

République Algérienne Démocratique Et Populaire
Ministère De L'enseignement Supérieur Et De La
Recherche Scientifique



Université Larbi Ben M'hidi-Oum El Bouaghi-
Faculté Des Sciences Exactes Et Des Sciences De La Nature Et De La Vie
Département Mathématique et Informatique

MEMOIRE
EN VUE DE L'OBTENTION DU DIPLOME DE MASTER EN
INFORMATIQUE
OPTION : Vision Artificielle

Thème

Clustering en utilisant Markov Blanket
Approximation dans un environnement IoT

Présenté par :

Rouag Nor EL Imane
Malla Amira

Dirigé par :

Dr. SID KARIMA

Devant le jury composé de :

Dr. KOUAH SOFIA Présidente
Dr. SID Karima Encadrant
Dr. BOULKAMH Chouaib Examineur

Année universitaire 2022-202

REMERCIEMENT

Tout d'abord, nous voudrions remercier le Dieu puissant et miséricordieux de nous avoir donné le courage, la force et la volonté de faire ce travail. Nous tenons à exprimer nos sincères remerciements à notre encadrante, Dr : Sid Karima, pour sa présence et son aide précieuse dans nos travaux consignés dans ce mémoire de fin d'études.

Nous adressons nos remerciements aux membres du jury, qui ont pris la peine d'évaluer ce travail et devant qui, nous avons l'honneur de l'exposer.

Nous remercions également tous nos professeurs, ainsi que tous ceux qui nous ont enseigné de près ou de loin.

Résumé

L'Internet des Objets « IoT » est largement appliqué dans le domaine d'agriculture «Agriculture intelligente » où il a prouvé son efficacité dans l'amélioration de la qualité et la quantité des produits agricoles en optimisant le coût, le temps et la consommation de l'eau. L'apprentissage automatique peut être utilisé en collaboration avec l'IoT pour avoir un système complet intelligent. Le clustering est une méthode d'apprentissage non-supervisé très utilisé lorsqu'on n'a pas les sorties désirées. Dans ce contexte, nous avons proposé une approche pour grouper les données agricoles captées dans un environnement IoT en utilisant l'Approximation d'une couverture de Markov qui se base principalement sur un coefficient nommé le coefficient de corrélation d'information afin d'extraire l'information utile « un groupe d'intérêt ».

Mots-clés : Internet des Objets ; Apprentissage automatique, Markov Blanket Approximation, Agriculture de précision, Clustering.

Abstract

Internet of Things "IoT" is widely applied in the field of agriculture "Smart Agriculture" where it has proven its effectiveness in improving the quality and quantity of agricultural products by optimizing the cost, time and consumption of the water. Machine learning can use in conjunction with IoT to have a complete intelligent system. Clustering is an unsupervised learning method generally used when you do not have the desired outputs. In this context, we have proposed an approach to group agricultural data captured in an IoT environment using the Markov Blanket Approximation, which based on a coefficient called the information correlation coefficient in order to extract the useful "interest group" information.

Keywords: Internet of Things; Machine Learning, Markov Blanket Approximation, Precision Agriculture, Clustering.

ملخص

يتم تطبيق إنترنت الأشياء "IoT" على نطاق واسع في مجال الزراعة "الزراعة الذكية" حيث اثبتت فعاليتها في تحسين جودة وكمية المنتجات الزراعية من خلال تحسين التكلفة والوقت واستهلاك الماء. يمكن استخدام التعلم الآلي جنبًا إلى جنب مع إنترنت الأشياء للحصول على نظام كامل ذكي. التجميع هو طريقة تعلم غير خاضعة للرقابة تستخدم على نطاق واسع عندما لا يكون لديك المخرجات المرغوبة. في هذا السياق، اقترحنا نهجًا لتجميع البيانات الزراعية التي تم التقاطها في بيئة إنترنت الأشياء باستخدام Markov Blanket Approximation الذي يعتمد بشكل أساسي على معامل يسمى معامل ارتباط المعلومات من أجل استخراج معلومات "مجموعة الاهتمامات" المفيدة.

الكلمات المفتاحية: إنترنت الأشياء، التعلم الآلي ، Markov Blanket Approximation ، الزراعة الدقيقة ، التجميع.

Table des matières

Introduction générale.....	1
Chapitre 1 : L’IoT : Architecture & Fonctionnement.....	3
1. Introduction.....	3
2. Internet des Objets : définition & caractéristiques.....	3
3. L’architecture de l’IoT.....	5
3.1. La couche de perception.....	6
3.2. La couche Réseau.....	6
3.3. La couche d’application.....	6
4. Les composants de l’IoT.....	7
5. Les technologies de l’IoT.....	9
6. L’IoT en Agriculture.....	9
7. Conclusion.....	11
Chapitre 2 : L’intelligence artificielle et ses sous-domaines « Clustering : unsupervised learning ».....	12
1. Introduction.....	12
2. L’intelligence artificielle et ses sous domaines.....	12
2.1. L’apprentissage automatique.....	13
a. Apprentissage supervisé.....	14
b. Apprentissage non-supervisé.....	14
c. L’apprentissage par renforcement.....	15
2.2. L’apprentissage profond.....	15
2.3. Data Science VS. Machine learning VS. Data Mining.....	16
3. Apprentissage non supervisé et le regroupement de données.....	17
3.1. Le clustering : définition.....	17

3.2. Les objectifs du clustering & types	18
a. Le Principe & les objectifs	18
b. Catégories d’algorithmes de clustering	19
c. Mesure de proximité.....	20
3.3. Les algorithmes de clustering	20
a. L’algorithme K-means.....	20
b. Fuzzy C-means (FCM)	21
c. Clustering hiérarchique	22
4. Le clustering dans le domaine d’agriculture	24
5. Conclusion	25
Chapitre 3 : L’approche proposée « Conception »	26
1. Introduction.....	26
2. La simulation des données : Simulateur TinkerCad	26
2.1. Description.....	26
2.2. Arduino & capteurs	28
3. Génération de la base de données	29
3.1. En utilisant le simulateur TinkerCad	29
3.2. Une base de données alternative.....	32
4. L’approche proposée.....	33
5. Clustering basé sur l’approximation d’une couverture de Markov.....	35
5.1. L’approximation d’une couverture de Markov	36
5.2. L’algorithme proposé	37
6. Conclusion	39
Chapitre 4 : Implémentation & Résultats.....	40
1. Introduction.....	40
2. Les outils et les bibliothèques de développement	40
2.1. Le langage Python	40

