5	1.0000	1.0000	10	1.0000	1.0000	15	1.0000	1.0000	20	1.0000	1.0000	25	1.0000	1.0000	30	1.0000	1.0000	 <p>1e-16 Training and Validation Error for k=6</p> <table border="1"> <thead> <tr> <th>Epochs</th> <th>Training Error</th> <th>Validation Error</th> </tr> </thead> <tbody> <tr><td>0</td><td>3.375e-16</td><td>3.225e-16</td></tr> <tr><td>5</td><td>3.375e-16</td><td>3.225e-16</td></tr> <tr><td>10</td><td>3.375e-16</td><td>3.225e-16</td></tr> <tr><td>15</td><td>3.375e-16</td><td>3.225e-16</td></tr> <tr><td>20</td><td>3.375e-16</td><td>3.225e-16</td></tr> <tr><td>25</td><td>3.375e-16</td><td>3.225e-16</td></tr> <tr><td>30</td><td>3.375e-16</td><td>3.225e-16</td></tr> </tbody> </table>	Epochs	Training Error	Validation Error	0	3.375e-16	3.225e-16	5	3.375e-16	3.225e-16	10	3.375e-16	3.225e-16	15	3.375e-16	3.225e-16	20	3.375e-16	3.225e-16	25	3.375e-16	3.225e-16	30	3.375e-16	3.225e-16
Epochs	Training Accuracy	Validation Accuracy																																																				
0	1.0000	1.0000																																																				
5	1.0000	1.0000																																																				
10	1.0000	1.0000																																																				
15	1.0000	1.0000																																																				
20	1.0000	1.0000																																																				
25	1.0000	1.0000																																																				
30	1.0000	1.0000																																																				
Epochs	Training Error	Validation Error																																																				
0	3.375e-16	3.225e-16																																																				
5	3.375e-16	3.225e-16																																																				
10	3.375e-16	3.225e-16																																																				
15	3.375e-16	3.225e-16																																																				
20	3.375e-16	3.225e-16																																																				
25	3.375e-16	3.225e-16																																																				
30	3.375e-16	3.225e-16																																																				

b.1.4. SVM

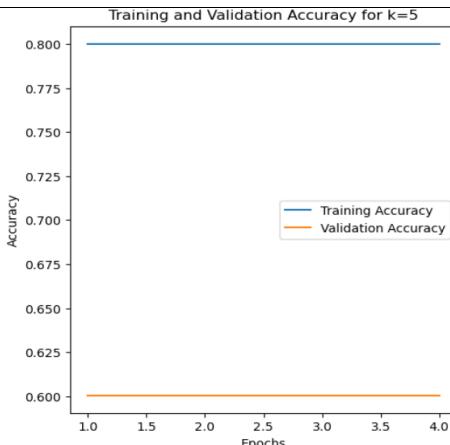
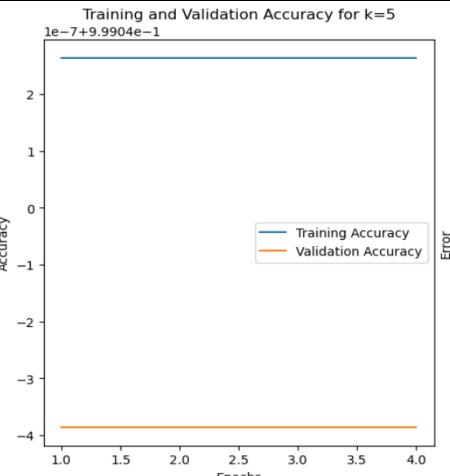
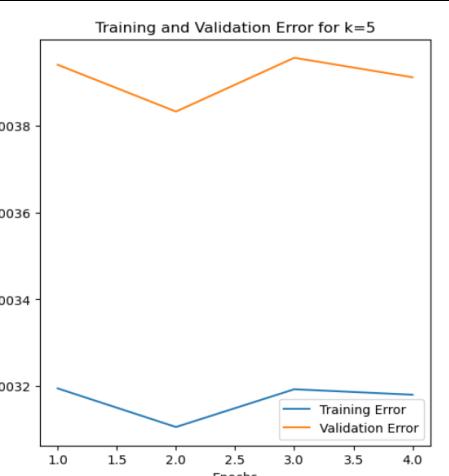
Tableau 7 : résultat CB/ML : modèle SVM

Feature Extraction	Grid Search	Precision	Recall	F1-Score	plot
CV k=3	Best parameters found: {'C': 0.1, 'gamma': 1, 'kernel': 'rbf'}	25,5%	50%	33,5%	
TFIDF k=3	Best parameters found: {'C': 0.1, 'gamma': 1, 'kernel': 'rbf'}	100%	100%	100%	

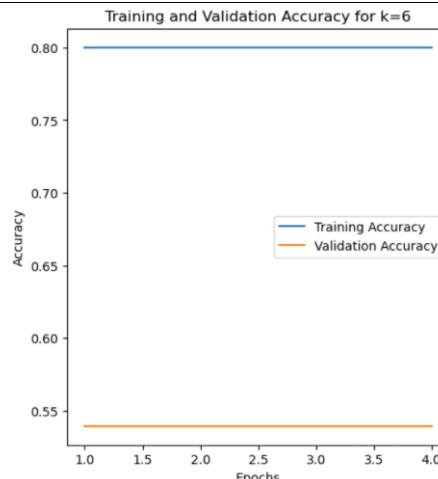
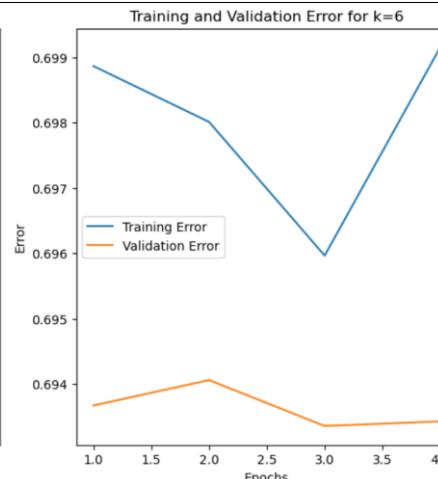
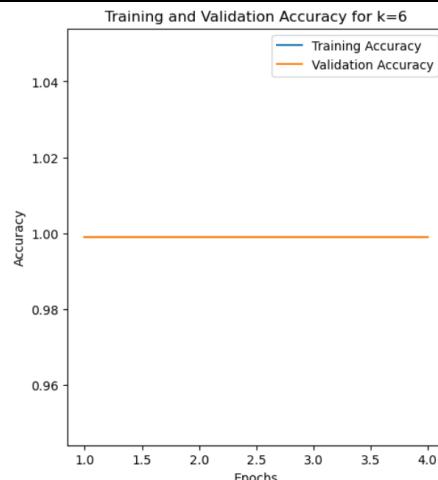
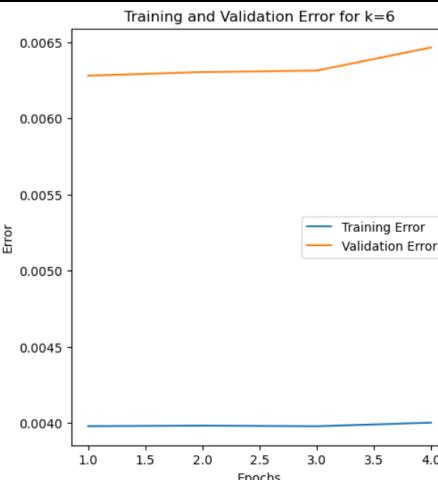
Caractérisation d'éléments transposables : les transposons à TIR

CV k=4	Best parameters found: {'C': 0.1, 'gamma': 1, 'kernel': 'rbf'}	100%	100%	100%	 <p>Training and Validation Accuracy for k=4</p> <table border="1"> <thead> <tr> <th>Epochs</th> <th>Training Accuracy</th> <th>Validation Accuracy</th> </tr> </thead> <tbody> <tr><td>1.0</td><td>0.800</td><td>0.600</td></tr> <tr><td>1.5</td><td>0.800</td><td>0.600</td></tr> <tr><td>2.0</td><td>0.800</td><td>0.600</td></tr> <tr><td>2.5</td><td>0.800</td><td>0.600</td></tr> <tr><td>3.0</td><td>0.800</td><td>0.600</td></tr> <tr><td>3.5</td><td>0.800</td><td>0.600</td></tr> <tr><td>4.0</td><td>0.800</td><td>0.600</td></tr> </tbody> </table>	Epochs	Training Accuracy	Validation Accuracy	1.0	0.800	0.600	1.5	0.800	0.600	2.0	0.800	0.600	2.5	0.800	0.600	3.0	0.800	0.600	3.5	0.800	0.600	4.0	0.800	0.600	 <p>Training and Validation Error for k=4</p> <table border="1"> <thead> <tr> <th>Epochs</th> <th>Training Error</th> <th>Validation Error</th> </tr> </thead> <tbody> <tr><td>1.0</td><td>0.702</td><td>0.693</td></tr> <tr><td>1.5</td><td>0.697</td><td>0.693</td></tr> <tr><td>2.0</td><td>0.696</td><td>0.693</td></tr> <tr><td>2.5</td><td>0.698</td><td>0.693</td></tr> <tr><td>3.0</td><td>0.699</td><td>0.693</td></tr> <tr><td>3.5</td><td>0.698</td><td>0.6935</td></tr> <tr><td>4.0</td><td>0.698</td><td>0.694</td></tr> </tbody> </table>	Epochs	Training Error	Validation Error	1.0	0.702	0.693	1.5	0.697	0.693	2.0	0.696	0.693	2.5	0.698	0.693	3.0	0.699	0.693	3.5	0.698	0.6935	4.0	0.698	0.694
Epochs	Training Accuracy	Validation Accuracy																																																				
1.0	0.800	0.600																																																				
1.5	0.800	0.600																																																				
2.0	0.800	0.600																																																				
2.5	0.800	0.600																																																				
3.0	0.800	0.600																																																				
3.5	0.800	0.600																																																				
4.0	0.800	0.600																																																				
Epochs	Training Error	Validation Error																																																				
1.0	0.702	0.693																																																				
1.5	0.697	0.693																																																				
2.0	0.696	0.693																																																				
2.5	0.698	0.693																																																				
3.0	0.699	0.693																																																				
3.5	0.698	0.6935																																																				
4.0	0.698	0.694																																																				
TFIDF k=4	Best parameters found: {'C': 0.1, 'gamma': 1, 'kernel': 'rbf'}	100%	100%	100%	 <p>Training and Validation Accuracy for k=4</p> <table border="1"> <thead> <tr> <th>Epochs</th> <th>Training Accuracy</th> <th>Validation Accuracy</th> </tr> </thead> <tbody> <tr><td>1.0</td><td>0.99915</td><td>0.99950</td></tr> <tr><td>1.5</td><td>0.99915</td><td>0.99950</td></tr> <tr><td>2.0</td><td>0.99915</td><td>0.99950</td></tr> <tr><td>2.5</td><td>0.99915</td><td>0.99950</td></tr> <tr><td>3.0</td><td>0.99915</td><td>0.99950</td></tr> <tr><td>3.5</td><td>0.99915</td><td>0.99950</td></tr> <tr><td>4.0</td><td>0.99915</td><td>0.99950</td></tr> </tbody> </table>	Epochs	Training Accuracy	Validation Accuracy	1.0	0.99915	0.99950	1.5	0.99915	0.99950	2.0	0.99915	0.99950	2.5	0.99915	0.99950	3.0	0.99915	0.99950	3.5	0.99915	0.99950	4.0	0.99915	0.99950	 <p>Training and Validation Error for k=4</p> <table border="1"> <thead> <tr> <th>Epochs</th> <th>Training Error</th> <th>Validation Error</th> </tr> </thead> <tbody> <tr><td>1.0</td><td>0.0030</td><td>0.00365</td></tr> <tr><td>1.5</td><td>0.00305</td><td>0.00368</td></tr> <tr><td>2.0</td><td>0.00308</td><td>0.00369</td></tr> <tr><td>2.5</td><td>0.00312</td><td>0.00372</td></tr> <tr><td>3.0</td><td>0.00315</td><td>0.00375</td></tr> <tr><td>3.5</td><td>0.00312</td><td>0.00372</td></tr> <tr><td>4.0</td><td>0.00308</td><td>0.00368</td></tr> </tbody> </table>	Epochs	Training Error	Validation Error	1.0	0.0030	0.00365	1.5	0.00305	0.00368	2.0	0.00308	0.00369	2.5	0.00312	0.00372	3.0	0.00315	0.00375	3.5	0.00312	0.00372	4.0	0.00308	0.00368
Epochs	Training Accuracy	Validation Accuracy																																																				
1.0	0.99915	0.99950																																																				
1.5	0.99915	0.99950																																																				
2.0	0.99915	0.99950																																																				
2.5	0.99915	0.99950																																																				
3.0	0.99915	0.99950																																																				
3.5	0.99915	0.99950																																																				
4.0	0.99915	0.99950																																																				
Epochs	Training Error	Validation Error																																																				
1.0	0.0030	0.00365																																																				
1.5	0.00305	0.00368																																																				
2.0	0.00308	0.00369																																																				
2.5	0.00312	0.00372																																																				
3.0	0.00315	0.00375																																																				
3.5	0.00312	0.00372																																																				
4.0	0.00308	0.00368																																																				

Caractérisation d'éléments transposables : les transposons à TIR

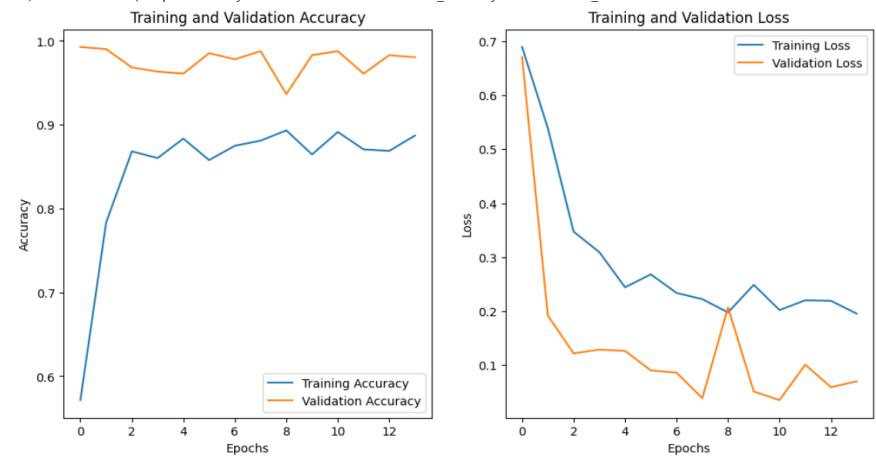
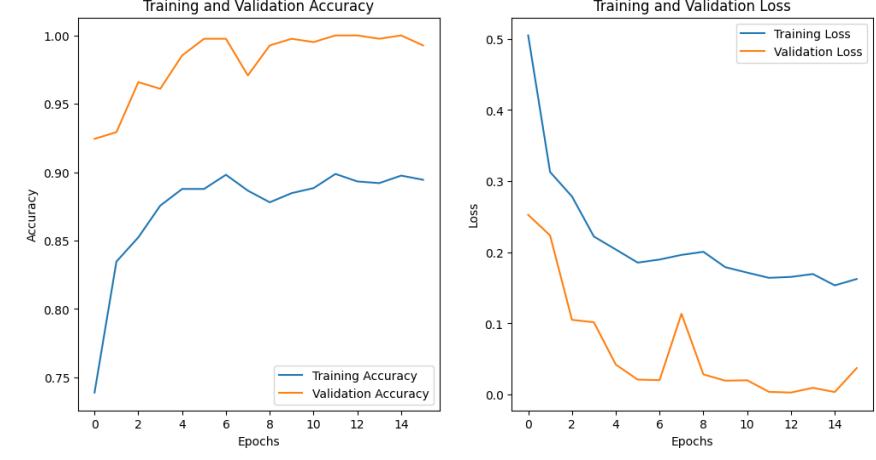
CV k=5	Best parameters found: {'C': 0.1, 'gamma': 1, 'kernel': 'rbf'}	76%	52%	38,5%	 
TFiDF k=5	Best parameters found: {'C': 0.1, 'gamma': 1, 'kernel': 'rbf'}	100%	100%	100%	 

Caractérisation d'éléments transposables : les transposons à TIR

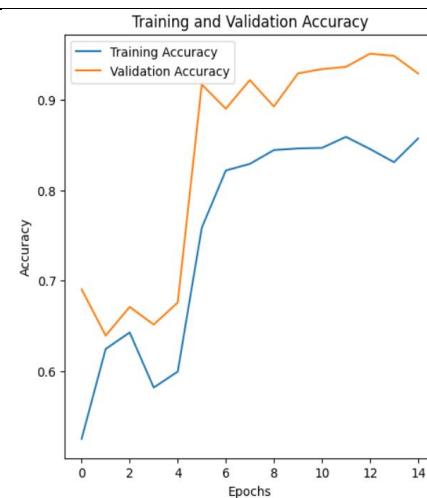
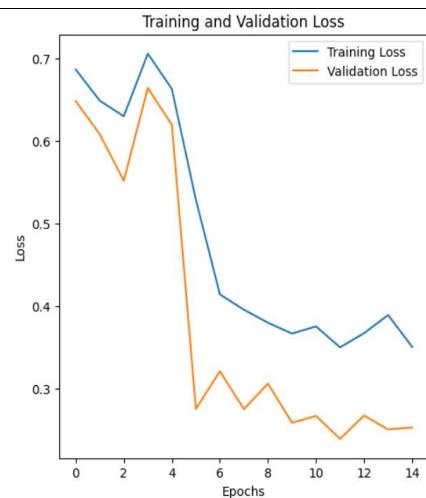
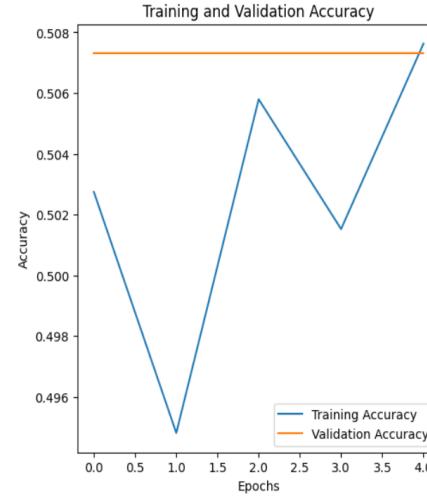
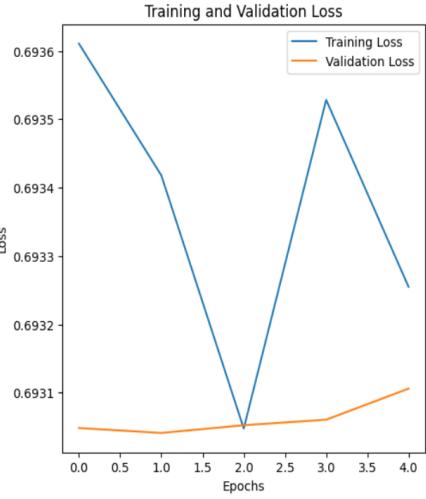
CV k=6	Best parameters found: {'C': 0.1, 'gamma': 1, 'kernel': 'rbf'}	72,5%	67%	63,5%	 
TFIDF k=6	Best parameters found: {'C': 0.1, 'gamma': 1, 'kernel': 'rbf'}	100%	100%	100%	 