2.2.	Anaconda Navigator	40
2.3.	L'environnement de programmation Spyder (sous Anaconda).....	41
2.4.	NumPy	42
2.5.	Pandas	42
2.6.	Pygubu	42
3.	Les implémentions	43
3.1.	La préparation de données	43
3.2.	Le calcul de coefficient de corrélation	43
3.3.	Le calcul de l'approximation d'une couverture de Markov & Clustering.....	44
3.4.	Générer le groupe d'intérêt « information utile ».....	46
4.	Résultats & Discussion	46
4.1.	Scénario d'évaluation 1 : Importer les données.....	47
4.2.	Scénario d'évaluation 2 : Le clustering	48
4.3.	Scénario d'évaluation 3 : Générer le groupe d'intérêt.....	50
5.	Conclusion	51
	Conclusion Générale	52
	Références Bibliographiques	53

Liste des Figures

Figure 1. 1. Ubiquité de l'IoT	4
Figure 1. 2. Architecture en trois couches de l'IoT	5
Figure 1. 3. Pyramide de l'IoT, architecture-fonctionnement de l'IoT	8
Figure 1. 4. Applications de l'IoT dans divers secteurs de l'agriculture intelligente	10
Figure 2. 1. Les sous domaines de l'intelligence artificielle.....	13
Figure 2. 2. L'apprentissage automatique en général	13
Figure 2. 3. Apprentissage supervisé vs. non-supervisé	15
Figure 2. 4. Réseau de neurones VS. Réseau de neurones profond	16
Figure 2. 5. Data Science VS. Machine learning VS. Data Mining.....	17
Figure 2. 6. Exemple de clustering sous 5 groupes	18
Figure 2. 7. Le processus général de clustering	19
Figure 2. 8. Clustering Hiérarchique VS. Clustering partitionnel.....	20
Figure 2. 9. Approche Bottom-up VS. Approche Top-Down.....	23
Figure 3. 1. La création d'un circuit sous TinkerCad	27
Figure 3. 2. La carte Arduino (entrées-composants-sorties)	28
Figure 3. 3. Les types de cartes Arduino	28
Figure 3. 4. La structure d'un capteur	29
Figure 3. 5. Le circuit obtenu	30
Figure 3. 6. Code source pour la génération du circuit	31
Figure 3. 7. Résultat de la simulation.....	31
Figure 3. 8. Structure de la base de données	32
Figure 3. 9. La structure générale de l'approche proposée	34
Figure 3. 10. Entropies individuelles (($H(X)$, $H(Y)$), jointes ($H(X, Y)$), l'information mutuelle $I(X; Y)$	36
Figure 3. 11. Exemple d'une couverture de Markov dans un réseau Bayésien,	36
Figure 4. 1. L'interface de l'Anaconda Navigator	41
Figure 4. 2. L'interface de Spyder	42
Figure 4. 3. Code Python pour télécharger Dataset.....	43

Figure 4. 4. Code Python pour calculer l'ICC.....	44
Figure 4. 5. Code Python pour calculer Markov Blanket Approximation	45
Figure 4. 6. Code Python pour le clustering.....	45
Figure 4. 7. Code Python pour générer le groupe d'intérêt	46
Figure 4. 8. L'interface principale	47
Figure 4. 9. Importer les données	48
Figure 4. 10. Le résultat de clustering	49
Figure 4. 11. Extraction de l'information utile.....	50

Liste des Tableaux

Tableau 1. 1. Les principaux composants de l'IoT	8
Tableau 1. 2. IoT en agriculture "étude des travaux connexes"	11
Tableau 2. 1. K-Means VS. Fuzzy C-Means.....	22
Tableau 2. 2. Comparaison entre le clustering partionnel & hiérarchique.....	23
Tableau 2. 3. Clustering en agriculture « étude des travaux connexes »	25
Tableau 3.1. Les composants du circuit	30
Tableau 4. 1. Les résultats de clustering selon des différents vecteurs d'intérêt	48

Introduction Générale

Introduction générale

L'Internet des Objets « Internet of Things en anglais » s'étend, se développe et devient de plus en plus populaire de jour en jour. Actuellement, environ 5 milliards d'objets sont connectés à l'Internet. Selon les estimations, plus de 50 milliards d'objets devraient être connectés à l'Internet d'ici 2025. L'IoT ouvre un monde de possibilités pour de nouvelles applications, qui sont désormais utilisées dans divers domaines, notamment les systèmes intelligents de surveillance domestique, la gestion de la chaîne d'approvisionnement des produits, l'agriculture de précisionetc. (Deepak, 2022). Le concept IoT a connu plusieurs définitions, la plus claire et la plus concise est celle donnée par ITU (International Telecommunication Union) : « *L'IoT est une technologie qui résout principalement l'interconnexion entre l'humain à un objet, un objet à un objet et l'humain à l'humain* ».

L'application de l'IoT en agriculture « Smart Agriculture » est un axe de recherche très actif, plusieurs travaux ont été enregistrés dans ce contexte où l'objectif principal consiste à améliorer la qualité et la quantité des produits agricoles en optimisant le coût, le temps et la consommation (ex. la consommation de l'eau à l'irrigation dans une serre intelligente). Les agriculteurs peuvent installer des capteurs pour collecter des données climatiques (ex. température, l'humidité..), puis, utiliser ces données pour extraire l'information utile « valeur » qui optimise l'opération d'irrigation.

L'IoT a une forte relation avec l'intelligence artificielle « IA », où nous pouvons voir la première comme une partie hardware et la deuxième comme une partie software dans un système complet intelligent. Généralement, l'IA définit comme un ensemble des algorithmes mimant le cerveau humain. L'apprentissage automatique est l'un des sous domaines de l'intelligence artificielle, divisé principalement en deux catégories : supervisé (la sortie désirée est disponible) et non-supervisé (la sortie désirée n'est pas disponible) (Sid, 2021). Le clustering est une méthode d'apprentissage non-supervisé qui vise à grouper les données dans des clusters en respectant la règle « *similarité maximale au sein d'un cluster et dissimilarité maximale entre les clusters* ». K-means, Fuzzy C-means, le clustering hiérarchique sont les algorithmes de clustering les plus utilisés dans « Smart Agriculture ». Les principales limitations de ces algorithmes sont : demande le nombre de clusters à l'avance (K-means, Fuzzy C-means) ; Se

Introduction générale

termine souvent à un optimum local, Ils évoluent mal avec une complexité temporelle d'au moins $O(n^2)$, où n est le nombre total d'objets (clustering hiérarchique). La proposition d'un nouvel algorithme surmonte ces limitations sera un défi à relever.

L'objectif de ce travail consiste à proposer une approche pour grouper des données agricoles captées dans un environnement IoT, afin d'extraire les informations les plus utiles « Valeurs » en appliquant un nouvel algorithme basé sur l'approximation d'une couverture de Markov. L'algorithme utilise les concepts de la théorie d'information : l'entropie, l'entropie conjointe, l'information mutuelle et le coefficient de corrélation de l'information.