b.2. DL : LSTM

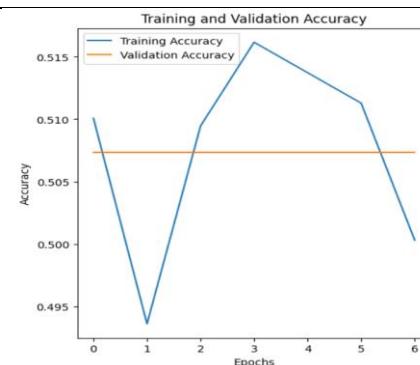
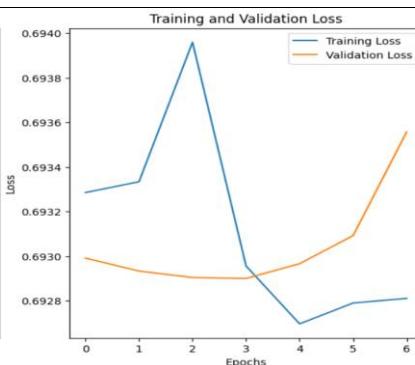
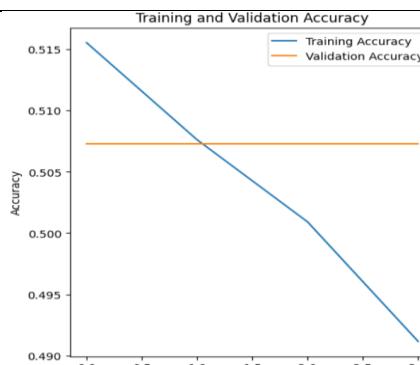
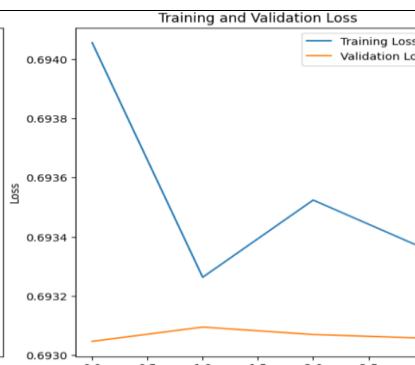
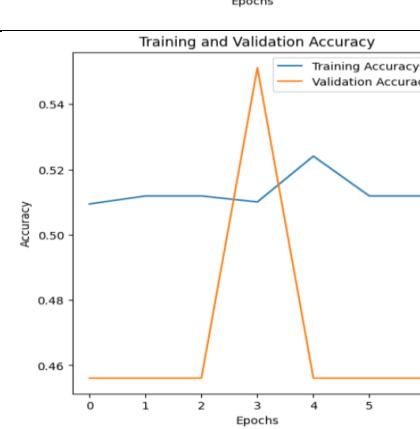
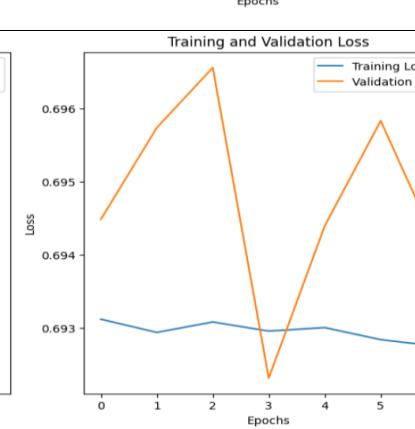
Tableau 8 : résultat CB/DL : modèle LSTM

Feature Extraction	accuracy	Precision	Recall	F1-Score	plot																																																																																																
CV k=3	99%	100%	100%	100%	 <p>Training and Validation Accuracy</p> <table border="1"> <thead> <tr> <th>Epochs</th> <th>Training Accuracy</th> <th>Validation Accuracy</th> </tr> </thead> <tbody> <tr><td>0</td><td>0.60</td><td>0.99</td></tr> <tr><td>1</td><td>0.75</td><td>0.98</td></tr> <tr><td>2</td><td>0.85</td><td>0.97</td></tr> <tr><td>3</td><td>0.88</td><td>0.96</td></tr> <tr><td>4</td><td>0.89</td><td>0.95</td></tr> <tr><td>5</td><td>0.87</td><td>0.97</td></tr> <tr><td>6</td><td>0.88</td><td>0.98</td></tr> <tr><td>7</td><td>0.89</td><td>0.97</td></tr> <tr><td>8</td><td>0.87</td><td>0.96</td></tr> <tr><td>9</td><td>0.88</td><td>0.98</td></tr> <tr><td>10</td><td>0.89</td><td>0.97</td></tr> <tr><td>11</td><td>0.87</td><td>0.98</td></tr> <tr><td>12</td><td>0.88</td><td>0.97</td></tr> </tbody> </table> <p>Training and Validation Loss</p> <table border="1"> <thead> <tr> <th>Epochs</th> <th>Training Loss</th> <th>Validation Loss</th> </tr> </thead> <tbody> <tr><td>0</td><td>0.70</td><td>0.68</td></tr> <tr><td>1</td><td>0.45</td><td>0.18</td></tr> <tr><td>2</td><td>0.35</td><td>0.12</td></tr> <tr><td>3</td><td>0.28</td><td>0.15</td></tr> <tr><td>4</td><td>0.25</td><td>0.12</td></tr> <tr><td>5</td><td>0.27</td><td>0.08</td></tr> <tr><td>6</td><td>0.24</td><td>0.08</td></tr> <tr><td>7</td><td>0.22</td><td>0.05</td></tr> <tr><td>8</td><td>0.25</td><td>0.20</td></tr> <tr><td>9</td><td>0.22</td><td>0.05</td></tr> <tr><td>10</td><td>0.21</td><td>0.03</td></tr> <tr><td>11</td><td>0.23</td><td>0.10</td></tr> <tr><td>12</td><td>0.22</td><td>0.08</td></tr> </tbody> </table>	Epochs	Training Accuracy	Validation Accuracy	0	0.60	0.99	1	0.75	0.98	2	0.85	0.97	3	0.88	0.96	4	0.89	0.95	5	0.87	0.97	6	0.88	0.98	7	0.89	0.97	8	0.87	0.96	9	0.88	0.98	10	0.89	0.97	11	0.87	0.98	12	0.88	0.97	Epochs	Training Loss	Validation Loss	0	0.70	0.68	1	0.45	0.18	2	0.35	0.12	3	0.28	0.15	4	0.25	0.12	5	0.27	0.08	6	0.24	0.08	7	0.22	0.05	8	0.25	0.20	9	0.22	0.05	10	0.21	0.03	11	0.23	0.10	12	0.22	0.08												
Epochs	Training Accuracy	Validation Accuracy																																																																																																			
0	0.60	0.99																																																																																																			
1	0.75	0.98																																																																																																			
2	0.85	0.97																																																																																																			
3	0.88	0.96																																																																																																			
4	0.89	0.95																																																																																																			
5	0.87	0.97																																																																																																			
6	0.88	0.98																																																																																																			
7	0.89	0.97																																																																																																			
8	0.87	0.96																																																																																																			
9	0.88	0.98																																																																																																			
10	0.89	0.97																																																																																																			
11	0.87	0.98																																																																																																			
12	0.88	0.97																																																																																																			
Epochs	Training Loss	Validation Loss																																																																																																			
0	0.70	0.68																																																																																																			
1	0.45	0.18																																																																																																			
2	0.35	0.12																																																																																																			
3	0.28	0.15																																																																																																			
4	0.25	0.12																																																																																																			
5	0.27	0.08																																																																																																			
6	0.24	0.08																																																																																																			
7	0.22	0.05																																																																																																			
8	0.25	0.20																																																																																																			
9	0.22	0.05																																																																																																			
10	0.21	0.03																																																																																																			
11	0.23	0.10																																																																																																			
12	0.22	0.08																																																																																																			
TFIDF k = 3	98,4%	98,5%	98,5%	98,5%	 <p>Training and Validation Accuracy</p> <table border="1"> <thead> <tr> <th>Epochs</th> <th>Training Accuracy</th> <th>Validation Accuracy</th> </tr> </thead> <tbody> <tr><td>0</td><td>0.75</td><td>0.92</td></tr> <tr><td>1</td><td>0.84</td><td>0.93</td></tr> <tr><td>2</td><td>0.87</td><td>0.96</td></tr> <tr><td>3</td><td>0.89</td><td>0.98</td></tr> <tr><td>4</td><td>0.90</td><td>0.99</td></tr> <tr><td>5</td><td>0.91</td><td>1.00</td></tr> <tr><td>6</td><td>0.90</td><td>0.99</td></tr> <tr><td>7</td><td>0.89</td><td>0.98</td></tr> <tr><td>8</td><td>0.88</td><td>0.99</td></tr> <tr><td>9</td><td>0.89</td><td>1.00</td></tr> <tr><td>10</td><td>0.90</td><td>1.00</td></tr> <tr><td>11</td><td>0.91</td><td>1.00</td></tr> <tr><td>12</td><td>0.90</td><td>1.00</td></tr> <tr><td>13</td><td>0.90</td><td>1.00</td></tr> <tr><td>14</td><td>0.91</td><td>1.00</td></tr> </tbody> </table> <p>Training and Validation Loss</p> <table border="1"> <thead> <tr> <th>Epochs</th> <th>Training Loss</th> <th>Validation Loss</th> </tr> </thead> <tbody> <tr><td>0</td><td>0.50</td><td>0.30</td></tr> <tr><td>1</td><td>0.32</td><td>0.25</td></tr> <tr><td>2</td><td>0.25</td><td>0.10</td></tr> <tr><td>3</td><td>0.20</td><td>0.05</td></tr> <tr><td>4</td><td>0.18</td><td>0.02</td></tr> <tr><td>5</td><td>0.17</td><td>0.01</td></tr> <tr><td>6</td><td>0.16</td><td>0.01</td></tr> <tr><td>7</td><td>0.18</td><td>0.10</td></tr> <tr><td>8</td><td>0.15</td><td>0.02</td></tr> <tr><td>9</td><td>0.14</td><td>0.01</td></tr> <tr><td>10</td><td>0.13</td><td>0.01</td></tr> <tr><td>11</td><td>0.14</td><td>0.01</td></tr> <tr><td>12</td><td>0.13</td><td>0.01</td></tr> <tr><td>13</td><td>0.14</td><td>0.02</td></tr> <tr><td>14</td><td>0.15</td><td>0.03</td></tr> </tbody> </table>	Epochs	Training Accuracy	Validation Accuracy	0	0.75	0.92	1	0.84	0.93	2	0.87	0.96	3	0.89	0.98	4	0.90	0.99	5	0.91	1.00	6	0.90	0.99	7	0.89	0.98	8	0.88	0.99	9	0.89	1.00	10	0.90	1.00	11	0.91	1.00	12	0.90	1.00	13	0.90	1.00	14	0.91	1.00	Epochs	Training Loss	Validation Loss	0	0.50	0.30	1	0.32	0.25	2	0.25	0.10	3	0.20	0.05	4	0.18	0.02	5	0.17	0.01	6	0.16	0.01	7	0.18	0.10	8	0.15	0.02	9	0.14	0.01	10	0.13	0.01	11	0.14	0.01	12	0.13	0.01	13	0.14	0.02	14	0.15	0.03
Epochs	Training Accuracy	Validation Accuracy																																																																																																			
0	0.75	0.92																																																																																																			
1	0.84	0.93																																																																																																			
2	0.87	0.96																																																																																																			
3	0.89	0.98																																																																																																			
4	0.90	0.99																																																																																																			
5	0.91	1.00																																																																																																			
6	0.90	0.99																																																																																																			
7	0.89	0.98																																																																																																			
8	0.88	0.99																																																																																																			
9	0.89	1.00																																																																																																			
10	0.90	1.00																																																																																																			
11	0.91	1.00																																																																																																			
12	0.90	1.00																																																																																																			
13	0.90	1.00																																																																																																			
14	0.91	1.00																																																																																																			
Epochs	Training Loss	Validation Loss																																																																																																			
0	0.50	0.30																																																																																																			
1	0.32	0.25																																																																																																			
2	0.25	0.10																																																																																																			
3	0.20	0.05																																																																																																			
4	0.18	0.02																																																																																																			
5	0.17	0.01																																																																																																			
6	0.16	0.01																																																																																																			
7	0.18	0.10																																																																																																			
8	0.15	0.02																																																																																																			
9	0.14	0.01																																																																																																			
10	0.13	0.01																																																																																																			
11	0.14	0.01																																																																																																			
12	0.13	0.01																																																																																																			
13	0.14	0.02																																																																																																			
14	0.15	0.03																																																																																																			

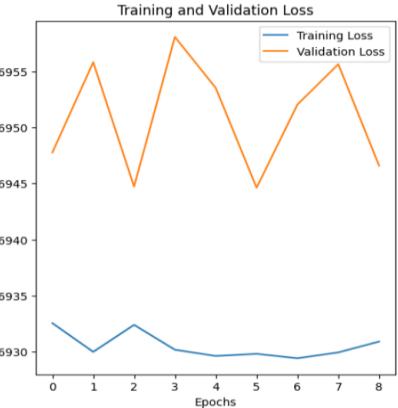
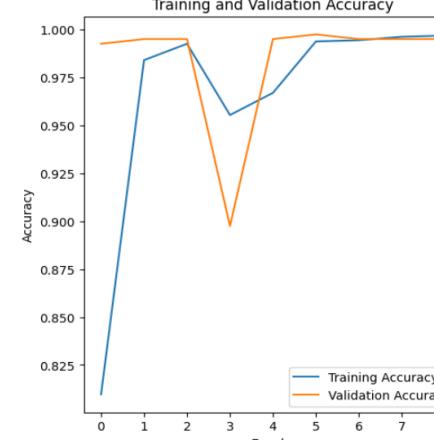
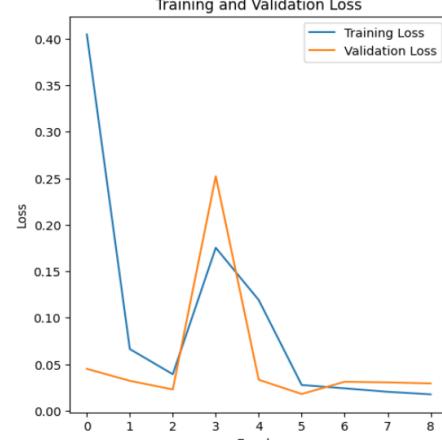
Caractérisation d'éléments transposables : les transposons à TIR

CV k=4	95,3%	96%	95%	95,5%	 
TFiDF k=4	46%	23,5%	50%	32%	 

Caractérisation d'éléments transposables : les transposons à TIR

CV k=5	46%	23%	50%	32%	 
TFIDF k=5	46%	23,5%	50%	32%	 
CV k=6	50%	45%	50%	34,5%	 

Caractérisation d'éléments transposables : les transposons à TIR

TFIDF k=6	49,7%	25%	50%	33%	 <p>Training and Validation Accuracy</p> <table border="1"> <thead> <tr> <th>Epochs</th> <th>Training Accuracy</th> <th>Validation Accuracy</th> </tr> </thead> <tbody> <tr><td>0</td><td>0.51</td><td>0.46</td></tr> <tr><td>1</td><td>0.51</td><td>0.46</td></tr> <tr><td>2</td><td>0.51</td><td>0.46</td></tr> <tr><td>3</td><td>0.51</td><td>0.46</td></tr> <tr><td>4</td><td>0.51</td><td>0.46</td></tr> <tr><td>5</td><td>0.51</td><td>0.46</td></tr> <tr><td>6</td><td>0.51</td><td>0.46</td></tr> <tr><td>7</td><td>0.51</td><td>0.46</td></tr> <tr><td>8</td><td>0.51</td><td>0.46</td></tr> </tbody> </table>  <p>Training and Validation Loss</p> <table border="1"> <thead> <tr> <th>Epochs</th> <th>Training Loss</th> <th>Validation Loss</th> </tr> </thead> <tbody> <tr><td>0</td><td>0.6932</td><td>0.6946</td></tr> <tr><td>1</td><td>0.6931</td><td>0.6953</td></tr> <tr><td>2</td><td>0.6932</td><td>0.6945</td></tr> <tr><td>3</td><td>0.6931</td><td>0.6955</td></tr> <tr><td>4</td><td>0.6931</td><td>0.6948</td></tr> <tr><td>5</td><td>0.6931</td><td>0.6945</td></tr> <tr><td>6</td><td>0.6931</td><td>0.6952</td></tr> <tr><td>7</td><td>0.6931</td><td>0.6954</td></tr> <tr><td>8</td><td>0.6931</td><td>0.6946</td></tr> </tbody> </table>	Epochs	Training Accuracy	Validation Accuracy	0	0.51	0.46	1	0.51	0.46	2	0.51	0.46	3	0.51	0.46	4	0.51	0.46	5	0.51	0.46	6	0.51	0.46	7	0.51	0.46	8	0.51	0.46	Epochs	Training Loss	Validation Loss	0	0.6932	0.6946	1	0.6931	0.6953	2	0.6932	0.6945	3	0.6931	0.6955	4	0.6931	0.6948	5	0.6931	0.6945	6	0.6931	0.6952	7	0.6931	0.6954	8	0.6931	0.6946
Epochs	Training Accuracy	Validation Accuracy																																																															
0	0.51	0.46																																																															
1	0.51	0.46																																																															
2	0.51	0.46																																																															
3	0.51	0.46																																																															
4	0.51	0.46																																																															
5	0.51	0.46																																																															
6	0.51	0.46																																																															
7	0.51	0.46																																																															
8	0.51	0.46																																																															
Epochs	Training Loss	Validation Loss																																																															
0	0.6932	0.6946																																																															
1	0.6931	0.6953																																																															
2	0.6932	0.6945																																																															
3	0.6931	0.6955																																																															
4	0.6931	0.6948																																																															
5	0.6931	0.6945																																																															
6	0.6931	0.6952																																																															
7	0.6931	0.6954																																																															
8	0.6931	0.6946																																																															
embedding	99%	99.5%	99,5%	100%	 <p>Training and Validation Accuracy</p> <table border="1"> <thead> <tr> <th>Epochs</th> <th>Training Accuracy</th> <th>Validation Accuracy</th> </tr> </thead> <tbody> <tr><td>0</td><td>0.825</td><td>0.900</td></tr> <tr><td>1</td><td>0.980</td><td>0.990</td></tr> <tr><td>2</td><td>0.995</td><td>0.995</td></tr> <tr><td>3</td><td>0.955</td><td>0.890</td></tr> <tr><td>4</td><td>0.970</td><td>0.995</td></tr> <tr><td>5</td><td>0.990</td><td>0.995</td></tr> <tr><td>6</td><td>0.995</td><td>0.995</td></tr> <tr><td>7</td><td>0.995</td><td>0.995</td></tr> <tr><td>8</td><td>0.995</td><td>0.995</td></tr> </tbody> </table>  <p>Training and Validation Loss</p> <table border="1"> <thead> <tr> <th>Epochs</th> <th>Training Loss</th> <th>Validation Loss</th> </tr> </thead> <tbody> <tr><td>0</td><td>0.40</td><td>0.04</td></tr> <tr><td>1</td><td>0.08</td><td>0.02</td></tr> <tr><td>2</td><td>0.03</td><td>0.02</td></tr> <tr><td>3</td><td>0.17</td><td>0.25</td></tr> <tr><td>4</td><td>0.12</td><td>0.02</td></tr> <tr><td>5</td><td>0.02</td><td>0.02</td></tr> <tr><td>6</td><td>0.02</td><td>0.02</td></tr> <tr><td>7</td><td>0.02</td><td>0.02</td></tr> <tr><td>8</td><td>0.02</td><td>0.02</td></tr> </tbody> </table>	Epochs	Training Accuracy	Validation Accuracy	0	0.825	0.900	1	0.980	0.990	2	0.995	0.995	3	0.955	0.890	4	0.970	0.995	5	0.990	0.995	6	0.995	0.995	7	0.995	0.995	8	0.995	0.995	Epochs	Training Loss	Validation Loss	0	0.40	0.04	1	0.08	0.02	2	0.03	0.02	3	0.17	0.25	4	0.12	0.02	5	0.02	0.02	6	0.02	0.02	7	0.02	0.02	8	0.02	0.02
Epochs	Training Accuracy	Validation Accuracy																																																															
0	0.825	0.900																																																															
1	0.980	0.990																																																															
2	0.995	0.995																																																															
3	0.955	0.890																																																															
4	0.970	0.995																																																															
5	0.990	0.995																																																															
6	0.995	0.995																																																															
7	0.995	0.995																																																															
8	0.995	0.995																																																															
Epochs	Training Loss	Validation Loss																																																															
0	0.40	0.04																																																															
1	0.08	0.02																																																															
2	0.03	0.02																																																															
3	0.17	0.25																																																															
4	0.12	0.02																																																															
5	0.02	0.02																																																															
6	0.02	0.02																																																															
7	0.02	0.02																																																															
8	0.02	0.02																																																															

b.3. Discutions des résultats

Les résultats obtenus dans cette étude montrent des performances très élevées pour la majorité des algorithmes testés, notamment les modèles RandomForest, ExtraTrees, et Naive Bayes, qui affichent souvent des scores parfaits de précision, rappel, et F1-score. Cependant, cette perfection apparente est le signe d'un surapprentissage (overfitting), où les modèles s'ajustent trop précisément aux données d'entraînement, au détriment de leur capacité à généraliser sur des données nouvelles et non vues.

Les résultats de l'analyse des modèles de deep learning montrent des performances très variées en fonction des méthodes d'extraction de caractéristiques et des configurations utilisées. Voici une synthèse des principales observations :

1. **LSTM avec embedding** : Ce modèle a démontré des performances exceptionnelles, atteignant une précision, un rappel et un score F1 presque parfaits (près de 99-100%). Cela suggère que le modèle est extrêmement efficace pour capturer les séquences et les relations dans les données, en particulier lorsqu'il est combiné avec des embeddings. Cependant, il est important de vérifier que ces résultats ne sont pas dus à un surapprentissage, ce qui pourrait expliquer une performance si élevée.
2. **Extraction de caractéristiques avec CountVectorizer (CV) et TF-IDF** : Les résultats montrent des performances très disparates selon les configurations :
 - Pour certaines configurations (par exemple, CV avec k=6 ou 4), les modèles montrent des performances très faibles, avec des scores F1 aussi bas que 32-34%. Cela peut indiquer que les données sont mal représentées par ces méthodes d'extraction de caractéristiques pour ces configurations spécifiques, ou que les modèles ne parviennent pas à capturer les nuances présentes dans les données.
 - D'autres configurations avec CV ou TF-IDF, comme CV=3 ou TF-IDF=3, montrent des performances remarquablement élevées, avec des scores proches de 99-100%. Cela pourrait indiquer que pour certaines valeurs de k ou certaines configurations de TF-IDF, le modèle parvient à capturer les informations pertinentes dans les données.

b.4. Résultat final

En termes de k pour les autres méthodes d'extraction de caractéristiques (comme CountVectorizer et TF-IDF), le modèle LSTM qui se distingue avec un très bon score est celui

utilisant **TF-IDF avec k=3**. Ce modèle affiche également des performances élevées avec des scores proches de 98.5% pour la précision, le rappel et le F1-score.

L'architecture du modèle que vous avez présenté comporte **7 couches** principales. Voici le décompte :

1. **2 couches LSTM** (lstm_14 et lstm_15)
2. **3 couches Dropout** (dropout_21, dropout_22, et dropout_23)
3. **2 couches Dense** (dense_14 et dense_15)

Layer (type)	Output Shape	Param #
lstm_14 (LSTM)	(None, 125, 100)	40,800
dropout_21 (Dropout)	(None, 125, 100)	0
lstm_15 (LSTM)	(None, 50)	30,200
dropout_22 (Dropout)	(None, 50)	0
dense_14 (Dense)	(None, 50)	2,550
dropout_23 (Dropout)	(None, 50)	0
dense_15 (Dense)	(None, 1)	51

Figure 40 : architecture modèle choisi pour CB : LSTM / 5-mers

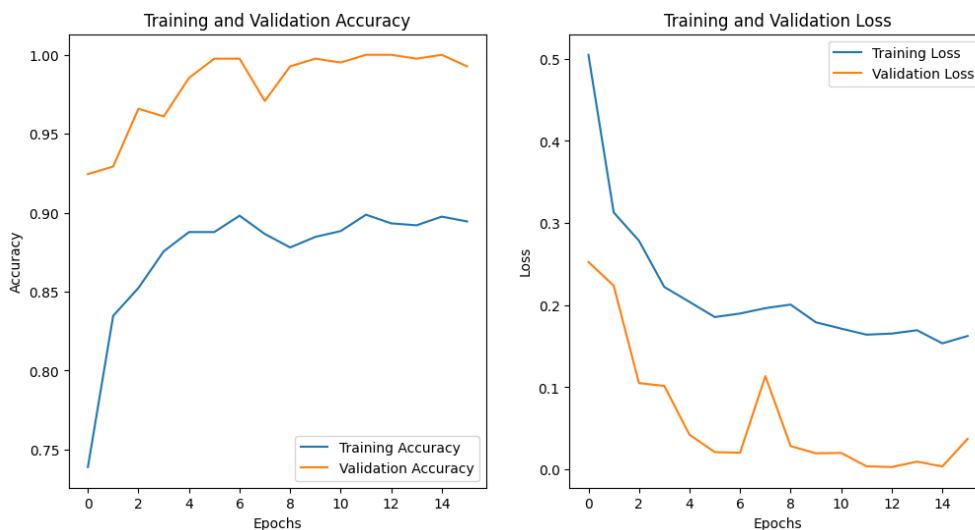


Figure 41 : plot Accuracy/Loss du modèle choisi pour CB : LSTM / 5-mers

Cette image montre deux graphiques côté à côté pour suivre l'évolution de la précision (accuracy) et de la perte (loss) pendant l'entraînement et la validation d'un modèle.

- **Graphique de gauche (Accuracy) :**
 - **Interprétation :**

- La précision d'entraînement augmente et se stabilise autour de 0.95 à partir de l'époque 6.
- La précision de validation reste stable autour de 0.95, indiquant que le modèle ne semble pas souffrir de surapprentissage (overfitting).
- **Graphique de droite (Loss) :**

- **Interprétation :**

- La perte d'entraînement diminue rapidement et se stabilise autour de 0.1 après quelques époques, ce qui est un bon signe que le modèle apprend efficacement.
- La perte de validation diminue également, mais avec plus de fluctuations, ce qui peut indiquer quelques variations dans la performance de validation, mais globalement, la perte reste basse.

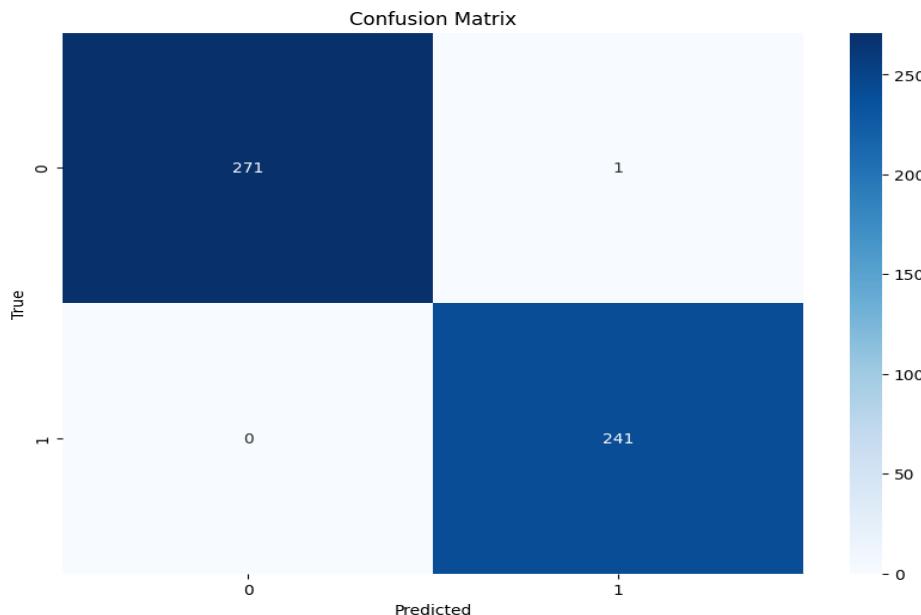


Figure 42 : matrice de confusion du modèle choisi pour CB : LSTM / 5-mers

Cette image représente une matrice de confusion, qui est utilisée pour évaluer les performances d'un modèle de classification binaire.

- **0 (Négatif) vs 1 (Positif)** : Les valeurs 0 et 1 représentent les deux classes, souvent nommées comme classe 0 (EnT) et classe 1 (ET).
- **Interprétation des cases :**
 - **271 (en haut à gauche)** : Le modèle a correctement prédit 271 exemples comme étant de la classe EnT (0).
 - **1 (en haut à droite)** : Il y a 1 exemple où le modèle a prédit la classe ET (1) alors qu'il s'agissait de la classe EnT (0) (faux positif).

- **0 (en bas à gauche)** : Il n'y a pas de faux négatifs, c'est-à-dire qu'il n'y a aucun exemple où la classe réelle est 1 mais le modèle a prédit 0.
- **241 (en bas à droite)** : Le modèle a correctement prédit 241 exemples comme étant de la classe ET (1).
- **Conclusion** : Le modèle montre une excellente performance avec très peu d'erreurs (seulement 1 faux positif).

c. Classification multi-classe

c.1. ML

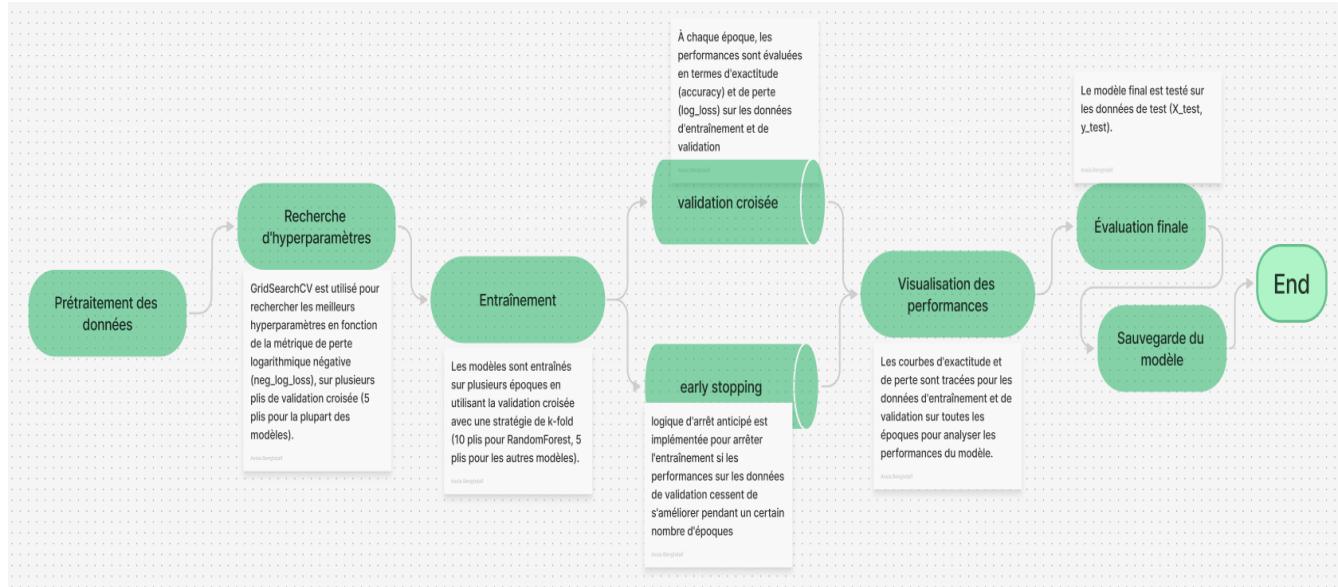


Figure 43 : logigramme pour la CM : partie ML

Dans la partie de prétraitement des données, nous avons d'abord appliqué l'analyse en composantes principales (PCA) sur les données après les avoir converties en une matrice dense. Nous avons déterminé le nombre optimal de composantes à conserver en fonction de la variance cumulative expliquée, afin de capturer 95% de la variance totale. Ensuite, nous avons réduit les données en fonction de ces composantes principales sélectionnées. Après la réduction dimensionnelle, nous avons calculé la matrice de corrélation des nouvelles composantes et nous avons identifié les paires de composantes hautement corrélées en utilisant un seuil de 0,8. Pour chaque paire fortement corrélée, nous avons conservé la composante avec le plus faible indice et nous avons supprimé l'autre. Finalement, nous avons filtré le jeu de données pour ne conserver que les composantes sélectionnées.

- **Exemple**

Par exemple dans 7-mers pour CountVectorizer la taille des features étaient 53609 ➔ 1700 d'après le graphe du PCA suivant :

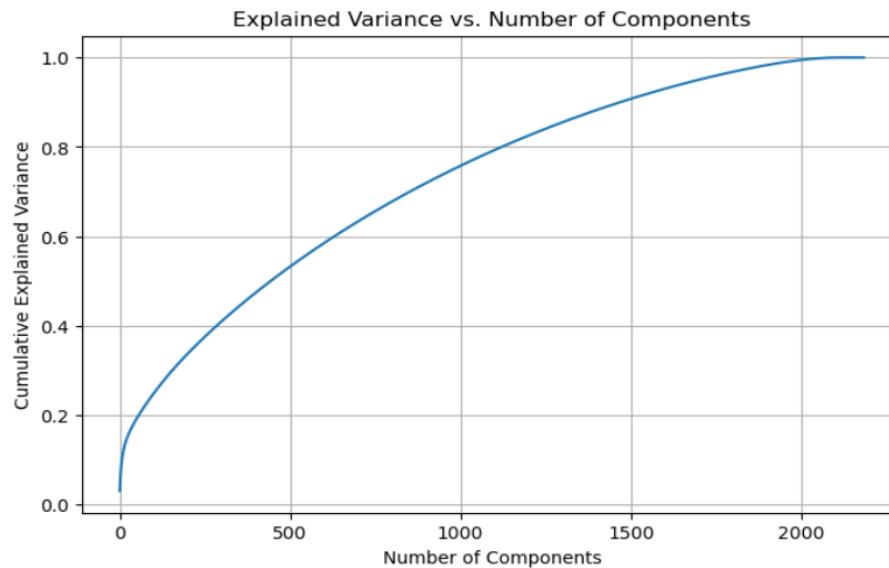
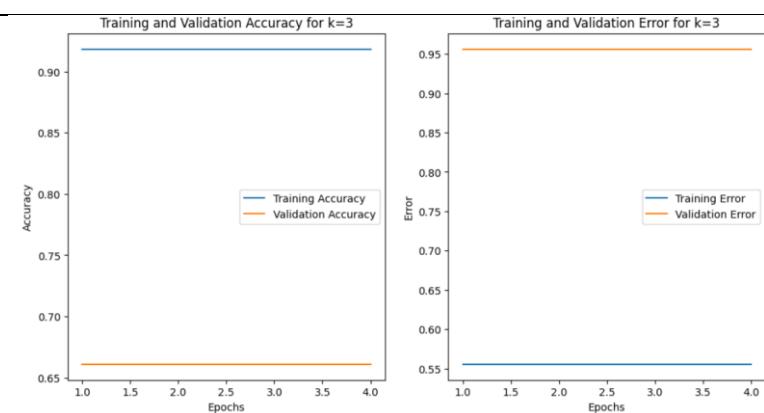
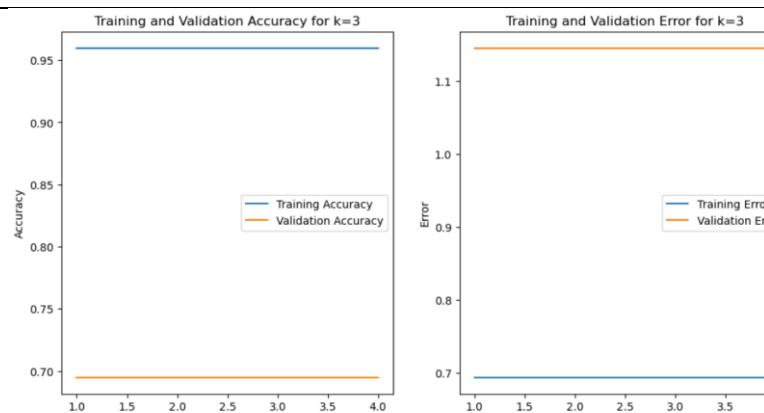


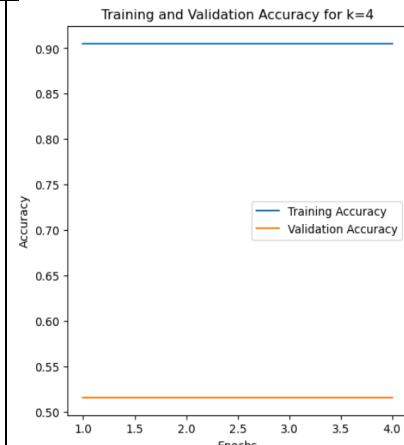
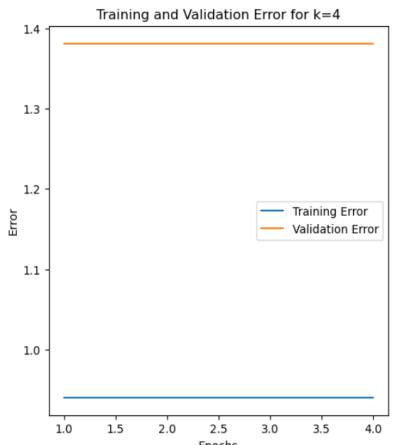
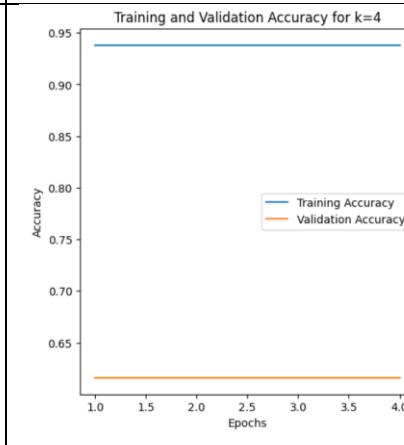
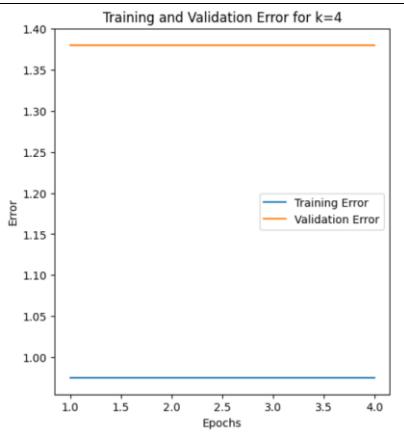
Figure 44 : exemple de graphe PCA pour 7-mers CV

c.1.1. Random Forest

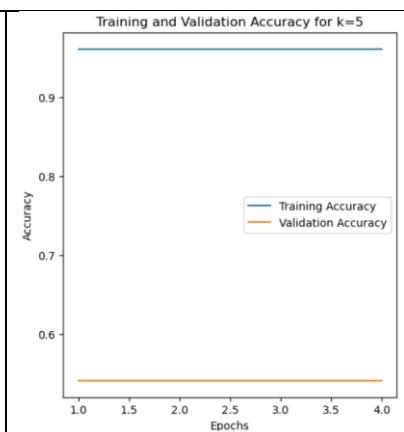
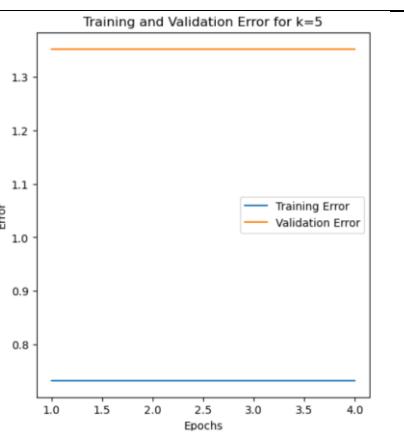
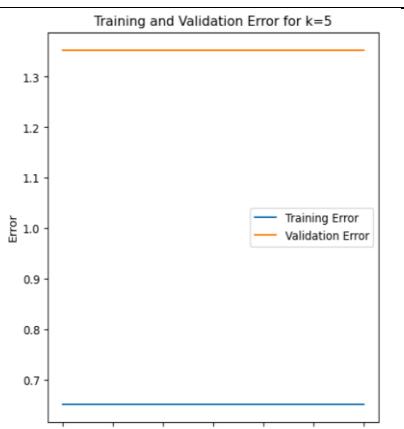
Tableau 9 : résultat CM/ML : modèle Random Forest