La suite de ce travail est organisée comme suit :

→ *Chapitre 1 : L'IoT : Architecture & Fonctionnement ;*

Introduit l'Internet des Objets, ses différentes définitions, son architecture, les principaux composants & technologies pour terminer par une discussion sur l'application de l'IoT en agriculture ;

→ *Chapitre 2 : L'intelligence artificielle et ses sous-domaines « clustering : unsupervised learning » ;*

Ce chapitre présente le concept d'apprentissage automatique, en mettant l'accent sur ses différents types, notamment l'apprentissage non-supervisé « le clustering » ;

→ *Chapitre 3 : Conception : approche proposée*

Ce chapitre est consacré à la présentation de l'approche proposée, la structuration de la base de données et l'algorithme de clustering développé.

→ *Chapitre 4 : Implémentation & Résultats*

Expose le processus d'implémentation et les résultats obtenus selon des différents scénarios d'évaluation.

Le travail se termine par une conclusion générale et quelques perspectives.

Chapitre 1 :

**L'IoT : Architecture &
Fonctionnement**

Chapitre 1 : L'IoT : Architecture & Fonctionnement

1. Introduction

Depuis 2014, le nombre d'objets connectés a dépassé le nombre d'humains connectés. L'Internet des Objets « IoT » un nouveau concept proposé pour optimiser la connexion des objets dans le monde entier. IoT est un réseau mondial dynamique qui se base sur des protocoles de communication standardisés permettant l'interaction et l'intégration des objets virtuels et physiques par l'échange de données et d'information (DJE Bri, 2018)

Dans ce chapitre, nous allons donner un aperçu général sur le concept de l'Internet des objets en exposant son architecture, son fonctionnement, ses technologies et ses domaines d'application, notamment dans le secteur agricole.

2. Internet des Objets : définition & caractéristiques

L'Internet des objets « Internet of Things –IoT- en anglais », Il n'existe pas une définition unique acceptable par la communauté mondiale des utilisateurs.

La définition la plus commune de l'Internet des Objets est : un réseau d'objets physiques. Internet n'est pas seulement un réseau d'ordinateurs. Mais, il est également évolué pour devenir un réseau d'appareils de tous types et de toutes tailles, tels que des véhicules, des téléphones intelligents, des appareils électroménagers, des jouets, des appareils photo, des instruments médicaux, des systèmes industriels, des personnes et bâtiments, tous connectés, communiquant et partageant des informations sur la base de protocoles stipulés afin de réaliser une réorganisation intelligente, un positionnement, un traçage, une surveillance personnelle en ligne et en temps réel (Talal, 2019).

Nous pouvons dire que l'objectif principal de l'IOT est de permettre aux « objets » physiques et virtuelles d'être connectés à tout moment, n'importe où, avec n'importe quoi et n'importe qui

Chapitre 1 : L'IoT : Architecture & Fonctionnement

(Figure 1.1), idéalement en utilisant n'importe quel *chemin/réseau/service*, pour atteindre un certain but (Talal, 2019).

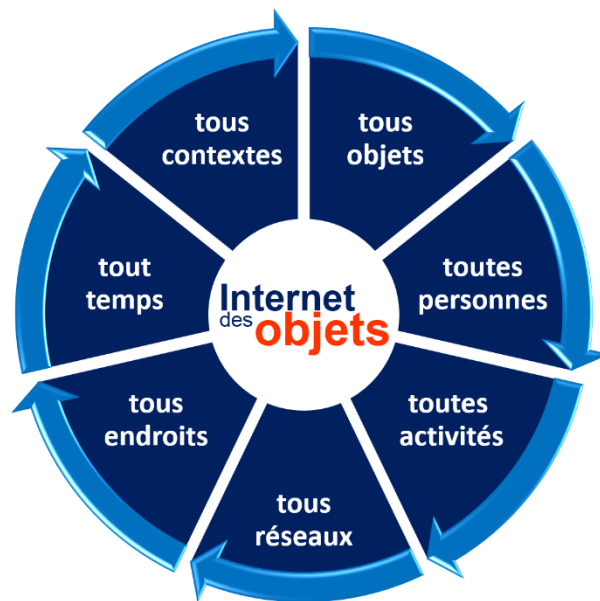


Figure 1. 1. Ubiquité de l'IoT

Selon (Safa, 2018), les caractéristiques de base de l'Internet des objets sont les suivantes :

1. *Intelligence* : elle rend les choses plus intelligentes en leur fournissant des techniques informatiques et des logiciels ;
2. *Connectivité* : elle a la capacité de produire et de consommer des données ;
3. *Interconnexion* : selon l'Internet des objets, tout peut être lié et utilisé au sein de l'infrastructure d'information et de communication ;
4. *Sens* : elle utilise des objets intelligents pour connaître l'environnement physique.
5. *Alimentation* : une alimentation intelligente doit être conçue ;
6. *Sécurité* : la sécurité et la sûreté des composants de l'Internet des objets doivent être garanties lors de la conception, du travail et de l'intégrité des données en circulation ;
7. *Diversité et différence* : les appareils et les réseaux utilisés dans l'Internet des objets sont interconnectés, malgré sa diversité basée sur ses différentes plates-formes physiques.

Chapitre 1 : L'IoT : Architecture & Fonctionnement

3. L'architecture de l'IoT

L'un des principaux défis à relever avec le domaine technologique pour favoriser le déploiement des systèmes IoT est de définir une architecture de référence qui supporte les fonctionnalités actuelles et les extensions futures. Pour cette raison, une telle architecture doit être (Marco, 2021):

- *Évolutive « scalable »*, afin de gérer le nombre croissant d'appareils et de services sans dégrader leurs performances ;
- *Interopérable*, afin que les appareils de différents fournisseurs (types) puissent coopérer pour atteindre des objectifs communs ;
- *Distributive*, pour permettre de créer un environnement distribué dans lequel, après avoir collecté les données à partir de différentes sources, sont traitées par différentes entités de manière distribuée et parallèle ;
- *Sécurisée*, afin d'arrêter l'accès non autorisé.

Actuellement, il n'existe pas une architecture de référence unique, mais l'architecture la plus utilisée c'est l'architecture en couches « Layers », où chaque couche est définie par ses fonctions et les périphériques utilisés dans cette couche. Il existe différents avis concernant le nombre de couches dans une architecture IoT. Cependant, selon la plupart des recherches, l'IoT fonctionne principalement sur trois couches appelées *couches de perception, de réseau et d'application*. La figure ci-dessous (Figure 1.2) illustre le cadre architectural de base à trois couches de l'IoT en précisant les appareils et les technologies qui englobent chaque couche (Rwan, 2015).