Feature Extraction	Grid Search	Accuracy	Precision	Recall	F1-Score	plot
CV k=3	Best parameters found: { 'bootstrap': False, 'max_depth': 10, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 8, 'n_estimators': 300}	50%	50%	50%	49%	 <p>The first plot, titled "Training and Validation Accuracy for k=3", shows accuracy on the y-axis (0.65 to 0.90) and epochs on the x-axis (1.0 to 4.0). It features two horizontal lines: a blue line for "Training Accuracy" at approximately 0.91 and an orange line for "Validation Accuracy" at approximately 0.65. The second plot, titled "Training and Validation Error for k=3", shows error on the y-axis (0.55 to 0.95) and epochs on the x-axis (1.0 to 4.0). It also features two horizontal lines: a blue line for "Training Error" at approximately 0.55 and an orange line for "Validation Error" at approximately 0.95.</p>
TFIDF k= 3	Best parameters found: { 'bootstrap': False, 'max_depth': 10, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 6, 'n_estimators': 300}	71%	72%	71%	71%	 <p>The first plot, titled "Training and Validation Accuracy for k=3", shows accuracy on the y-axis (0.70 to 0.95) and epochs on the x-axis (1.0 to 4.0). It features two horizontal lines: a blue line for "Training Accuracy" at approximately 0.95 and an orange line for "Validation Accuracy" at approximately 0.70. The second plot, titled "Training and Validation Error for k=3", shows error on the y-axis (0.7 to 1.1) and epochs on the x-axis (1.0 to 4.0). It features two horizontal lines: a blue line for "Training Error" at approximately 0.75 and an orange line for "Validation Error" at approximately 1.05.</p>

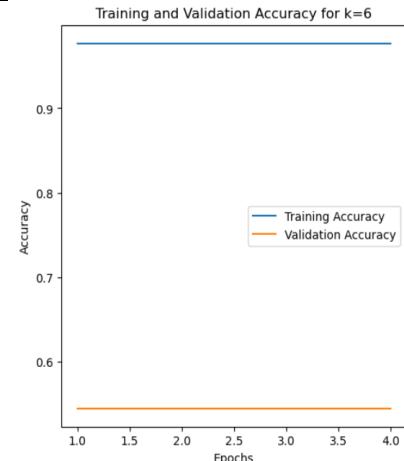
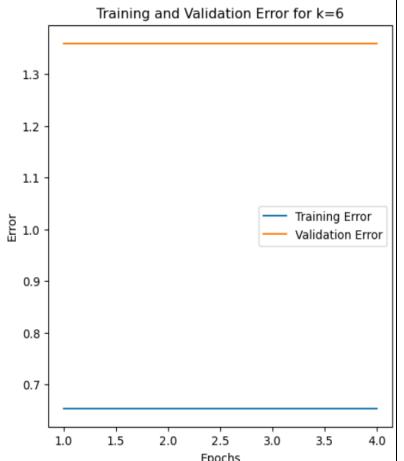
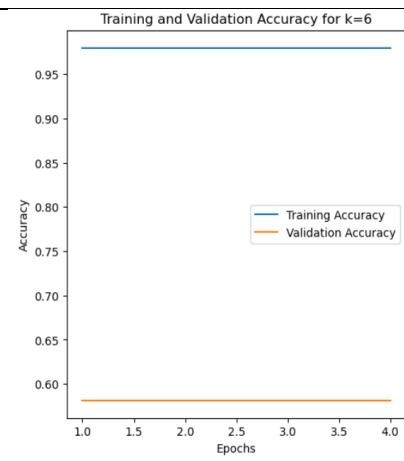
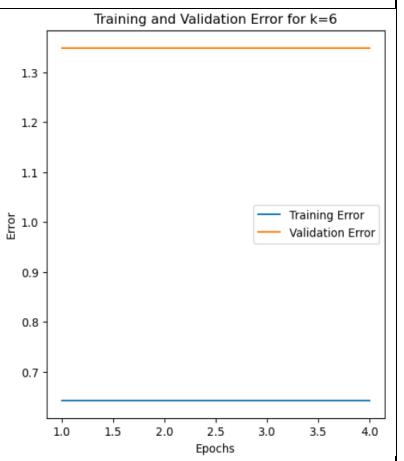
Caractérisation d'éléments transposables : les transposons à TIR

CV k=4	Best parameters found: <pre>{'bootstrap': True, 'max_depth': 10, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 8, 'n_estimators': 300}</pre>	54%	55%	54%	54%	 <p>Training and Validation Accuracy for k=4</p> <table border="1"> <thead> <tr> <th>Epochs</th> <th>Training Accuracy</th> <th>Validation Accuracy</th> </tr> </thead> <tbody> <tr><td>1.0</td><td>~0.90</td><td>~0.50</td></tr> <tr><td>2.0</td><td>~0.90</td><td>~0.50</td></tr> <tr><td>3.0</td><td>~0.90</td><td>~0.50</td></tr> <tr><td>4.0</td><td>~0.90</td><td>~0.50</td></tr> </tbody> </table>	Epochs	Training Accuracy	Validation Accuracy	1.0	~0.90	~0.50	2.0	~0.90	~0.50	3.0	~0.90	~0.50	4.0	~0.90	~0.50	 <p>Training and Validation Error for k=4</p> <table border="1"> <thead> <tr> <th>Epochs</th> <th>Training Error</th> <th>Validation Error</th> </tr> </thead> <tbody> <tr><td>1.0</td><td>~1.00</td><td>~1.35</td></tr> <tr><td>2.0</td><td>~1.00</td><td>~1.35</td></tr> <tr><td>3.0</td><td>~1.00</td><td>~1.35</td></tr> <tr><td>4.0</td><td>~1.00</td><td>~1.35</td></tr> </tbody> </table>	Epochs	Training Error	Validation Error	1.0	~1.00	~1.35	2.0	~1.00	~1.35	3.0	~1.00	~1.35	4.0	~1.00	~1.35
Epochs	Training Accuracy	Validation Accuracy																																			
1.0	~0.90	~0.50																																			
2.0	~0.90	~0.50																																			
3.0	~0.90	~0.50																																			
4.0	~0.90	~0.50																																			
Epochs	Training Error	Validation Error																																			
1.0	~1.00	~1.35																																			
2.0	~1.00	~1.35																																			
3.0	~1.00	~1.35																																			
4.0	~1.00	~1.35																																			
TFiDF k=4	Best parameters found: <pre>{'bootstrap': False, 'max_depth': 10, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 10, 'n_estimators': 500}</pre>	45%	51%	44%	40%	 <p>Training and Validation Accuracy for k=4</p> <table border="1"> <thead> <tr> <th>Epochs</th> <th>Training Accuracy</th> <th>Validation Accuracy</th> </tr> </thead> <tbody> <tr><td>1.0</td><td>~0.95</td><td>~0.65</td></tr> <tr><td>2.0</td><td>~0.95</td><td>~0.65</td></tr> <tr><td>3.0</td><td>~0.95</td><td>~0.65</td></tr> <tr><td>4.0</td><td>~0.95</td><td>~0.65</td></tr> </tbody> </table>	Epochs	Training Accuracy	Validation Accuracy	1.0	~0.95	~0.65	2.0	~0.95	~0.65	3.0	~0.95	~0.65	4.0	~0.95	~0.65	 <p>Training and Validation Error for k=4</p> <table border="1"> <thead> <tr> <th>Epochs</th> <th>Training Error</th> <th>Validation Error</th> </tr> </thead> <tbody> <tr><td>1.0</td><td>~1.00</td><td>~1.40</td></tr> <tr><td>2.0</td><td>~1.00</td><td>~1.40</td></tr> <tr><td>3.0</td><td>~1.00</td><td>~1.40</td></tr> <tr><td>4.0</td><td>~1.00</td><td>~1.40</td></tr> </tbody> </table>	Epochs	Training Error	Validation Error	1.0	~1.00	~1.40	2.0	~1.00	~1.40	3.0	~1.00	~1.40	4.0	~1.00	~1.40
Epochs	Training Accuracy	Validation Accuracy																																			
1.0	~0.95	~0.65																																			
2.0	~0.95	~0.65																																			
3.0	~0.95	~0.65																																			
4.0	~0.95	~0.65																																			
Epochs	Training Error	Validation Error																																			
1.0	~1.00	~1.40																																			
2.0	~1.00	~1.40																																			
3.0	~1.00	~1.40																																			
4.0	~1.00	~1.40																																			

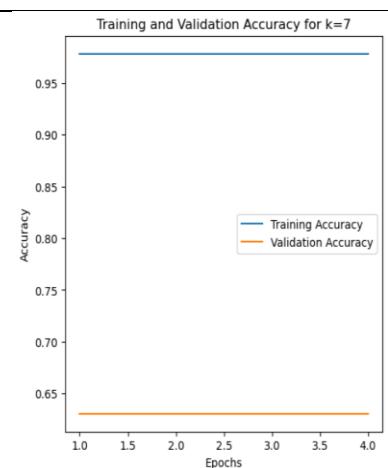
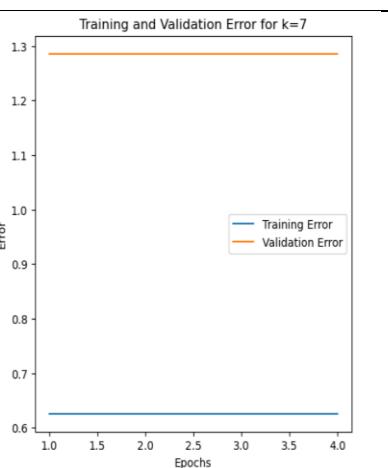
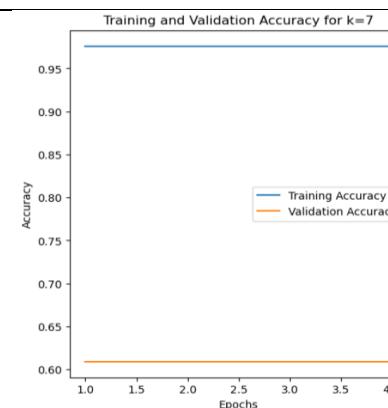
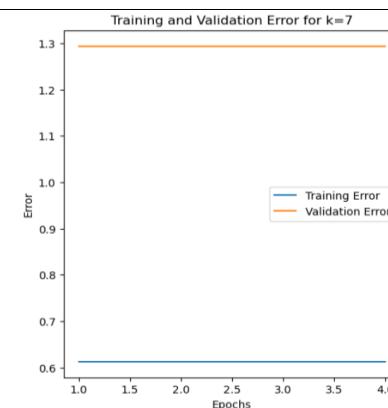
Caractérisation d'éléments transposables : les transposons à TIR

CV k=5	Best parameters found: <pre>{'bootstrap': False, 'max_depth': 10, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 6, 'n_estimators': 500}</pre>	54%	55%	54%	53%		
TFIDF k=5	Best parameters found: <pre>{'bootstrap': False, 'max_depth': 10, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 6, 'n_estimators': 500}</pre>	54%	57%	54%	53%		

Caractérisation d'éléments transposables : les transposons à TIR

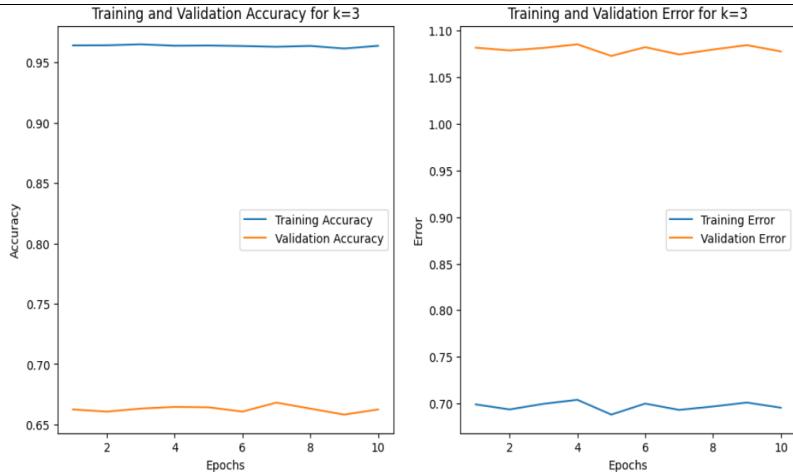
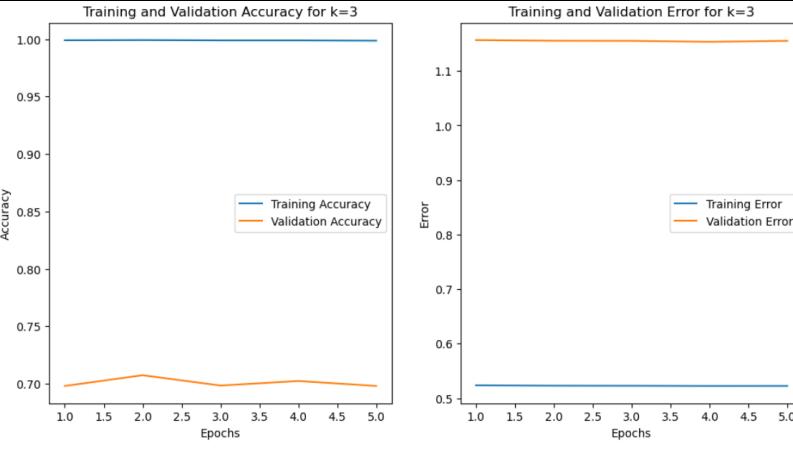
CV k=6	Best parameters found: {'bootstrap': False, 'max_depth': 10, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 6, 'n_estimators': 100}	45%	50%	46%	45%	 Training and Validation Accuracy for k=6 Accuracy vs Epochs	 Training and Validation Error for k=6 Error vs Epochs
TFIDF k=6	Best parameters found: {'bootstrap': False, 'max_depth': 10, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 6, 'n_estimators': 500}	48%	55%	48%	46%	 Training and Validation Accuracy for k=6 Accuracy vs Epochs	 Training and Validation Error for k=6 Error vs Epochs

Caractérisation d'éléments transposables : les transposons à TIR

CV k=7	Best parameters found: {'bootstrap': False, 'max_depth': 10, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 6, 'n_estimators': 500}	59%	61%	59%	59%		
TFIDF k=7	Best parameters found: {'bootstrap': False, 'max_depth': 10, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 6, 'n_estimators': 500}	56,5%	63%	56,1%	57,1%		

c.1.2. Extra Trees

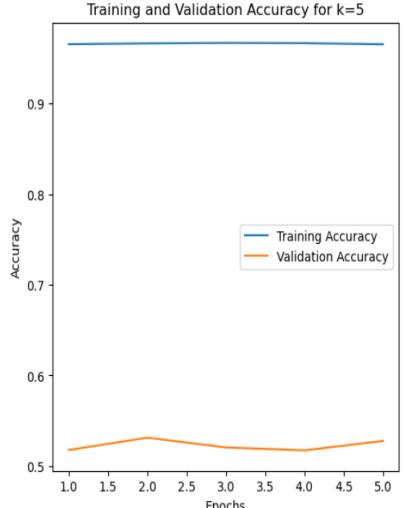
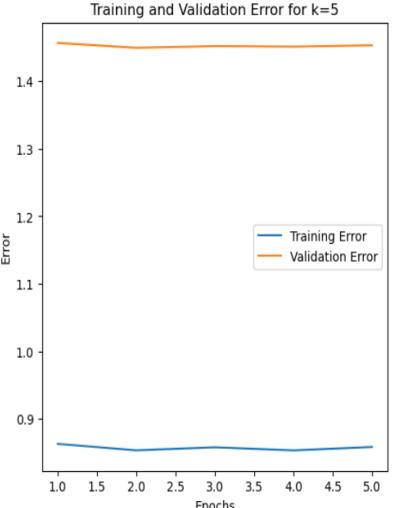
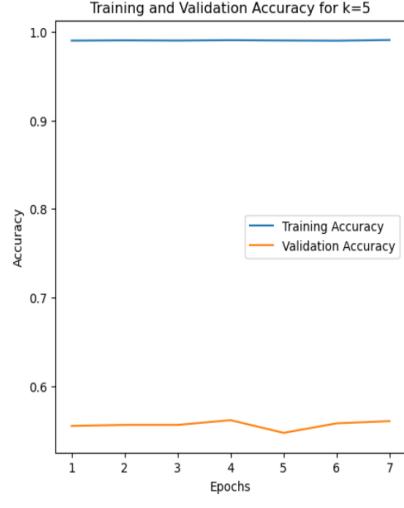
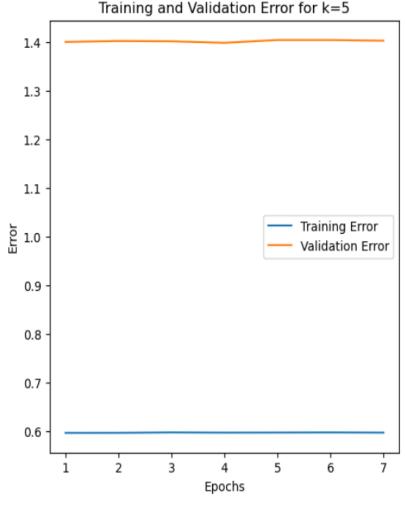
Tableau 10 : résultat CM/ML : modèle Extra Trees

Feature Extraction	Grid Search	Accuracy	Precision	Recall	F1-Score	plot																																																																		
CV k=3	Best parameters found: {'bootstrap': True, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 4, 'n_estimators': 200}	54%	54%	53%	53%	 <p>Training and Validation Accuracy for k=3</p> <table border="1"> <thead> <tr> <th>Epochs</th> <th>Training Accuracy</th> <th>Validation Accuracy</th> </tr> </thead> <tbody> <tr><td>1</td><td>~0.96</td><td>~0.66</td></tr> <tr><td>2</td><td>~0.96</td><td>~0.66</td></tr> <tr><td>3</td><td>~0.96</td><td>~0.66</td></tr> <tr><td>4</td><td>~0.96</td><td>~0.66</td></tr> <tr><td>5</td><td>~0.96</td><td>~0.66</td></tr> <tr><td>6</td><td>~0.96</td><td>~0.66</td></tr> <tr><td>7</td><td>~0.96</td><td>~0.66</td></tr> <tr><td>8</td><td>~0.96</td><td>~0.66</td></tr> <tr><td>9</td><td>~0.96</td><td>~0.66</td></tr> <tr><td>10</td><td>~0.96</td><td>~0.66</td></tr> </tbody> </table> <p>Training and Validation Error for k=3</p> <table border="1"> <thead> <tr> <th>Epochs</th> <th>Training Error</th> <th>Validation Error</th> </tr> </thead> <tbody> <tr><td>1</td><td>~1.08</td><td>~1.08</td></tr> <tr><td>2</td><td>~1.08</td><td>~1.08</td></tr> <tr><td>3</td><td>~1.08</td><td>~1.08</td></tr> <tr><td>4</td><td>~1.08</td><td>~1.08</td></tr> <tr><td>5</td><td>~1.08</td><td>~1.08</td></tr> <tr><td>6</td><td>~1.08</td><td>~1.08</td></tr> <tr><td>7</td><td>~1.08</td><td>~1.08</td></tr> <tr><td>8</td><td>~1.08</td><td>~1.08</td></tr> <tr><td>9</td><td>~1.08</td><td>~1.08</td></tr> <tr><td>10</td><td>~1.08</td><td>~1.08</td></tr> </tbody> </table>	Epochs	Training Accuracy	Validation Accuracy	1	~0.96	~0.66	2	~0.96	~0.66	3	~0.96	~0.66	4	~0.96	~0.66	5	~0.96	~0.66	6	~0.96	~0.66	7	~0.96	~0.66	8	~0.96	~0.66	9	~0.96	~0.66	10	~0.96	~0.66	Epochs	Training Error	Validation Error	1	~1.08	~1.08	2	~1.08	~1.08	3	~1.08	~1.08	4	~1.08	~1.08	5	~1.08	~1.08	6	~1.08	~1.08	7	~1.08	~1.08	8	~1.08	~1.08	9	~1.08	~1.08	10	~1.08	~1.08
Epochs	Training Accuracy	Validation Accuracy																																																																						
1	~0.96	~0.66																																																																						
2	~0.96	~0.66																																																																						
3	~0.96	~0.66																																																																						
4	~0.96	~0.66																																																																						
5	~0.96	~0.66																																																																						
6	~0.96	~0.66																																																																						
7	~0.96	~0.66																																																																						
8	~0.96	~0.66																																																																						
9	~0.96	~0.66																																																																						
10	~0.96	~0.66																																																																						
Epochs	Training Error	Validation Error																																																																						
1	~1.08	~1.08																																																																						
2	~1.08	~1.08																																																																						
3	~1.08	~1.08																																																																						
4	~1.08	~1.08																																																																						
5	~1.08	~1.08																																																																						
6	~1.08	~1.08																																																																						
7	~1.08	~1.08																																																																						
8	~1.08	~1.08																																																																						
9	~1.08	~1.08																																																																						
10	~1.08	~1.08																																																																						
TFIDF k = 3	Best parameters found: {'bootstrap': True, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 4, 'n_estimators': 200}	70%	70%	70%	70%	 <p>Training and Validation Accuracy for k=3</p> <table border="1"> <thead> <tr> <th>Epochs</th> <th>Training Accuracy</th> <th>Validation Accuracy</th> </tr> </thead> <tbody> <tr><td>1.0</td><td>1.00</td><td>0.70</td></tr> <tr><td>2.0</td><td>1.00</td><td>0.70</td></tr> <tr><td>3.0</td><td>1.00</td><td>0.70</td></tr> <tr><td>4.0</td><td>1.00</td><td>0.70</td></tr> <tr><td>5.0</td><td>1.00</td><td>0.70</td></tr> </tbody> </table> <p>Training and Validation Error for k=3</p> <table border="1"> <thead> <tr> <th>Epochs</th> <th>Training Error</th> <th>Validation Error</th> </tr> </thead> <tbody> <tr><td>1.0</td><td>0.52</td><td>1.10</td></tr> <tr><td>2.0</td><td>0.52</td><td>1.10</td></tr> <tr><td>3.0</td><td>0.52</td><td>1.10</td></tr> <tr><td>4.0</td><td>0.52</td><td>1.10</td></tr> <tr><td>5.0</td><td>0.52</td><td>1.10</td></tr> </tbody> </table>	Epochs	Training Accuracy	Validation Accuracy	1.0	1.00	0.70	2.0	1.00	0.70	3.0	1.00	0.70	4.0	1.00	0.70	5.0	1.00	0.70	Epochs	Training Error	Validation Error	1.0	0.52	1.10	2.0	0.52	1.10	3.0	0.52	1.10	4.0	0.52	1.10	5.0	0.52	1.10																														
Epochs	Training Accuracy	Validation Accuracy																																																																						
1.0	1.00	0.70																																																																						
2.0	1.00	0.70																																																																						
3.0	1.00	0.70																																																																						
4.0	1.00	0.70																																																																						
5.0	1.00	0.70																																																																						
Epochs	Training Error	Validation Error																																																																						
1.0	0.52	1.10																																																																						
2.0	0.52	1.10																																																																						
3.0	0.52	1.10																																																																						
4.0	0.52	1.10																																																																						
5.0	0.52	1.10																																																																						

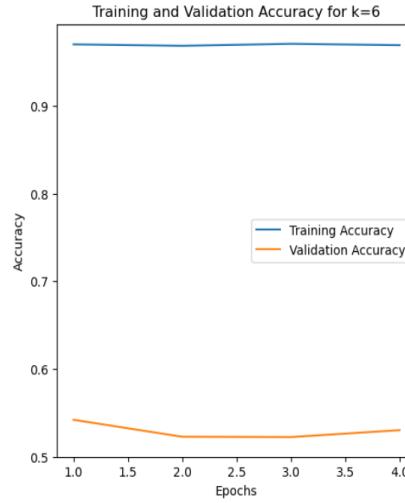
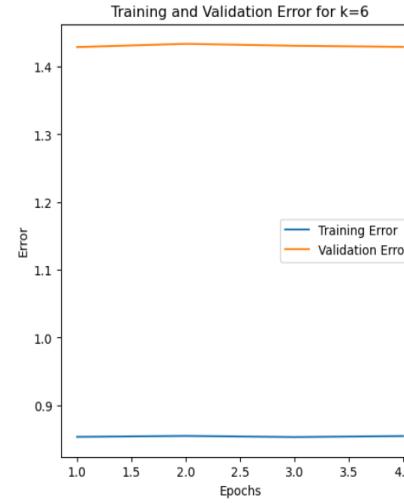
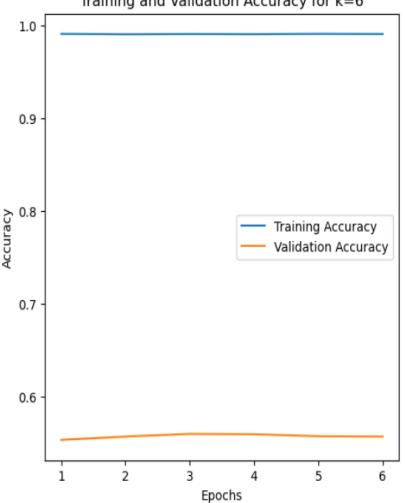
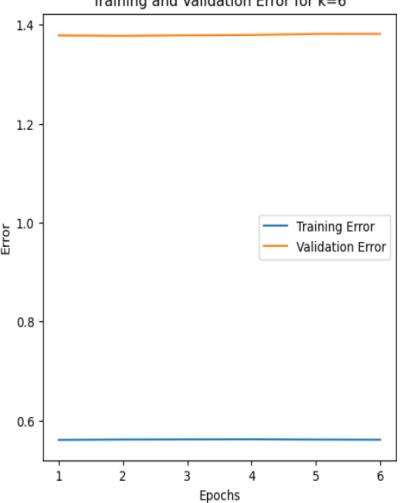
Caractérisation d'éléments transposables : les transposons à TIR

CV k=4	Best parameters found: {'bootstrap': True, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 4, 'n_estimators': 200}	54%	54%	53%	52%	<p>Training and Validation Accuracy for k=4</p> <table border="1"> <thead> <tr> <th>Epochs</th> <th>Training Accuracy</th> <th>Validation Accuracy</th> </tr> </thead> <tbody> <tr><td>1.0</td><td>~0.95</td><td>~0.52</td></tr> <tr><td>1.5</td><td>~0.95</td><td>~0.52</td></tr> <tr><td>2.0</td><td>~0.95</td><td>~0.53</td></tr> <tr><td>2.5</td><td>~0.95</td><td>~0.52</td></tr> <tr><td>3.0</td><td>~0.95</td><td>~0.52</td></tr> <tr><td>3.5</td><td>~0.95</td><td>~0.52</td></tr> <tr><td>4.0</td><td>~0.95</td><td>~0.52</td></tr> <tr><td>4.5</td><td>~0.95</td><td>~0.51</td></tr> <tr><td>5.0</td><td>~0.95</td><td>~0.51</td></tr> </tbody> </table>	Epochs	Training Accuracy	Validation Accuracy	1.0	~0.95	~0.52	1.5	~0.95	~0.52	2.0	~0.95	~0.53	2.5	~0.95	~0.52	3.0	~0.95	~0.52	3.5	~0.95	~0.52	4.0	~0.95	~0.52	4.5	~0.95	~0.51	5.0	~0.95	~0.51	<p>Training and Validation Error for k=4</p> <table border="1"> <thead> <tr> <th>Epochs</th> <th>Training Error</th> <th>Validation Error</th> </tr> </thead> <tbody> <tr><td>1.0</td><td>~0.90</td><td>~1.45</td></tr> <tr><td>1.5</td><td>~0.88</td><td>~1.44</td></tr> <tr><td>2.0</td><td>~0.87</td><td>~1.43</td></tr> <tr><td>2.5</td><td>~0.86</td><td>~1.42</td></tr> <tr><td>3.0</td><td>~0.87</td><td>~1.43</td></tr> <tr><td>3.5</td><td>~0.86</td><td>~1.42</td></tr> <tr><td>4.0</td><td>~0.87</td><td>~1.43</td></tr> <tr><td>4.5</td><td>~0.86</td><td>~1.42</td></tr> <tr><td>5.0</td><td>~0.86</td><td>~1.41</td></tr> </tbody> </table>	Epochs	Training Error	Validation Error	1.0	~0.90	~1.45	1.5	~0.88	~1.44	2.0	~0.87	~1.43	2.5	~0.86	~1.42	3.0	~0.87	~1.43	3.5	~0.86	~1.42	4.0	~0.87	~1.43	4.5	~0.86	~1.42	5.0	~0.86	~1.41
Epochs	Training Accuracy	Validation Accuracy																																																																	
1.0	~0.95	~0.52																																																																	
1.5	~0.95	~0.52																																																																	
2.0	~0.95	~0.53																																																																	
2.5	~0.95	~0.52																																																																	
3.0	~0.95	~0.52																																																																	
3.5	~0.95	~0.52																																																																	
4.0	~0.95	~0.52																																																																	
4.5	~0.95	~0.51																																																																	
5.0	~0.95	~0.51																																																																	
Epochs	Training Error	Validation Error																																																																	
1.0	~0.90	~1.45																																																																	
1.5	~0.88	~1.44																																																																	
2.0	~0.87	~1.43																																																																	
2.5	~0.86	~1.42																																																																	
3.0	~0.87	~1.43																																																																	
3.5	~0.86	~1.42																																																																	
4.0	~0.87	~1.43																																																																	
4.5	~0.86	~1.42																																																																	
5.0	~0.86	~1.41																																																																	
TFIDF k=4	Best parameters found: {'bootstrap': True, 'max_features': 'sqrt', 'min_samples_leaf': 3, 'min_samples_split': 8, 'n_estimators': 200}	47%	50%	47%	45%	<p>Training and Validation Accuracy for k=4</p> <table border="1"> <thead> <tr> <th>Epochs</th> <th>Training Accuracy</th> <th>Validation Accuracy</th> </tr> </thead> <tbody> <tr><td>1</td><td>~0.99</td><td>~0.62</td></tr> <tr><td>2</td><td>~0.99</td><td>~0.63</td></tr> <tr><td>3</td><td>~0.99</td><td>~0.61</td></tr> <tr><td>4</td><td>~0.99</td><td>~0.63</td></tr> <tr><td>5</td><td>~0.99</td><td>~0.64</td></tr> <tr><td>6</td><td>~0.99</td><td>~0.62</td></tr> <tr><td>7</td><td>~0.99</td><td>~0.61</td></tr> <tr><td>8</td><td>~0.99</td><td>~0.61</td></tr> </tbody> </table>	Epochs	Training Accuracy	Validation Accuracy	1	~0.99	~0.62	2	~0.99	~0.63	3	~0.99	~0.61	4	~0.99	~0.63	5	~0.99	~0.64	6	~0.99	~0.62	7	~0.99	~0.61	8	~0.99	~0.61	<p>Training and Validation Error for k=4</p> <table border="1"> <thead> <tr> <th>Epochs</th> <th>Training Error</th> <th>Validation Error</th> </tr> </thead> <tbody> <tr><td>1</td><td>~0.90</td><td>~1.45</td></tr> <tr><td>2</td><td>~0.89</td><td>~1.44</td></tr> <tr><td>3</td><td>~0.88</td><td>~1.43</td></tr> <tr><td>4</td><td>~0.88</td><td>~1.42</td></tr> <tr><td>5</td><td>~0.88</td><td>~1.41</td></tr> <tr><td>6</td><td>~0.88</td><td>~1.42</td></tr> <tr><td>7</td><td>~0.88</td><td>~1.41</td></tr> <tr><td>8</td><td>~0.88</td><td>~1.41</td></tr> </tbody> </table>	Epochs	Training Error	Validation Error	1	~0.90	~1.45	2	~0.89	~1.44	3	~0.88	~1.43	4	~0.88	~1.42	5	~0.88	~1.41	6	~0.88	~1.42	7	~0.88	~1.41	8	~0.88	~1.41						
Epochs	Training Accuracy	Validation Accuracy																																																																	
1	~0.99	~0.62																																																																	
2	~0.99	~0.63																																																																	
3	~0.99	~0.61																																																																	
4	~0.99	~0.63																																																																	
5	~0.99	~0.64																																																																	
6	~0.99	~0.62																																																																	
7	~0.99	~0.61																																																																	
8	~0.99	~0.61																																																																	
Epochs	Training Error	Validation Error																																																																	
1	~0.90	~1.45																																																																	
2	~0.89	~1.44																																																																	
3	~0.88	~1.43																																																																	
4	~0.88	~1.42																																																																	
5	~0.88	~1.41																																																																	
6	~0.88	~1.42																																																																	
7	~0.88	~1.41																																																																	
8	~0.88	~1.41																																																																	

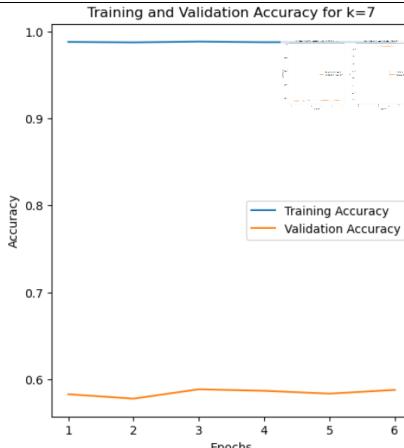
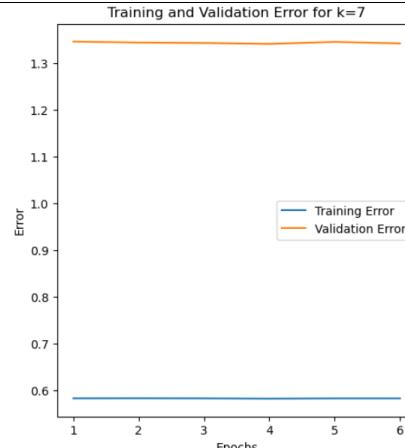
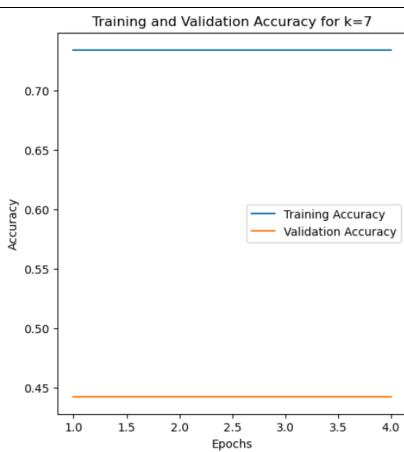
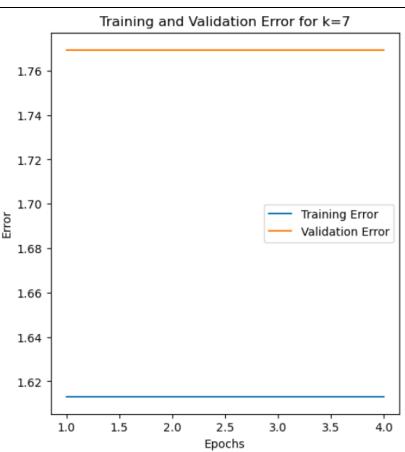
Caractérisation d'éléments transposables : les transposons à TIR

CV k=5	Best parameters found: <pre>{'bootstrap': True, 'max_features': 'log2', 'min_samples_leaf': 2, 'min_samples_split': 4, 'n_estimators': 200}</pre>	51%	52%	51%	48%		
TFIDF k=5	Best parameters found: <pre>{'bootstrap': True, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 4, 'n_estimators': 200}</pre>	51%	53%	51%	49%		