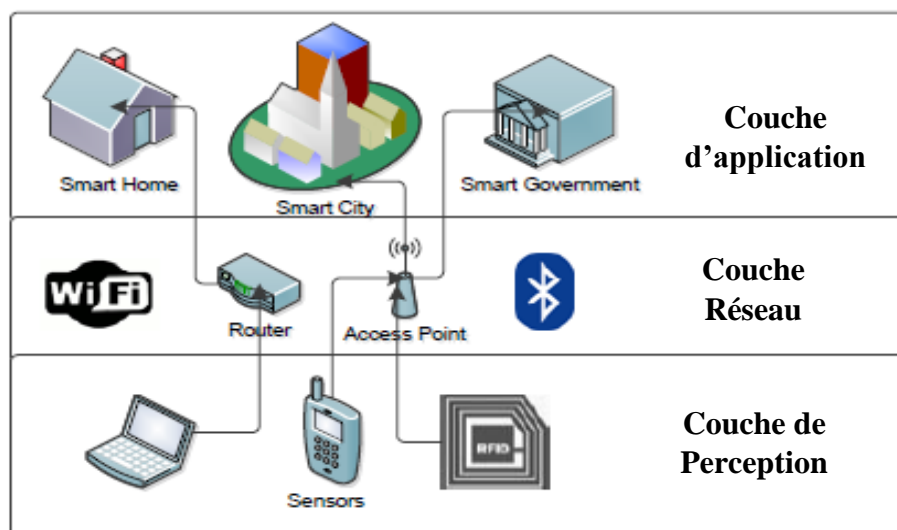


Figure 1. 2. Architecture en trois couches de l'IoT (Rwan, 2015).

Chapitre 1 : L'IoT : Architecture & Fonctionnement

3.1.La couche de perception

La couche de perception est également connue sous le nom de couche « Capteurs » ou « Sensors », elle représente le niveau physique, comprend des objets qui, étant capables d'interagir avec le monde extérieur et être dotés de capacités informatiques, devenir en quelque sorte « intelligent » ou « Smart ». Le but de cette couche est d'acquérir les données de l'environnement à l'aide de capteurs et d'actionneurs, détecte, collecte et traite les informations puis les transmet à la couche réseau (Marco, 2021) (Rwan, 2015). Ces objets intelligents, qui sont les briques essentielles sur lesquelles repose l'IoT, peuvent être des objets d'usage courant (un réfrigérateur, une télévision, une voiture, etc.) ou de simples appareils équipés de capteurs et de capacités de calcul. En général, les objets intelligents sont caractérisés par (Marco, 2021):

- *Communication* : les objets peuvent se connecter les uns aux autres et aux ressources sur Internet pour utiliser des données et des services, mettre à jour leur état et coopérer pour atteindre des objectifs communs ;
- *Identification* : les objets doivent être identifiés de manière unique ;
- *Adressable* : les objets peuvent être directement accessibles, c'est-à-dire adressés, pour être interrogés et/ou configurés à distance ;
- *Détection et activation* : les objets peuvent collecter des informations sur l'environnement et les manipuler grâce à l'utilisation de capteurs et d'actionneurs ;

3.2.La couche Réseau

La couche réseau de l'IoT remplit la fonction de routage et de transmission des données vers différents hubs et appareils IoT sur l'Internet. À cette couche, les plates-formes de cloud computing, les passerelles Internet, les dispositifs de commutation et de routage, etc. fonctionnent en utilisant les technologies les plus récentes, telles que WiFi, LTE, Bluetooth, 3G, Zigbee, etc. Les passerelles réseaux jouent le rôle d'un médiateur entre des différents nœuds IoT en agrégeant, filtrant et transmettant des données vers et depuis différents capteurs (Rwan, 2015)

3.3.La couche d'application

La couche application garantit l'authenticité, l'intégrité et la confidentialité des données. À cette couche, l'objectif de l'IoT qui consiste à créer un environnement intelligent est atteint. Elle comprend tous les logiciels nécessaires pour offrir un service spécifique. Dans ce niveau, les données des niveaux précédents sont stockées, agrégées, filtrées et traitées, et des bases de

Chapitre 1 : L'IoT : Architecture & Fonctionnement

données, des logiciels d'analyse, etc. sont utilisés. À la suite de ce processus de traitement, les données sont mises à disposition de véritables applications IoT (smart city, smart home, etc.) (Rwan, 2015) (Marco, 2021).

4. Les composants de l'IoT

Un système IOT est composé de plusieurs acteurs et composants technologiques qui assure le bon fonctionnement de ce dernier. Une solution IOT s'articule autour de différents composants, récapitulés dans le tableau suivant (Tableau 1.1) (Hizkia, 2019) (Meftah, 2020).

Composant IoT	Description
Objets physiques	Things.
Capteurs	Sentir l'environnement physique.
Actionneurs	Affecter l'environnement physique.
Données « Data »	Des éléments bruts récoltés depuis les objets ou les outils de processus industriels pour l'IOT. Afin de générer une valeur aux utilisateurs à partir de ces données, il est nécessaire de les stocker, archiver et sauvegarder dans des bases de données bien structurées (Figure 1.3).
Valeur « Information »	La valeur est le résultat d'un processus de traitement, corréler et analyser les données. Ces valeurs doivent elles-aussi être stockées, archivées et sauvegardées dans des bases de données spécialisées (Figure 1.3).
Personnes	Ex. l'être humain peut contrôler l'environnement via une application mobile.
Services	Ex. les services cloud, peuvent utiliser pour : <ul style="list-style-type: none">• Traiter les Big Data et les transformer en informations précieuses ;• Construire et exécuter des applications innovantes.
Plateformes	Type de middleware utilisé pour connecter les composants IoT (objets, personnes, servicesetc). doit fournir de nombreuses fonctions : <ul style="list-style-type: none">• Accès aux appareils ;

Chapitre 1 : L'IoT : Architecture & Fonctionnement

	<ul style="list-style-type: none">• Assurer une installation / un comportement correct de l'appareil ;• Analyse de données ;• Connexion interopérable au réseau local, au cloud ou à d'autres appareils.
Réseau (connectivité)	Les composants IoT sont reliés entre eux par des réseaux, utilisant diverses technologies, normes et protocoles sans fil et filaires pour fournir une connectivité omniprésente.

Tableau 1. 1. Les principaux composants de l'IoT

Le fonctionnement de l'IoT peut être résumé dans les étapes suivantes (Safa, 2018):

1. Les données sont obtenues à partir du sens des objets identifiés et connectés, par exemple les changements chimiques dans l'air, etc., selon le type de capteur. Différents capteurs peuvent être combinés pour concevoir un service intelligent.
2. Déclencheur de l'action, les données reçues à partir des objets sont traitées par un système ou dispositif intelligent qui détermine l'action automatique à invoquer.
3. Il existe un mécanisme de feedback à l'administration par l'appareil ou le système intelligent sur l'état du système général et évoque les résultats des actes.