Caractérisation d'éléments transposables : les transposons à TIR

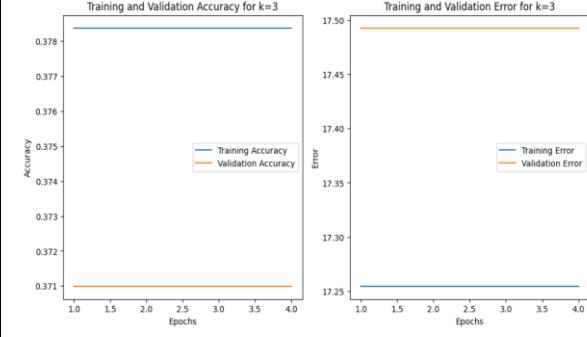
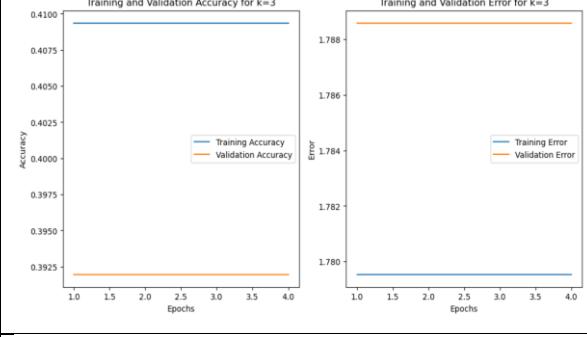
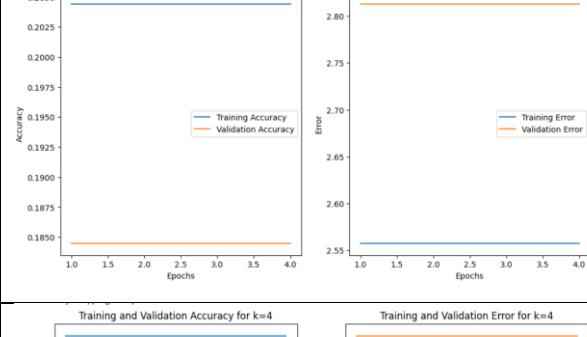
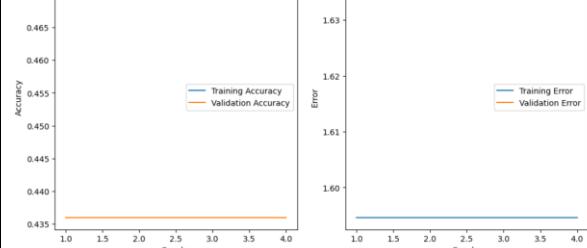
CV k=6	Best parameters found: {'bootstrap': True, 'max_features': 'sqrt', 'min_samples_leaf': 3, 'min_samples_split': 4, 'n_estimators': 200}	47%	54%	48%	45%		
TFIDF k=6	Best parameters found: {'bootstrap': True, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 4, 'n_estimators': 200}	47%	51%	47%	45%		

Caractérisation d'éléments transposables : les transposons à TIR

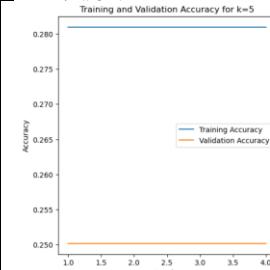
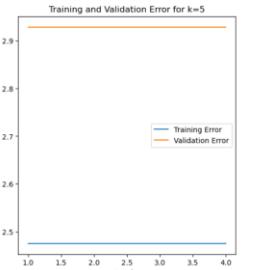
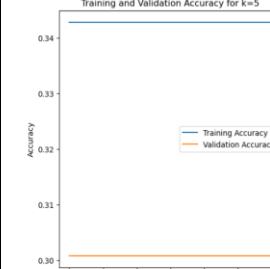
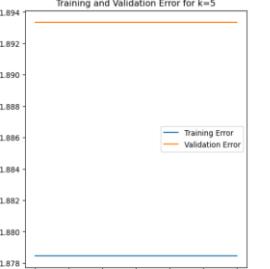
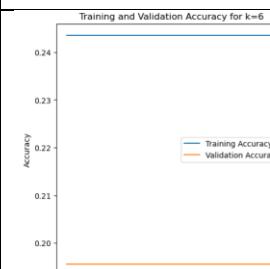
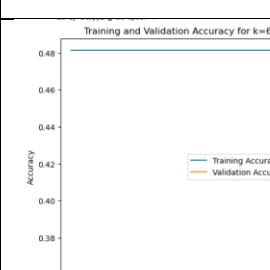
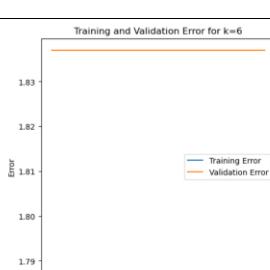
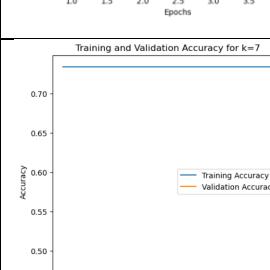
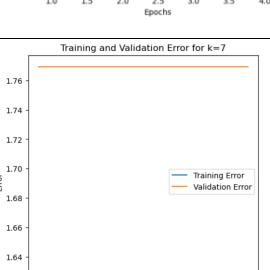
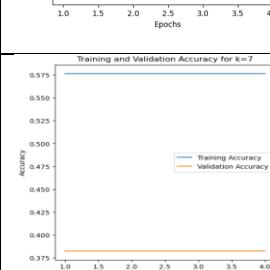
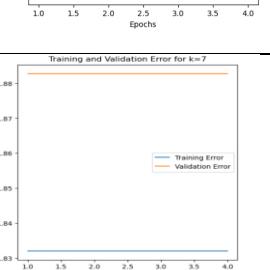
CV k=7	Best parameters found: {'bootstrap': True, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 4, 'n_estimators': 200}	55%	55,9%	56,1%	54,2%		
TFIDF k=7	Best parameters found: {'bootstrap': True, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 4, 'n_estimators': 200}	43%	47%	43%	44%		

c.1.3. Naïve Bayes

Tableau 11 : résultat CM/ML : modèle Naive Bayes

Feature Extraction	Grid Search	Accuracy	Precision	Recall	F1-Score	plot
CV k=3	Best parameters found: {'alpha': 10.0}	36%	34%	36%	34%	
TFIDF k=3	Best parameters found: {'alpha': 0.0001}	37%	37%	37%	35%	
CV k=4	Best parameters found: {'alpha': 0.001}	36%	35%	35%	34%	
TFIDF k=4	Best parameters found: {'alpha': 0.0001}	36%	42%	38%	37%	

Caractérisation d'éléments transposables : les transposons à TIR

CV k=5	Best parameters found: {'alpha': 0.01}	36%	37%	36%	35%	 
TFiDF k=5	Best parameters found: {'alpha': 1.0}	27%	27%	27%	18%	 
CV k=6	Best parameters found: {'alpha': 0.1}	35%	37 %	35%	34%	 
TFiDF k=6	Best parameters found: {'alpha': 0.1}	0.27	0.39	0.28	0.21	 
CV k=7	Best parameters found: {'alpha': 0.0001}	43%	47%	43%	44%	 
TFiDF k=7	Best parameters found: {'alpha': 0.0001}	37,5%	41%	37,4%	38%	 

c.2. DL

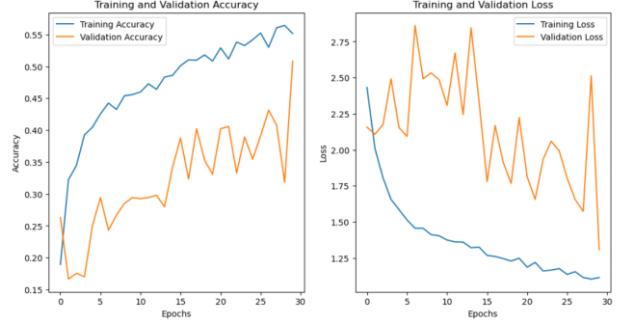
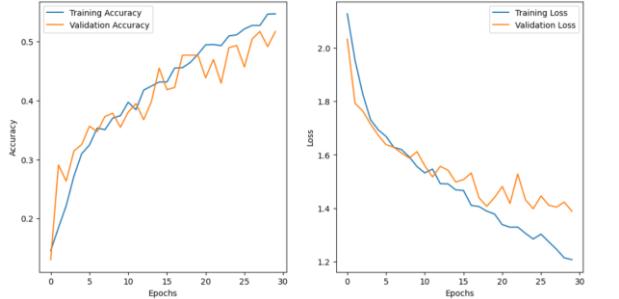
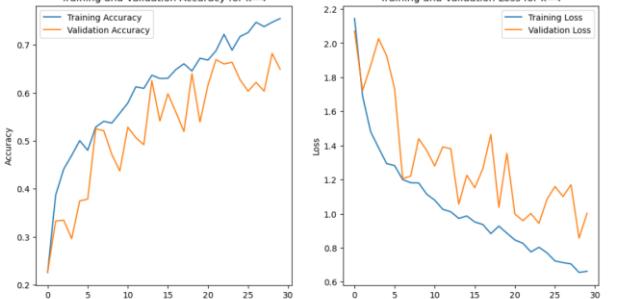
c.2.1. Bert

Tableau 12 : résultat CM/DL : modèle Bert

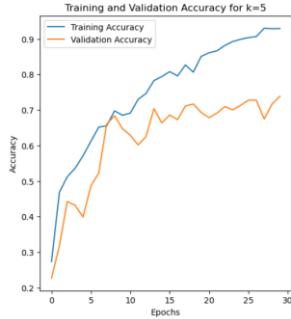
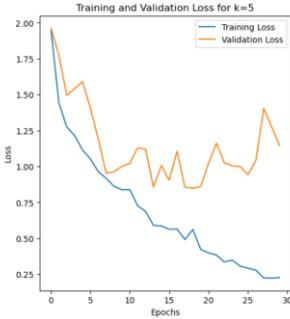
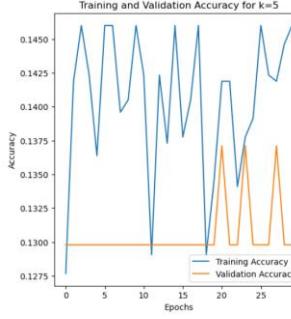
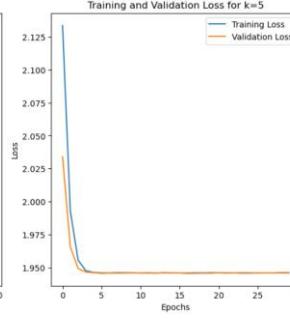
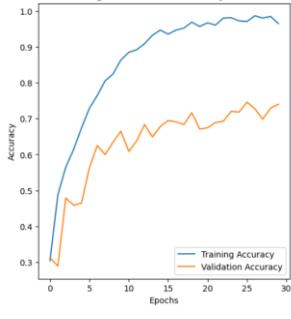
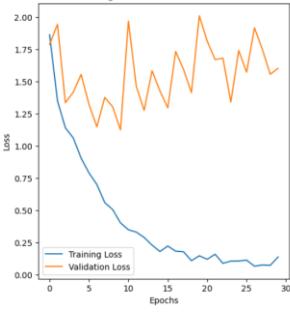
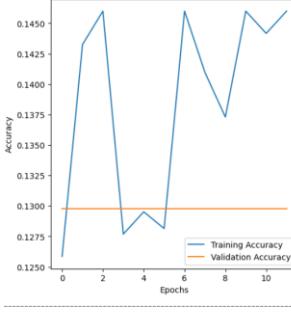
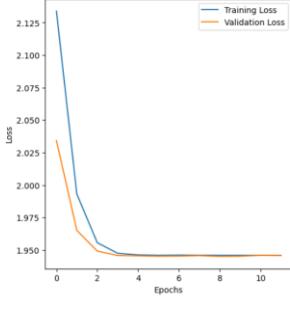
Feature Extraction	accuracy	Precision	Recall	F1-Score	plot
K=3	18%	3%	14%	4%	
K=4	12%	2%	14%	3%	
K=5	12%	2%	14%	3%	
K=6	12%	2%	14%	3%	

c.2.1. CNN

Tableau 13 : résultat CM/DL : modèle CNN

Feature Extraction	accuracy	Precision	Recall	F1-Score	Plot
Cv=3	52%	53%	51%	50%	
Tfidf=3	50%	52%	50%	50%	
Cv=4	64%	64%	63%	62%	
Tfidf=4	61%	62%	60%	61%	

Caractérisation d'éléments transposables : les transposons à TIR

Cv=5	80%	81%	80%	80%	 
Tfidf =5	14%	2%	14%	4%	 
CV k=6	71%	71%	71%	71%	 
Tfidf = 6	14%	2%	14%	4%	 

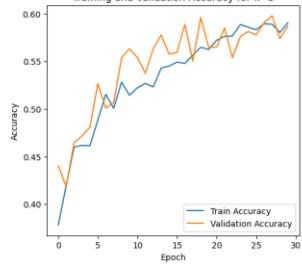
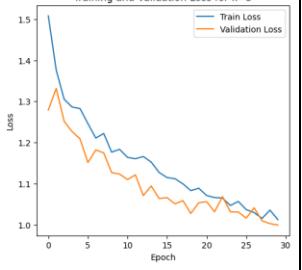
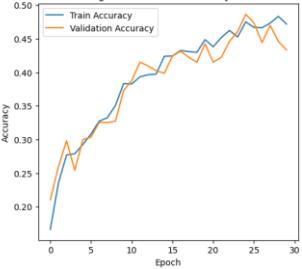
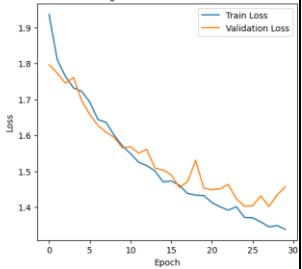
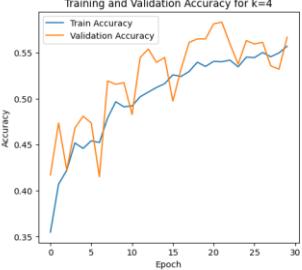
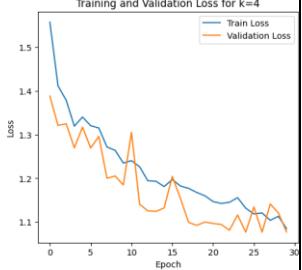
c.2.1. CNN et BiLSTM

Layer (type)
conv1d_8 (Conv1D)
max_pooling1d_7 (MaxPooling1D)
bidirectional_8 (Bidirectional)
dropout_8 (Dropout)
dense_18 (Dense)
dense_19 (Dense)
dense_20 (Dense)

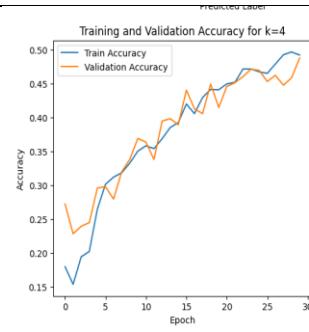
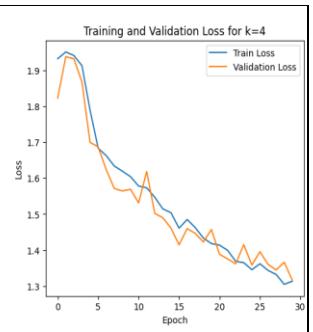
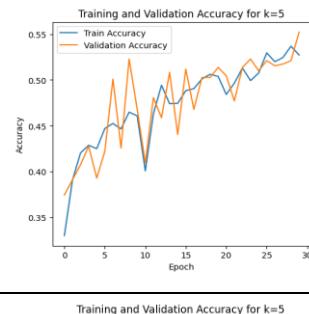
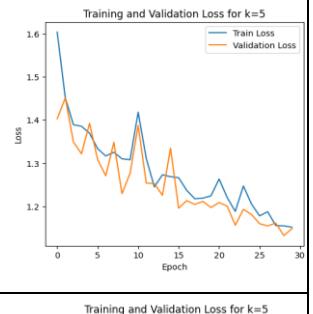
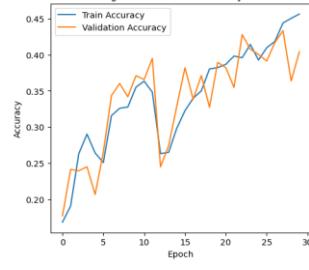
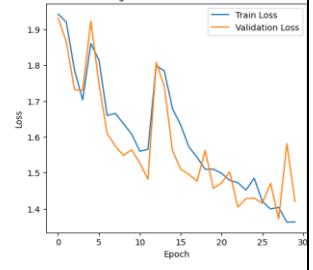
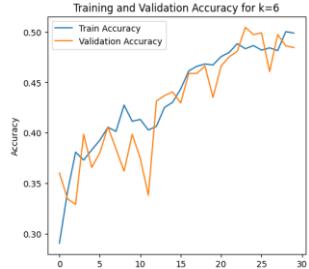
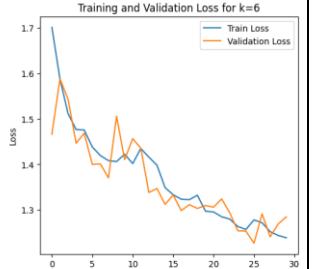
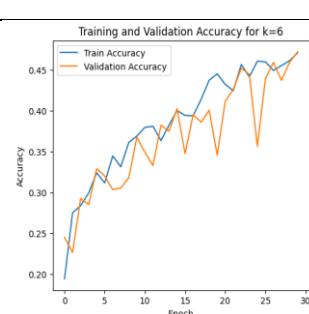
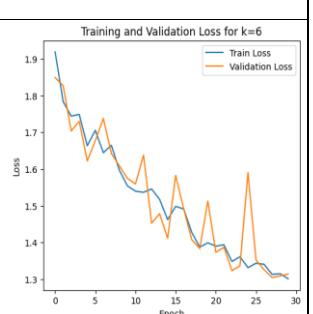
Figure 45 : CNN& BiLSTM architecture

Cette image indique le l'architecture du modèle : le nombre des couche CNN et bidirectionnel LSTM.

Tableau 14 : résultat CM/DL : modèle CNN & BiLSTM

Feature Extraction	accuracy	Precision	Recall	F1-Score	Plot
Cv k=3	59%	60%	59%	59%	 
Tfidf k=3	43%	46%	43%	41%	 
Cv k=4	56%	56%	55%	56%	 

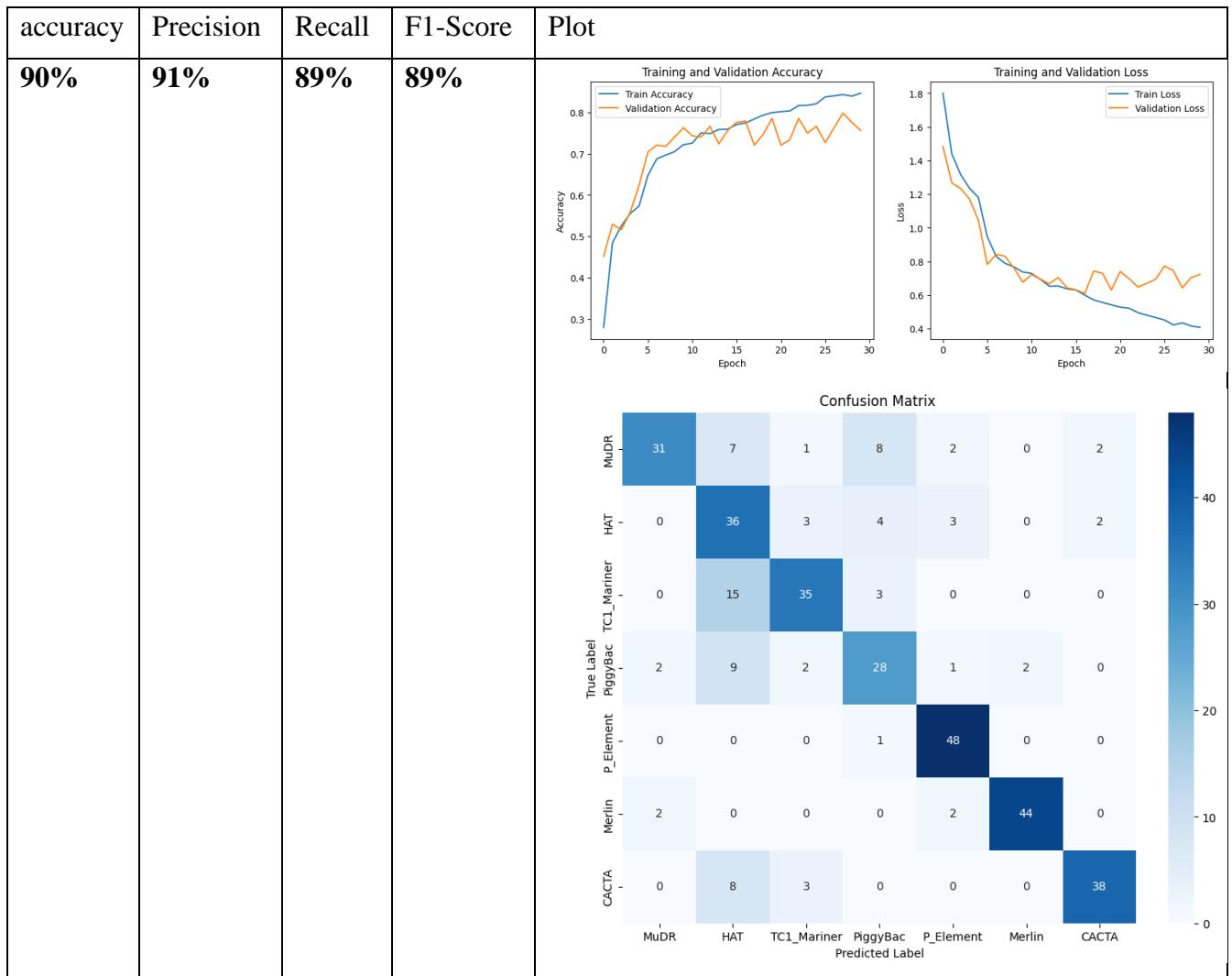
Caractérisation d'éléments transposables : les transposons à TIR

Tfidf k=4	49%	49%	49%	49%	 
Cv k=5	54%	56%	53%	53%	 
Tfidf k=5	39%	40%	40%	34%	 
CV k=6	49%	48%	49%	47%	 
Tfidf k=6	46%	48%	46%	44%	 

c.2.2. CNN et GRU

Dans ce modèle nous avons utilisé One Hot Encoding comme vectoriser pour {'a': [1, 0, 0, 0], 't': [0, 1, 0, 0], 'c': [0, 0, 1, 0], 'g': [0, 0, 0, 1]}

Tableau 15 : résultat CM/DL : modèle CNN & GRU



c.3. Discutions des résultats

Les résultats illustraient montrent les performances de plusieurs modèles de classification basés sur différents paramètres d'extraction de caractéristiques, ainsi que leurs métriques associées telles que l'exactitude (accuracy), la précision, le rappel, et le F1-score. Voici un résumé des résultats pour chaque modèle testé :

1. BERT :

- Les résultats sont faibles avec des performances d'exactitude très basses (autour de 12 à 18%), ce qui indique que ce modèle n'a pas bien fonctionné pour les k-mers testés (k=3 à k=6).
- Le F1-score, la précision, et le rappel sont tous également très faibles, autour de 2 à 14%.
- Le modèle Bert utilisé est DNABERT, utilisé DNABERT, un modèle pré-entraîné basé sur l'architecture BERT, spécialement adapté pour le traitement des séquences d'ADN. Ce modèle a été initialement pré-entraîné sur des séquences du génome humain en utilisant des k-mers. Nous avons ensuite effectué un fine-tuning sur ce modèle en utilisant nos propres données spécifiques pour améliorer ses performances sur les tâches de classification de séquences.

2. CNN :

- Ce modèle présente des résultats variables selon les paramètres.
- Pour CV k=5, l'exactitude atteint 80% avec un F1-score correspondant de 80%, ce qui indique une très bonne performance.
- Cependant, avec TF-IDF k=5 ou k=6, les performances chutent considérablement, avec une exactitude autour de 14% et un F1-score très bas.

3. CNN & BiLSTM :

- Les performances varient en fonction de l'extraction de caractéristiques (k-mers) utilisée.
- Par exemple, avec CV k=3, l'exactitude est de 59%, mais avec TF-IDF k=3, elle chute à 43%.
- De manière générale, ce modèle montre une variabilité dans les résultats avec des exactitudes variant entre 39% et 59% en fonction de la méthode d'extraction.

4. CNN & GRU :

Ce modèle montre une meilleure performance avec une exactitude de **90%**.

- Les autres métriques, telles que la précision, le rappel et le F1-score, sont toutes autour de 89% à 92%, ce qui indique un bon équilibre entre les prédictions correctes et les erreurs.

En résumé, les performances varient considérablement selon le modèle utilisé et la méthode d'extraction des caractéristiques. CNN avec GRU ou BiLSTM tend à offrir de meilleures performances que BERT, en particulier avec certains paramètres de k-mers et d'extraction des caractéristiques.

c.4. Résultat final

Nous avons choisi d'utiliser le modèle combinant CNN et GRU en raison de ses performances exceptionnelles démontrées lors de l'évaluation. Ce modèle a atteint une précision de 90%, un score de rappel de 91%, et un F1-Score de 89%, ce qui montre un excellent équilibre entre la précision et la capacité à identifier correctement les classes positives. L'architecture CNN et GRU permet de capturer efficacement les caractéristiques locales à travers les convolutions tout en exploitant la capacité des GRU à modéliser les dépendances séquentielles. Cette combinaison s'est avérée être la plus performante pour mes données, c'est pourquoi nous avons décidé de la retenir pour la suite de notre projet.

9. Conclusion

A travers ce chapitre, nous avons présenté la plus importante étape qui est la collecte des données, maintenant il est possible pour passer à la réalisation du projet.

Chapitre IV : Réalisation du projet

1. Introduction

Ce chapitre est composé d'une présentation de l'ensemble des technologies, des outils et des choix techniques de ce projet.

Ensuite, on poursuit par une illustration des différents processus de fonctionnement du système à travers des captures d'écran commentées.

2. Choix techniques

2.1. Environnement de développement

a. Kaggle



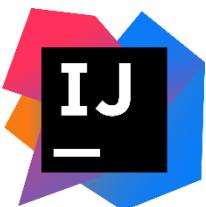
Kaggle est une plateforme web interactive qui propose des compétitions d'apprentissage automatique en science des données. La plateforme fournit des jeux de données, des notebooks et des didacticiels gratuits dont les scientifiques de données ont besoin pour réaliser leurs projets d'apprentissage automatique.

b. Google Colab



Colab un outil permet d'importer un ensemble de données d'images, d'entraîner un classificateur d'images sur cet ensemble et d'évaluer le modèle, tout cela avec quelques lignes de code. Les notebooks Colab exécutent ce code sur les serveurs cloud de Google.

c. IntelliJ IDEA



Un environnement de développement destiné au développement de logiciels informatiques reposant sur la technologie Java. Il a été utilisé pour le développement de l'application mobile.

d. Visual Studio Code



VS Code est présenté lors de la conférence des développeurs Build d'avril 2015 comme un éditeur de code multiplateforme, open source et gratuit, supportant une dizaine de langage. Il a été utilisé pour le développement du cote webservices REST.

2.2. Outils utilisés

i. Langages utilisés

a. Python

Python est un langage de programmation interprété, multiparadigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet.

b. Java pour l'Android

Java est une technique informatique développée initialement par Sun Microsystems puis acquise par Oracle à la suite du rachat de l'entreprise. Défini à l'origine comme un langage de programmation, Java a évolué pour devenir un ensemble cohérent d'éléments techniques et non techniques

ii. Framework et bibliothèques utilisées

a) Numpy

NumPy est une bibliothèque pour langage de programmation Python, destinée à manipuler des matrices ou tableaux multidimensionnels ainsi que des fonctions mathématiques opérant sur ces tableaux.

b) Matplotlib

Matplotlib est une bibliothèque du langage de programmation Python destinée à tracer et visualiser des données sous formes de graphiques.

c) Pandas

Pandas est une bibliothèque écrite pour le langage de programmation Python permettant la manipulation et l'analyse des données. Elle propose en particulier des structures de données et des opérations de manipulation de tableaux numériques et de séries temporelles.

d) SeqIO

Biopython fournit un module, Bio.SeqIO pour lire et écrire des séquences depuis et vers un fichier (n'importe quel flux) respectivement. Il prend en charge presque tous les formats de fichiers disponibles en bio-informatique. La plupart des logiciels offrent une approche différente pour différents formats de fichiers. Mais, Biopython suit consciemment une approche unique pour présenter les données de séquence analysées à l'utilisateur via son objet SeqRecord.

3. Architecture

Le projet est se forme d'un client server, le client est une application mobile réalisé avec java Android et le serveur est un application Flask avec l'API RESTful (les web services) avec le langage Python, les services ce sont les fonctions de prétraitement (tokenisation, vectorisation) et les modèles ML/DL sauvegarder déjà. Les données sont transformées avec des requêtes http en format JSON.

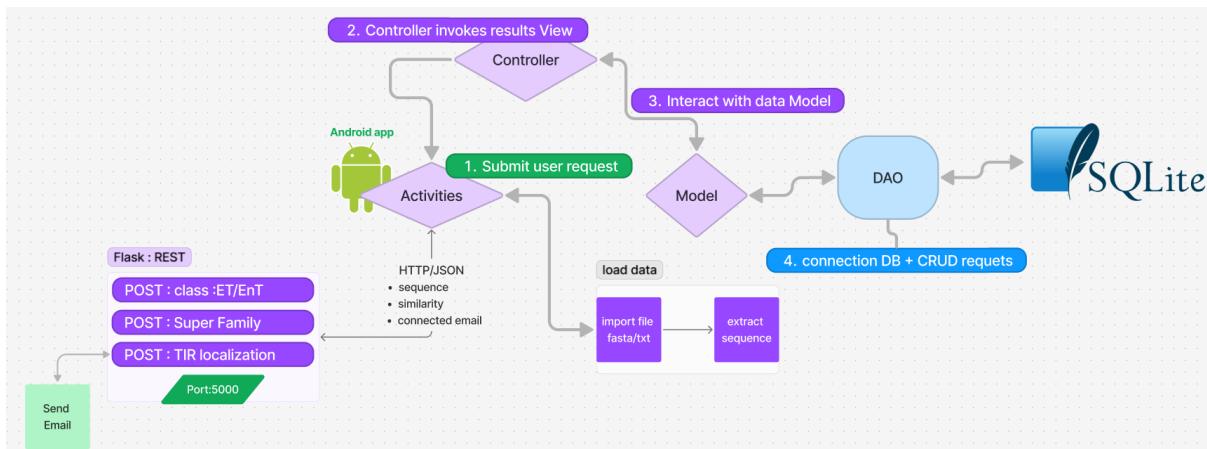


Figure 46 : architecture front-end / back-end

L'architecture illustre un projet de type front-end et back-end avec une application Android connectée à un service REST backend, une base de données SQLite et une fonctionnalité d'envoi d'e-mails. Voici une description détaillée de chaque composant de cette architecture :

1. Front-End (Application Android)

- Activités Android :** L'utilisateur interagit avec l'application Android pour soumettre des requêtes. Ces requêtes peuvent inclure des séquences livrées par l'utilisateur, des similarités et l'e-mails associé au user connecter, qui sont envoyés sous forme de requêtes HTTP/JSON vers le backend.

2. Back-End (Flask REST API)

- Service REST Flask :** Le backend est un service REST développé avec Flask qui écoute sur le port 5000. Ce service gère plusieurs types de requêtes POST, notamment :
 - POST : classification binaire : ET/EnT**
 - POST : classification multiclass : Super Family**
 - POST : localisation des TIRs**

3. Envoi d'e-mails

- Service d'envoi d'e-mails :** Après le traitement des données, un e-mail est envoyé, potentiellement pour notifier l'utilisateur par les résultats de la 3eme partie Localisation des TIR. Ce service est intégré au système et interagit avec Flask REST pour envoyer les informations par e-mail.

4. DAO (Data Access Object)

- Connexion à la base de données :** Le DAO gère les interactions avec la base de données SQLite, effectuant des opérations CRUD (Create, Read, Update, Delete) pour stocker et récupérer les données nécessaires de l'utilisateur connecté.

5. Contrôleur (Controller)

- Controller :** Après la réception d'une requête, le contrôleur invoque la vue des résultats en interagissant avec le modèle de données **Model** comme illustré ci-dessous.

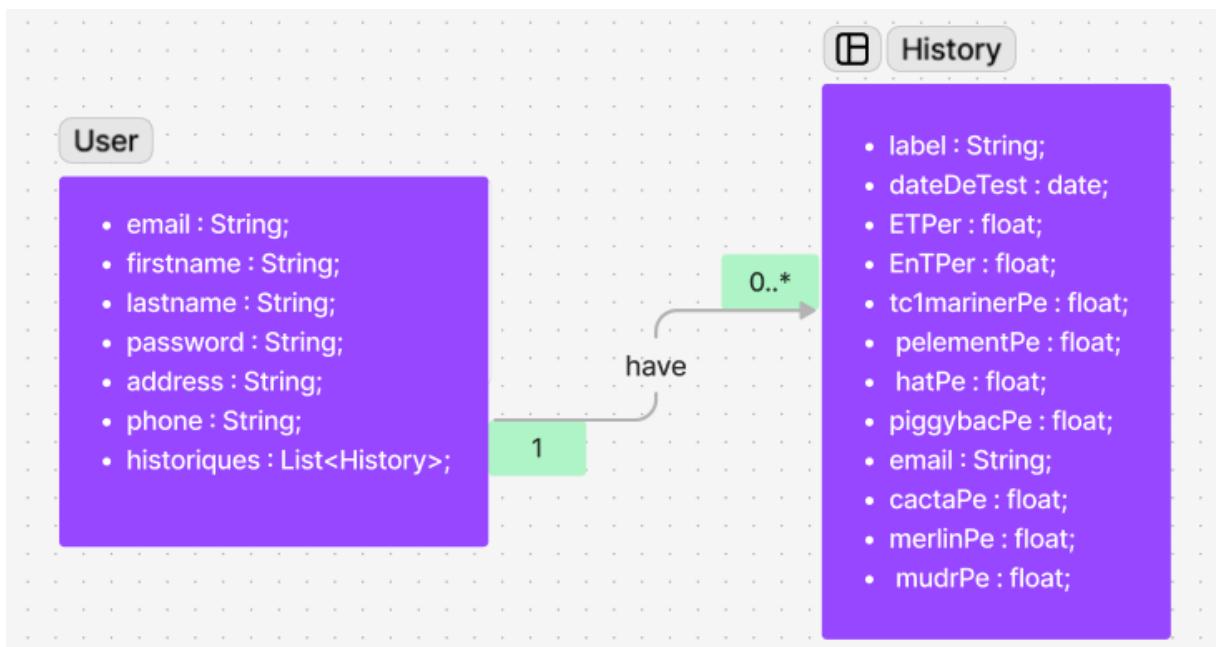


Figure 47 : diagramme de classe Front-End

4. Présentation des interfaces

En s'appuyant sur les outils décrits ci-dessus, nous avons pu réaliser notre application Mobile. Cette partie présente les interfaces du projet.

a. Login / signup

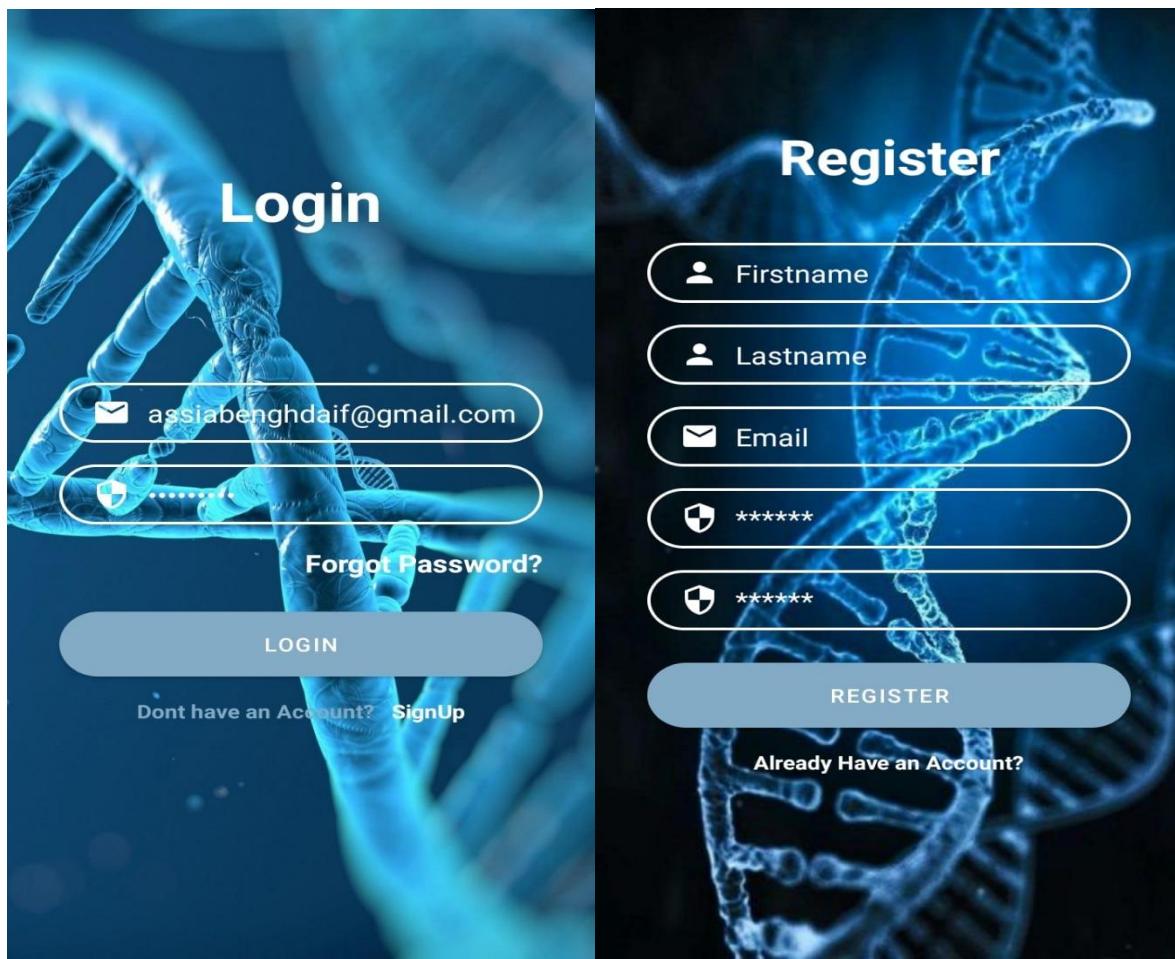


Figure 48 : activités login/signup

b. Activité Profile

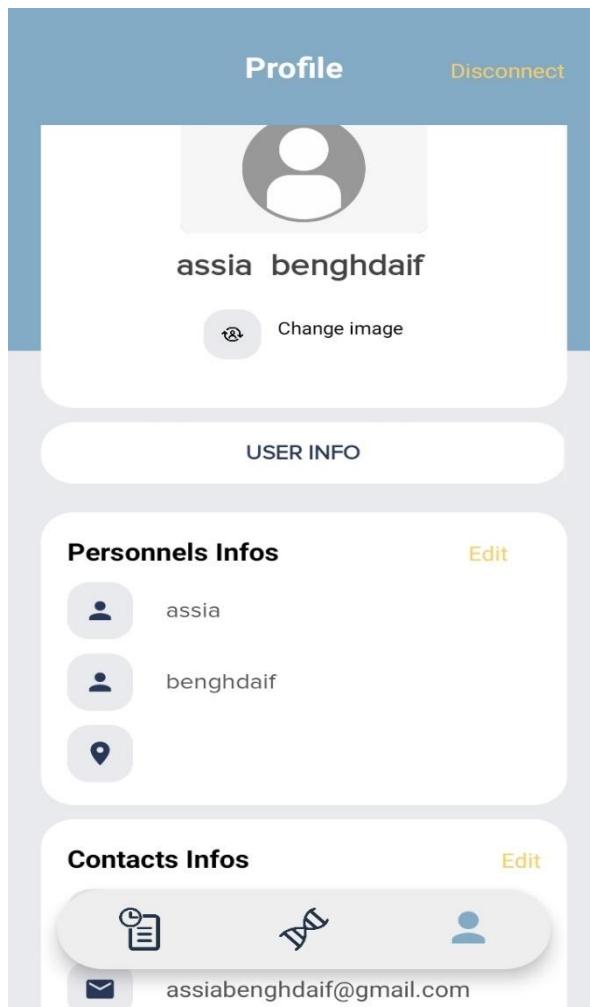


Figure 49 : activité profile

Cette activité est permise à l'utilisateur de modifier ses informations fournis déjà en sign up et d'ajouter des nouvelles informations tell que numéro de téléphone, adresse et importer / pris une photo de profile. Ainsi que déconnecter. Toutes les données sont sauvegardées dans une Base de Données Relationnelle (SGBDR) SQLite.

c. Activité de traitement

D'après les activités suivantes l'utilisateur peut importer un fichier FASTA/TXT contient une séquence d'ADN, automatiquement un bouton « Predict » pour faire prédire la classe de cette séquence « ET/EnT » comme illustrer ci-dessus avec un Pie Chart des probabilités de chaque classe.