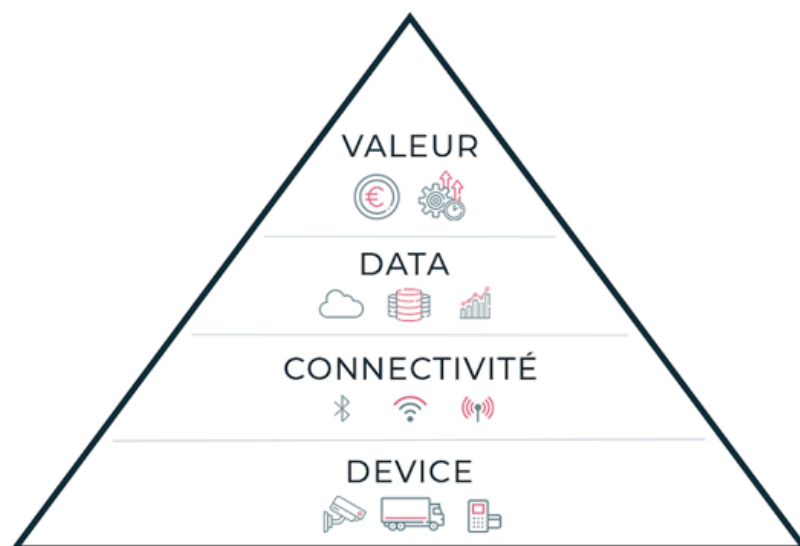


Figure 1. 3. Pyramide de l'IoT, architecture-fonctionnement de l'IoT¹

¹ <https://www.matooma.com/fr/s-informer/actualites-iot-m2m/architecture-solution-iot>

Chapitre 1 : L'IoT : Architecture & Fonctionnement

5. Les technologies de l'IoT

Les technologies IoT désignent les différentes solutions techniques qui permettent d'identifier les objets, de capter, de stocker, de traiter, et de transférer les données dans un environnement physique et entre des univers virtuels. Nous citons dans ce qui suit les principales technologies (Safa, 2018).

- *Radio Frequency Identification (RFID)* : Identification par radiofréquence, elle utilise des ondes radio pour la transmission sans fil de l'identité d'un objet sous un format de numéro de série. La technologie RFID joue un rôle important dans l'IoT pour résoudre les problèmes d'identification des objets.
- *Wireless Sensor Network (WSN)*: Réseau de capteurs sans fil, il s'agit d'un réseau sans fil composé d'appareils distribués qui utilisent des capteurs intelligents pour former un ensemble et coopérer à la surveillance des conditions physiques ou environnementales, telles que la température, le son, etc. Les informations collectées sont envoyées à un système centralisé pour une étape d'analyse.
- *Middleware* : « Internet of Things middleware » est un logiciel qui sert d'interface entre les composants de l'IoT, rendant possible la communication entre des éléments qui n'en seraient autrement pas capables.
- *Wireless Fidelity (Wi-Fi)* : il s'agit d'une technologie de mise en réseau qui permet aux ordinateurs et autres appareils de communiquer via un signal sans fil.
- *Bluetooth* : il s'agit d'une technologie sans fil peu coûteuse qui utilise des ondes radio à courte portée.
- *ZigBee* : un protocole développé pour améliorer les avantages des réseaux de capteurs sans fil.

6. L'IoT en Agriculture

L'IoT est largement appliqué dans plusieurs secteurs, nous citons : la E-santé, la ville intelligente, le transport... etc. Dans cette section, nous allons focaliser sur l'application de l'IoT dans le secteur d'agriculture pour avoir une agriculture intelligente (Smart Agriculture) (Figure 1.4). L'utilisation de l'Internet des objets dans l'agriculture peut permettre aux agriculteurs d'améliorer leur productivité, de réduire les coûts et d'augmenter la qualité de leurs produits. Par exemple, en utilisant des capteurs IoT pour surveiller les conditions météorologiques et l'humidité du sol, les agriculteurs peuvent ajuster l'irrigation et la

Chapitre 1 : L'IoT : Architecture & Fonctionnement

fertilisation pour maximiser la croissance des plantes tout en évitant le gaspillage d'eau et de nutriments.

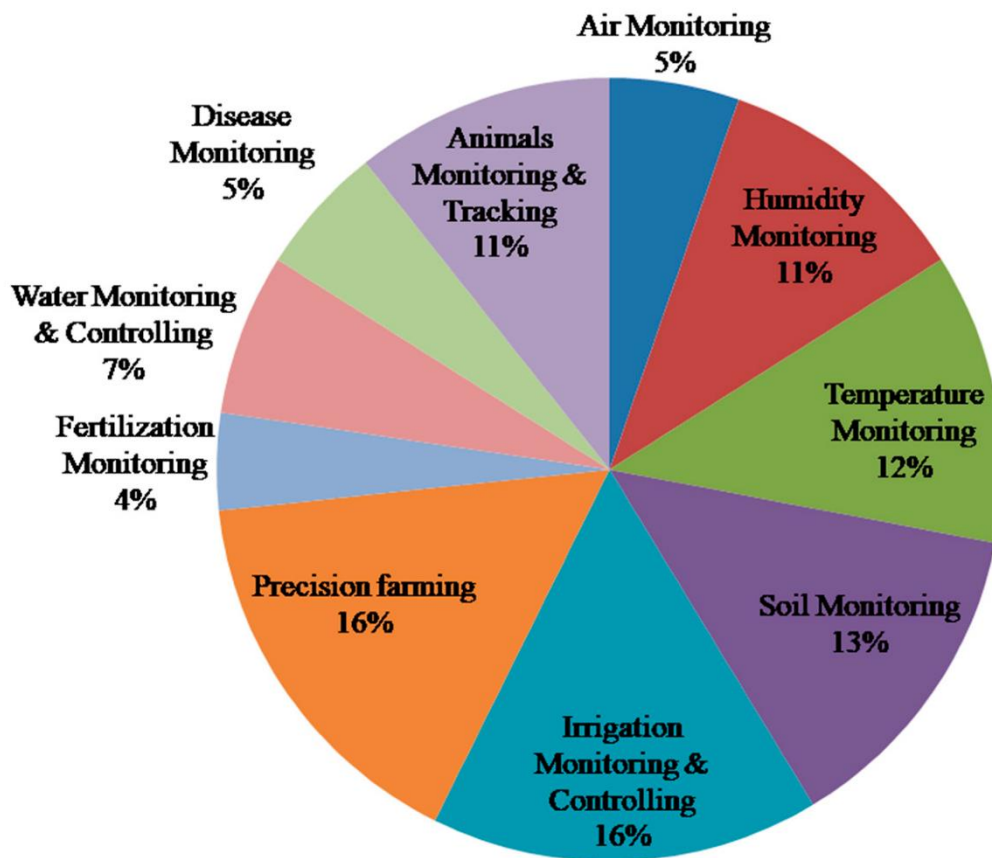


Figure 1. 4. Applications de l'IoT dans divers secteurs de l'agriculture intelligente (Deepak, 2022)

Les agriculteurs peuvent utiliser des capteurs pour augmenter les rendements et prévenir les dommages aux cultures. Les serres intelligentes sont un autre champ où l'IoT peut être appliqué où les systèmes hydroponiques à petite échelle sont inclus. Les serres intelligentes sont de plus en plus courantes dans les zones urbaines car elles permettent une meilleure surveillance (Deepak, 2022).