Caractérisation d'éléments transposables : les transposons à TIR



Figure 50 : activité des classes ET/EnT

Si et seulement si une séquence contient des ET une zone de choix sera afficher.

L'utilisateur est devant un « seekbar » pour choisir la similarité entre les TIRs pour les localiser.

Même principe que l'activité dernière la classe sera afficher avec un Pie Chart des probabilités et en même temps un email sera envoyé à l'utilisateur avec les résultats (deux fichiers .TXT contient les résultats (**voir Chapitre IV, 2.3)**) de la localisation des TIRs.

Caractérisation d'éléments transposables : les transposons à TIR

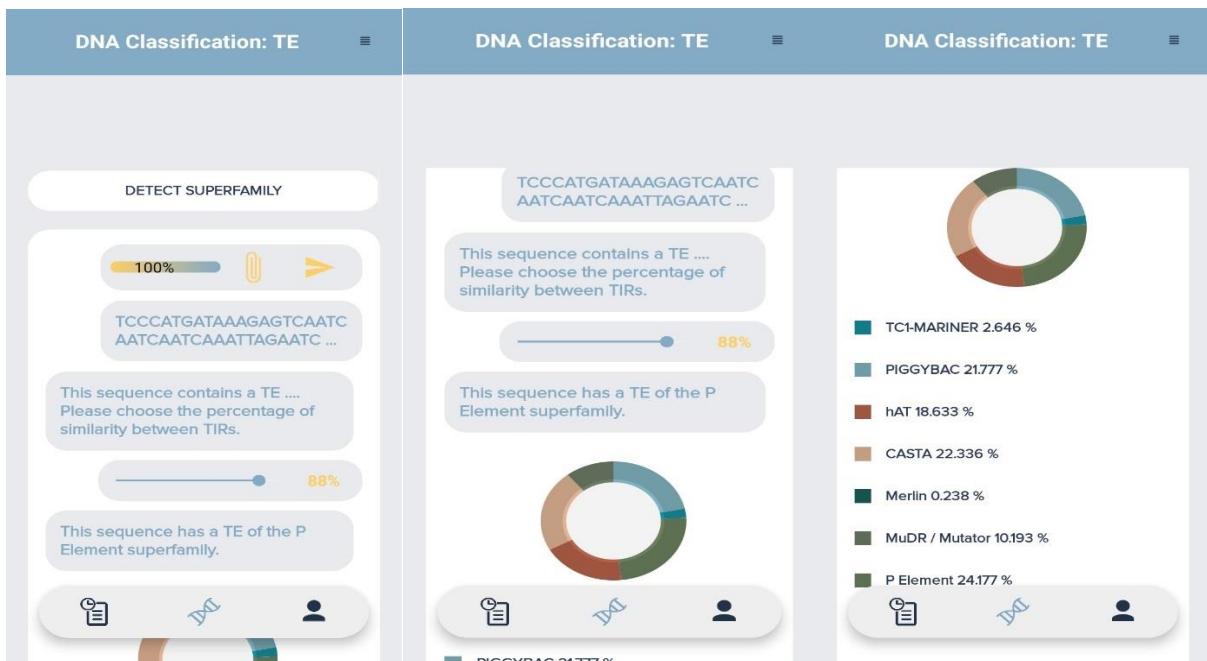


Figure 51 : activité ET SuperFamily

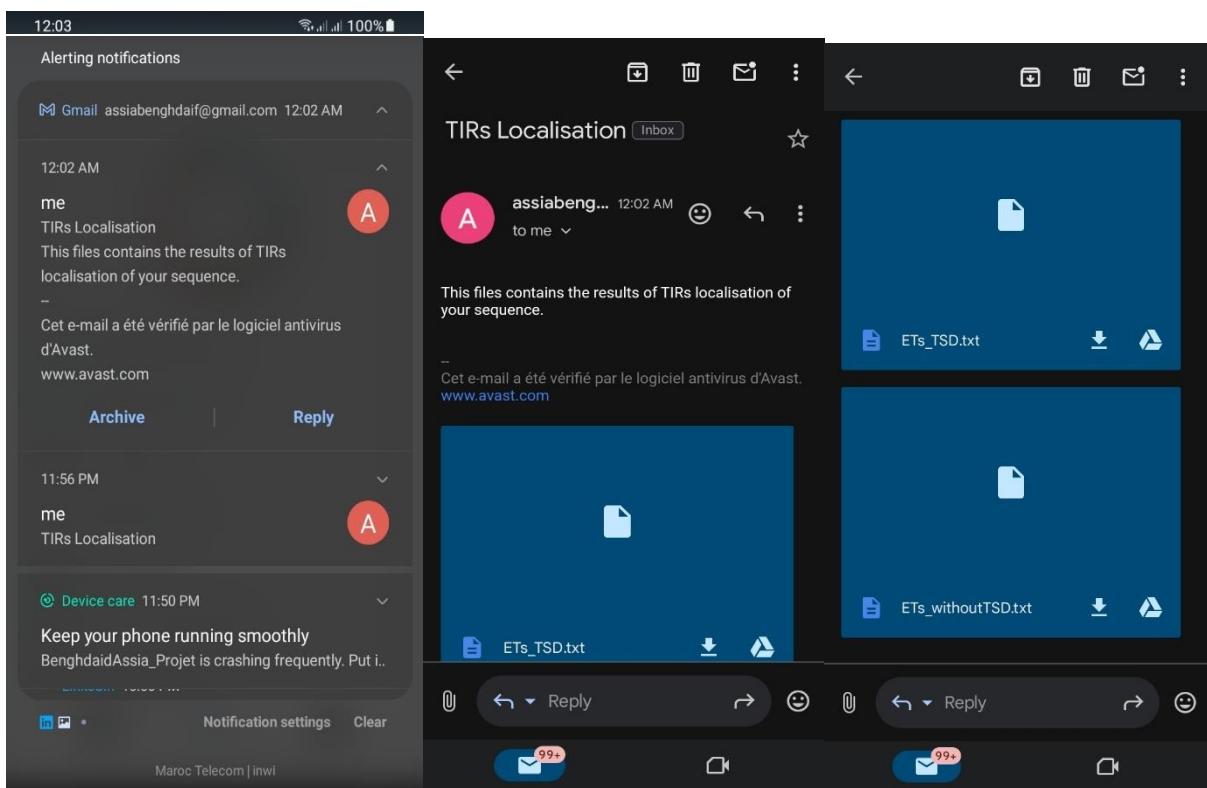


Figure 52 : TIRs Localisation : email envoyé

d. Activité historique

L'utilisateur peut avoir son historique des tests faite déjà comme illustrer il suffit de cliquer sur la ligne est va rediriger vers une autre activité avec l'ensemble des informations de la prédition :

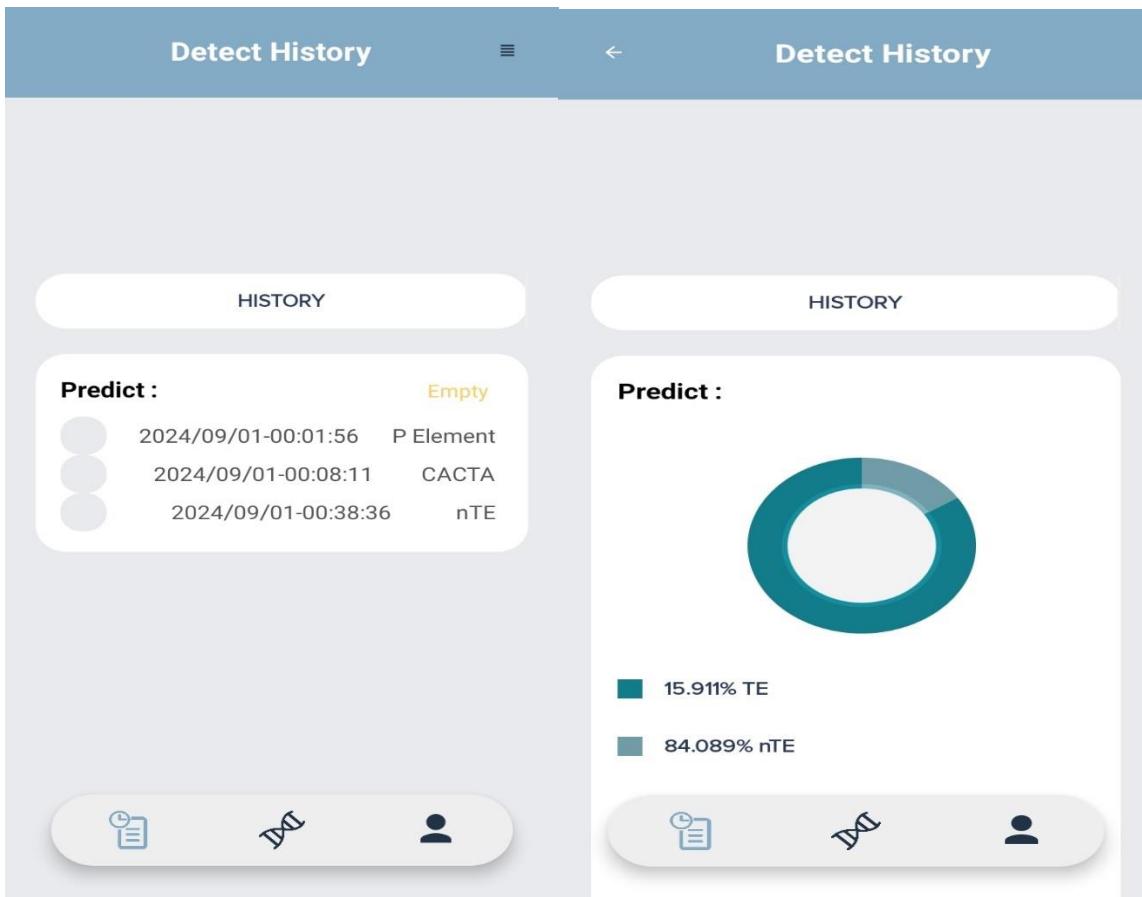


Figure 53 : activité historique

5. Conclusion

Dans ce chapitre, on a présenté l'environnement matériel et logiciel du projet. Nous avons par la suite, élaboré quelques aperçus du fruit de notre travail à travers des interfaces résultant des jeux de tests effectués.

Conclusion et perspectives

Le présent projet est le fruit d'un effort considérable que nous avons pu investir grâce au désir d'améliorer la manipulation et faciliter la détection et la classification de ET et ses superfamilles.

En fait, notre travail s'inscrit dans le domaine bio-informatique pour le but de fiabiliser et bien maintenir la classification des ETs par la manipulation des séquences d'ADN et l'extraction des données pertinentes de ces séquences tous en utilisant l'IA et les algorithmes de ML/DL.

Dans ce rapport nous avons exposé les étapes que m'a aidé à développer m'application mobile ainsi que les modèles entraînés :

Premièrement, nous avons présenté le cadre général du projet et des généralités sur les éléments transposables.

Deuxièmes, nous avons présenté en détail les superfamilles (tc1-mariner, BiggyBac, P élément, hAT, Merlin, CACTA et MuDR) et leurs structures.

Puis, la présentation générale des modèles DL/ML utilisés.

Ensuite, la présentation de l'étude technique, architecture de classification et le processus suivis pour le prétraitement des données et les modèles utilisés.

Et finalement, la présentation des choix techniques pour développer l'application puis la présentation des interfaces.

En guise de perspectives, ce projet n'est que le commencement, en proposant par la suite de gérer toutes les autres superfamilles. De plus, nous pourrions même prédire quand ces éléments transposables pourraient subir des mutations futur.

Bibliographie et Webographie

<https://www.tensorflow.org/>

<https://www.python.org/>

<https://scikit-learn.org/>

<https://www.kaggle.com/>

<https://www.tensorflow.org/guide/keras>

<https://www.jetbrains.com/idea/>

<https://www.java.com/fr/>

<https://colab.research.google.com/>

<https://flask-restful.readthedocs.io/en/latest/>

[1] <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2998295/>

[2] <https://www.sciencedirect.com/science/article/abs/pii/S1055790315000664>

[3] <https://genomebiology.biomedcentral.com/articles/10.1186/s13059-019-1905-y>

[4] <https://mobilednajournal.biomedcentral.com/articles/10.1186/s13100-020-00223-x>

[5] <https://pubmed.ncbi.nlm.nih.gov/31976169/>

[6] Manisha Panta, Avdesh Mishra, Md Tamjidul Hoque, Joel Atallah, ClassifyTE: a stacking-based prediction of hierarchical classification of transposable elements, Bioinformatics, Volume 37, Issue 17, September 2021, Pages 2529–2536, <https://doi.org/10.1093/bioinformatics/btab146>

[7] Haidong Yan, Aureliano Bombarely, Song Li, DeepTE: a computational method for de novo classification of transposons with convolutional neural network, Bioinformatics, Volume 36, Issue 15, August 2020, Pages 4269–4275, <https://doi.org/10.1093/bioinformatics/btaa519>

[8] Murilo Horacio Pereira da Cruz, Douglas Silva Domingues, Priscila Tiemi Maeda Saito, Alexandre Rossi Paschoal, Pedro Henrique Bugatti, TERL: classification of transposable elements by convolutional neural networks, Briefings in Bioinformatics, Volume 22, Issue 3, May 2021, bbaa185, <https://doi.org/10.1093/bib/bbaa185>

[9] <https://mobilednajournal.biomedcentral.com/articles/10.1186/s13100-020-00212-0>

[10] <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1461711/>

- [11] <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4399808/>
- [12] <https://mobilizednajournal.biomedcentral.com/articles/10.1186/s13100-019-0153-8>
- [13] <https://www.biorxiv.org/content/10.1101/2020.01.09.900282v1.full>
- [14] <https://pubmed.ncbi.nlm.nih.gov/15190130/>
- [15] <https://mobilizednajournal.biomedcentral.com/articles/10.1186/s13100-022-00265-3>
- [16] <https://www.iab-aib.org/fichier/ia/IA.pdf>
- [17] (PDF) Natural Language Processing: History, Evolution, Application, and Future Work
https://www.researchgate.net/publication/350058919_Natural_Language_Processing_History_Evolution_Application_and_Future_Work [accessed Aug 12 2024].
- [18] What is natural language generation (NLG)? <https://www.qualtrics.com/au/experience-management/customer/natural-language-generation/>.
- [19] IBM (2023). What is Natural Language Processing? | IBM. [online] IBM. <https://www.ibm.com/topics/natural-language-processing>.
- [20] Kavlakoglu, E. (2020). NLP vs. NLU vs. NLG: the differences between three natural language processing concepts. [online] IBM Blog. <https://www.ibm.com/blog/nlp-vs-nlu-vs-nlg-the-differences-between-three-natural-language-processing-concepts/>.
- [21] <https://www.ibm.com/topics/random-forest>
- [22]
<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesClassifier.html>
- [23] https://www.researchgate.net/publication/341967355_Navo_Minority_Over-sampling_Technique_NMOTe_A_Consistent_Performance_Booster_on_Imbalanced_Datasets/figures?lo=1
- [24] <https://www.ibm.com/docs/fr/spss-modeler/saas?topic=models-how-svm-works>
- [25] <https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/>
- [26] <https://www.sciencedirect.com/science/article/abs/pii/S0168169923000935>
- [27] <https://academic.oup.com/bioinformatics/article/37/15/2112/6128680>
- [28] <https://arxiv.org/pdf/2306.15006>

[29] <https://www.uniprot.org/>

[30] <https://www.dfam.org/>

[31] <https://www.ncbi.nlm.nih.gov/>

Annexes

Annexe 1

Superfamille	Taille des séquences	Motif des TSD	Caractéristiques
Tc1/Mariner	Environ 1 300 à 2 500 paires de bases	2 paires de bases (pb) (exemple : "TA")	Contient une seule ORF codant pour la transposase. Utilise un mécanisme de "cut and paste" pour la transposition
hAT	Entre 2 000 et 5 000 paires de bases.	Typiquement 8 pb (exemple : "TTAATTAA").	Nommés d'après les trois premiers membres découverts (hobo, Activator, Tam3). Codent pour une transposase
Mutator (MuDR)	Environ 4 000 à 10 000 paires de bases.	9 pb (séquence spécifique variable).	Codent pour plusieurs protéines, y compris une transposase et une régulatrice. Sont connus pour leur rôle dans la mutagenèse chez les plantes
CACTA	Environ 3 000 à 12 000 paires de bases.	2 à 3 pb (exemples : "TA" ou "TTA").	Connus pour leur structure complexe et leur capacité à réarranger le génome. Codent pour une transposase et parfois d'autres protéines régulatrices.
P-element	: Environ 2 900 paires de bases	8 pb (exemple : "TCTAGAAA").	Découvert initialement chez la drosophile, contient des séquences codant pour une transposase.
PiggyBac	4 pb (exemple : "TTAA").
Merlin	8 à 9 pb.

Annexe 2 : définitions

La **Linguistique** est la science qui étudie le langage humain sous tous ses aspects. Elle s'intéresse à la structure, au fonctionnement, à l'évolution et à l'usage des langues.

La **classification binaire** (ou la classification binomiale) est une transformation de données qui vise à répartir les membres d'un ensemble dans deux groupes disjoints selon que l'élément possède ou non une propriété ou fonctionnalité donnée.

La **classification multiclassée** est une tâche de classification où un modèle doit attribuer une seule étiquette parmi plusieurs classes possibles à chaque instance. Contrairement à la classification binaire, où il n'y a que deux classes, la classification multiclassée implique trois classes ou plus.