Plusieurs travaux dans la littérature ont été appliqués à l'IoT en agriculture afin d'améliorer la qualité et la quantité des produits agricoles. Le tableau suivant récapitule une étude de quelques travaux connexes.

Travail	Application / Objectif
(Veerpal, 2019)	Smart farming, Smart Greenhouse « Serre intelligente ».
(Negi, 2021)	Système d'agriculture intelligente utilisant les propriétés du sol.

Chapitre 1 : L'IoT : Architecture & Fonctionnement

(Bairam, 2022)	L'agriculture de précision.
(Fathy, 2023)	Un système d'irrigation sécurisé pour l'agriculture de précision utilisant le chiffrement rapide.
(Gaganpreet, 2023)	Explorer et comprendre les variables de prise de décision liées aux obstacles dans les pratiques d'agriculture de précision basées sur l'IoT.
(Shadi, 2023)	Développer un écosystème pour une gestion optimisée des cultures.
(Babita, 2023)	Optimiser l'irrigation à l'eau.

Tableau 1. 2. IoT en agriculture "étude des travaux connexes"

En résumé, l'IoT et ces technologies permettent une surveillance efficace des cultures à distance en fournissant des informations sur le climat, l'humidité, la température et la fertilité du sol...etc. Les agriculteurs peuvent connaître l'état de leur récolte à tout moment grâce à ces technologies, à tout moment et de n'importe où (Deepak, 2022).

7. Conclusion

Dans ce qui précède, nous avons présenté le concept de l'Internet des Objets, où nous avons exposé la définition la plus connue pour ce concept, l'architecture, le fonctionnement et les différentes technologies IoT. Dans la dernière section, nous avons parlé sur l'application de l'IoT en agriculture où elle montre son efficacité dans l'optimisation du processus agricole.

Chapitre 2 :

**L'intelligence artificielle et
ses sous-domaines**

**« clustering : unsupervised
learning »**

Chapitre 2 : L'intelligence artificielle et ses sous-domaines « Clustering : unsupervised learning »

1. Introduction

Généralement, un programme informatique est conçu pour résoudre un certain problème, par exemple un programme pour calculer les moyennes des étudiants et les classer. En revanche, il y a des problèmes où nous ne connaissons pas la solution exacte afin de la traduire sous forme d'un programme exécutable par la machine, nous pouvons donner un exemple par la reconnaissance automatique des symboles écrits à la main à partir d'une image scannée. Cependant, avec ce type de problème nous pouvons avoir des bases de données qui regroupent des instances sur le problème étudié (Ah-Pine, 2019), l'utilisation de ces données pour apprendre et faire des prédictions sur des nouvelles instances *c'est exactement l'objectif de l'intelligence artificielle*.

Dans ce chapitre, nous allons présenter l'intelligence artificielle et ses sous domaine où nous allons focaliser sur le clustering « apprentissage non-supervisé » et les principaux algorithmes de clustering pour terminer par l'application de ce dernier dans le domaine d'agriculture.

2. L'intelligence artificielle et ses sous domaines

Il existe plusieurs définitions pour le concept d'intelligence artificielle « IA ». D'après IBM², c'est la science et l'ingénierie qui vise à simuler le fonctionnement du cerveau humain, ou du moins sa logique lorsqu'il s'agit de prendre des décisions. En d'autre terme, L'IA est un ensemble d'algorithmes donnant à une machine la capacité d'analyser des nouvelles situations et faire des prédictions à partir de données déjà acquises. Usuellement, L'IA peut être divisée en plusieurs sous-domaines, notamment l'apprentissage automatique (Machine Learning en anglais), apprentissage profond (Deep Learning en anglais), voir (Figure 2.1).

² <https://www.ibm.com/>

Chapitre 2 : L'intelligence artificielle et ses sous-domaines « clustering : unsupervised learning »

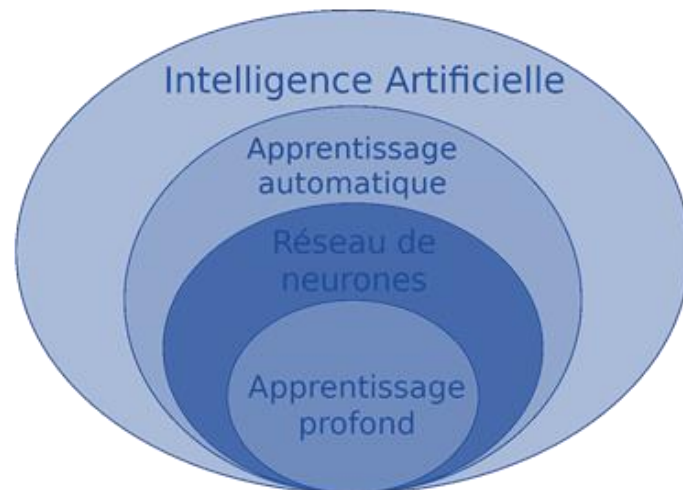


Figure 2. 1. Les sous domaines de l'intelligence artificielle.³

2.1. L'apprentissage automatique

L'apprentissage automatique « Machine Learning en anglais » est une branche de l'IA qui donne aux ordinateurs la capacité d'apprendre à partir des données existantes afin d'améliorer leurs performances à résoudre des tâches impossibles pour un être humain. La « machine » est l'algorithme **qui analyse des données volumineuses**, qui sont ingérables pour un être humain, afin de générer **des modèles prédictifs** qui facilitent la prise de décision dans des nouvelles situations (Ah-Pine, 2019) (Figure 2.2).

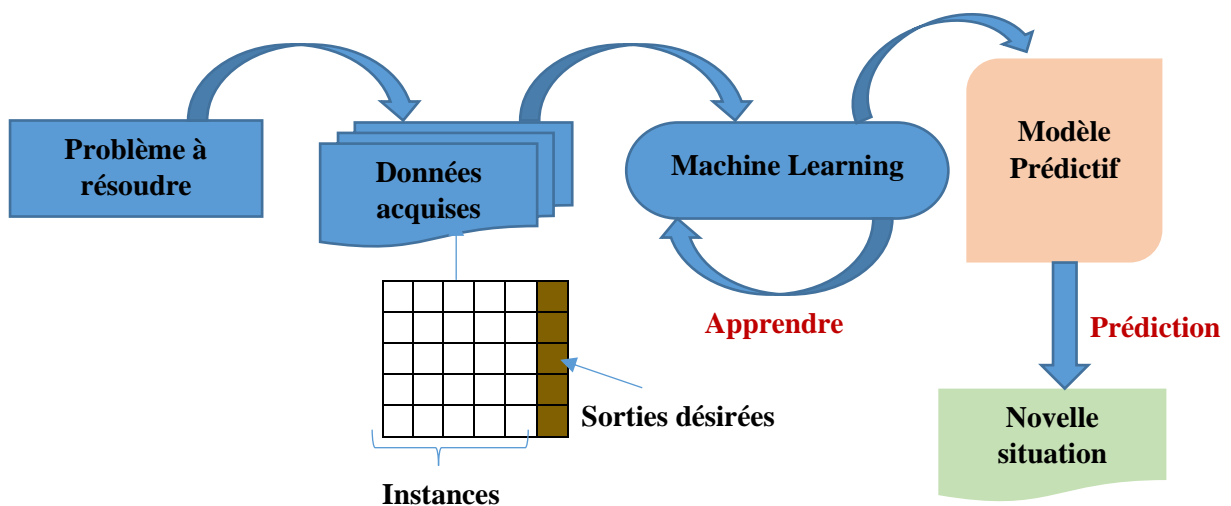


Figure 2. 2. L'apprentissage automatique en général

³ https://eduscol.education.fr/sti/si-ens-paris-saclay/ressources_pedagogiques/introduction-a-lapprentissage-automatique

Chapitre 2 : L'intelligence artificielle et ses sous-domaines « clustering : unsupervised learning »

Généralement, les algorithmes d'apprentissage automatique sont divisés en trois catégories principales : l'apprentissage supervisé, l'apprentissage non-supervisé et l'apprentissage par renforcement. Les deux premières branches (supervisé et non-supervisé) sont liées directement à la présence et l'absence de la sortie désirée dans l'ensemble de données du problème (Sid, 2018) (Sid, 2020) (Figure 2.3).

a. Apprentissage supervisé

L'apprentissage supervisé « Supervised Learning en anglais » comporte des variables d'entrée (X) et une variable de sortie (Y). Vous utilisez un algorithme pour apprendre la fonction de **mapping** de l'entrée à la sortie.

$$Y = f(X)$$

L'objectif est d'appréhender si bien la fonction de **mapping** que, lorsque vous avez de nouvelles données d'entrée (X), vous pouvez prédire immédiatement la sortie (Y).

La fonction de mapping, la machine ou l'algorithme utilise un ensemble de données de formation (training set) afin d'apprendre, le processus consiste à faire des prédictions itératives sur les données d'apprentissage et les corriger en comparant avec les sorties réelles jusqu'à que l'algorithme atteigne un niveau de performance prédictive acceptable (ISMAILI, 2019).

Avec l'apprentissage supervisé nous pouvons développer des modèles pour résoudre deux types de problèmes :

- Les problèmes de **Régression** : lorsque la variable de sortie est une catégorie.
- Les problèmes de **Classification** : lorsque la variable de sortie est une valeur réelle (ex. 0 ou 1).

b. Apprentissage non-supervisé

L'apprentissage non supervisé « Unsupervised Learning en anglais » ne dispose que de données d'entrée (X) et pas de variables de sortie correspondantes. L'objectif principal de l'apprentissage non supervisé est de modéliser la structure ou la distribution sous-jacente dans les données afin d'extraire l'information la plus utile. Dans ce type d'apprentissage, les algorithmes sont laissés à leurs propres mécanismes pour découvrir et présenter la structure intéressante des données (ISMAILI, 2019).

Chapitre 2 : L'intelligence artificielle et ses sous-domaines « clustering : unsupervised learning »

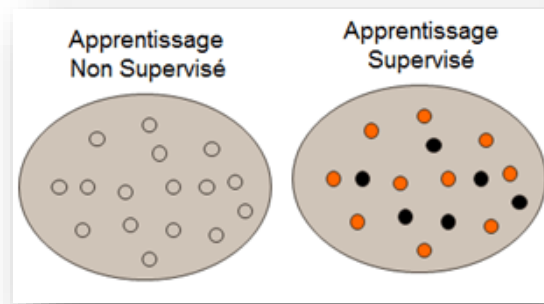


Figure 2. 3. Apprentissage supervisé vs. non-supervisé

c. L'apprentissage par renforcement

Dans l'apprentissage par renforcement, un agent est plongé au sein d'un environnement, et prend ses décisions en fonction de son état courant. En retour, l'environnement fournit à l'agent une récompense, qui peut être positive ou négative. L'agent cherche, à travers d'expériences itérées, un comportement décisionnel (appelé stratégie ou politique) optimal, en ce sens qu'il maximise la somme des récompenses au cours du temps (Dutech, 2003).

2.2.L'apprentissage profond

L'apprentissage automatique a été pendant une longue durée l'outil principal pour développer des modèles prédictifs dans de nombreux domaines de recherche. Mais avec l'augmentation très rapide de la taille des bases de données et la variété de ces données, l'apprentissage profond a été présenté pour surmonter ces défis (Sid, 2020) (Sid, 2022).

L'apprentissage profond « Deep Learning en anglais » est basé principalement sur le réseau de neurone classique « ANN », mais avec de nombreuses améliorations (Figure 2.4). Avec l'apprentissage profond, nous pouvons créer des modèles prédictifs plus profonds et plus larges avec plusieurs couches cachées d'unités de traitement non linéaires pour l'apprentissage des représentations de données (caractéristiques) directement à partir des données brutes avec une intervention minimale ou nulle de l'utilisateur, ce qui facilite l'adaptation du modèle résultant à de nouvelles instances (minimiser / éliminer le problème de sur-apprentissage) (Chen, 2018) (Sid, 2020).

Chapitre 2 : L'intelligence artificielle et ses sous-domaines « clustering : unsupervised learning »

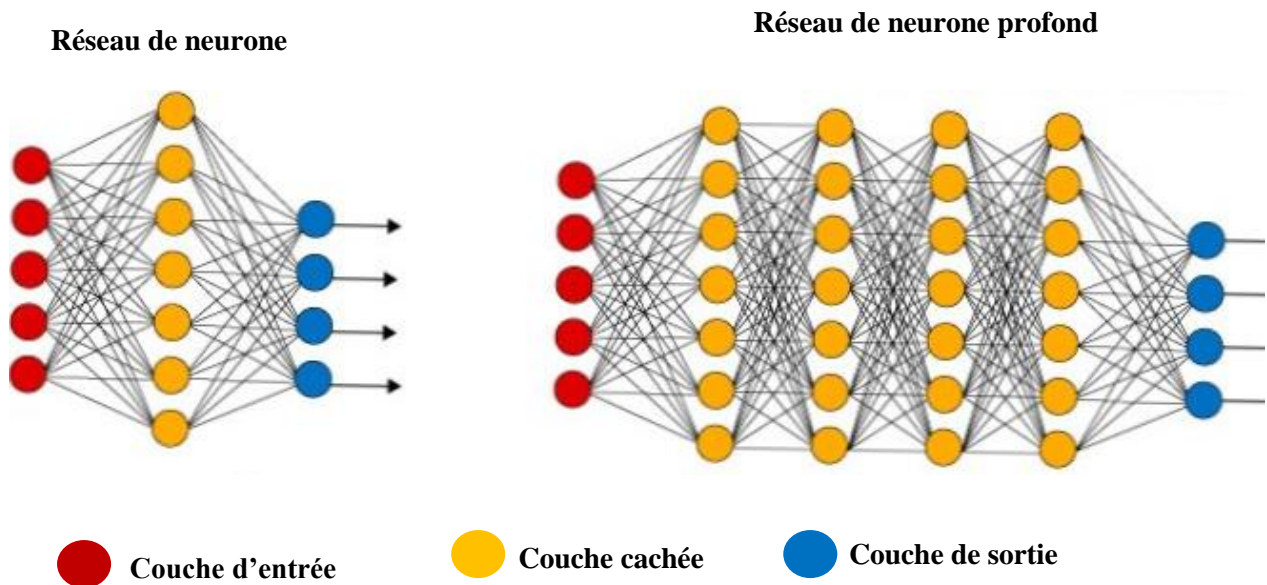


Figure 2. 4. Réseau de neurones VS. Réseau de neurones profond⁴

2.3.Data Science VS. Machine learning VS. Data Mining

Au cours de nos recherches, nous avons constaté qu'il y avait confusion dans les concepts « data science, data mining et machine learning » (Figure 2.5) et nous avons voulu les clarifier dans cette section (Merima, 2021).

« **Data Science** » est la discipline scientifique qui étudie tout ce qui concerne les données, depuis l'acquisition de données, le stockage de données, le nettoyage de données, l'analyse de données, la visualisation de données, l'interprétation de données, la prise de décisions, créer une valeur à partir de données et communiquer des informations.

La science des données est l'étude de l'extraction généralisable de connaissances à partir de données.

« **Data Mining** » exploration des données. Il sert à analyser des volumes énormes de données afin d'extraire des relations et liens entre celles-ci. L'objectif est de transformer les données en informations compréhensibles et exploitables.

« **Machine Learning** », l'objectif est de rendre une machine « intelligente » en lui soumettant un algorithme et des modèles de données afin d'automatiser des tâches complexes.

⁴ https://fr.differbetween.com/article/difference_between_neural_network_and_deep_learning

Chapitre 2 : L'intelligence artificielle et ses sous-domaines « clustering : unsupervised learning »

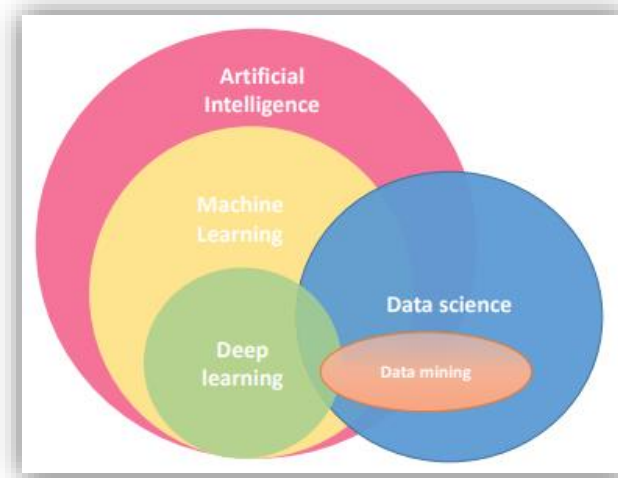


Figure 2. 5. Data Science VS. Machine learning VS. Data Mining

3. Apprentissage non supervisé et le regroupement de données

3.1. Le clustering : définition

Le processus d'identification des éléments proches les uns des autres et distincts des entités des autres, c'est exactement le regroupement ou le partitionnement, largement utilisé dans l'analyse des données (Neha, 2020).

Le clustering est une méthode d'apprentissage non-supervisé qui consiste à regrouper des points de données **par similarité** ou **par distance** en respectant la règle « similarité maximale au sein du groupe et dissimilarité maximale entre les groupes ». Pour un ensemble donné de points, vous pouvez utiliser un algorithme de clustering pour classer ces points de données individuels dans des groupes spécifiques (généralement le nombre de groupe est donné comme un paramètre d'entrée), voir (Figure 2.6). En conséquence, les points de données d'un groupe particulier présentent des propriétés similaires, en même temps, les points de données de différents groupes ont des caractéristiques différentes (Hausmane, 2019).

Chapitre 2 : L'intelligence artificielle et ses sous-domaines « clustering : unsupervised learning »

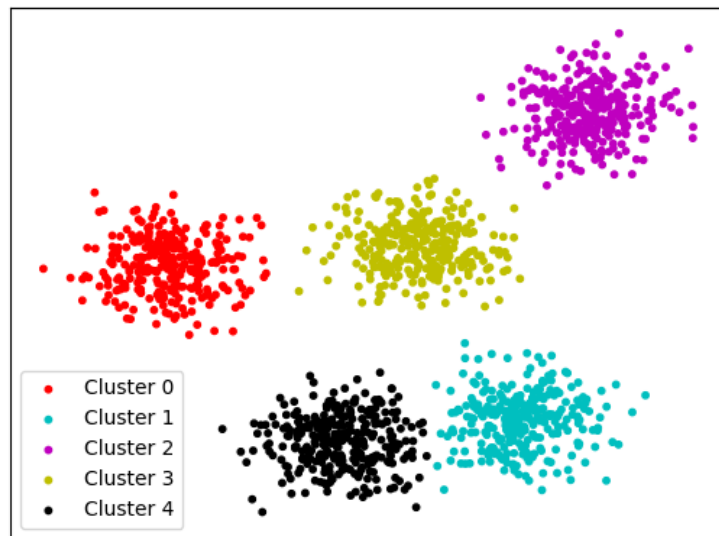


Figure 2. 6. Exemple de clustering sous 5 groupes

3.2. Les objectifs du clustering & types

a. Le Principe & les objectifs

Comme nous avons dit le clustering ou le partitionnement des données est une méthode non supervisée rassemble un ensemble d'algorithmes d'apprentissage dont le but est de regrouper entre elles des données non étiquetées (n'existent pas les sorties désirés Y) qui présentes des propriétés similaires.

Le résultat d'un algorithme de clustering peut être utilisé pour préparer le terrain à une application ultérieure d'un algorithme d'apprentissage supervisé « classification ». Nous pouvons résumer les principaux objectifs du clustering dans les quatre points suivants (Henry, 2022):

- Découvrir des structures et des modèles dans des données à grande échelle.
- Regrouper les données avec des patterns similaires ensemble.
- Cela réduit la complexité et facilite l'interprétation des relations entre les données.
- Parfois, le partitionnement est le but, exemple : dans la segmentation des images où l'objectif principal est segmenté l'image sous forme des sections (groupes).

Chapitre 2 : L'intelligence artificielle et ses sous-domaines « clustering : unsupervised learning »

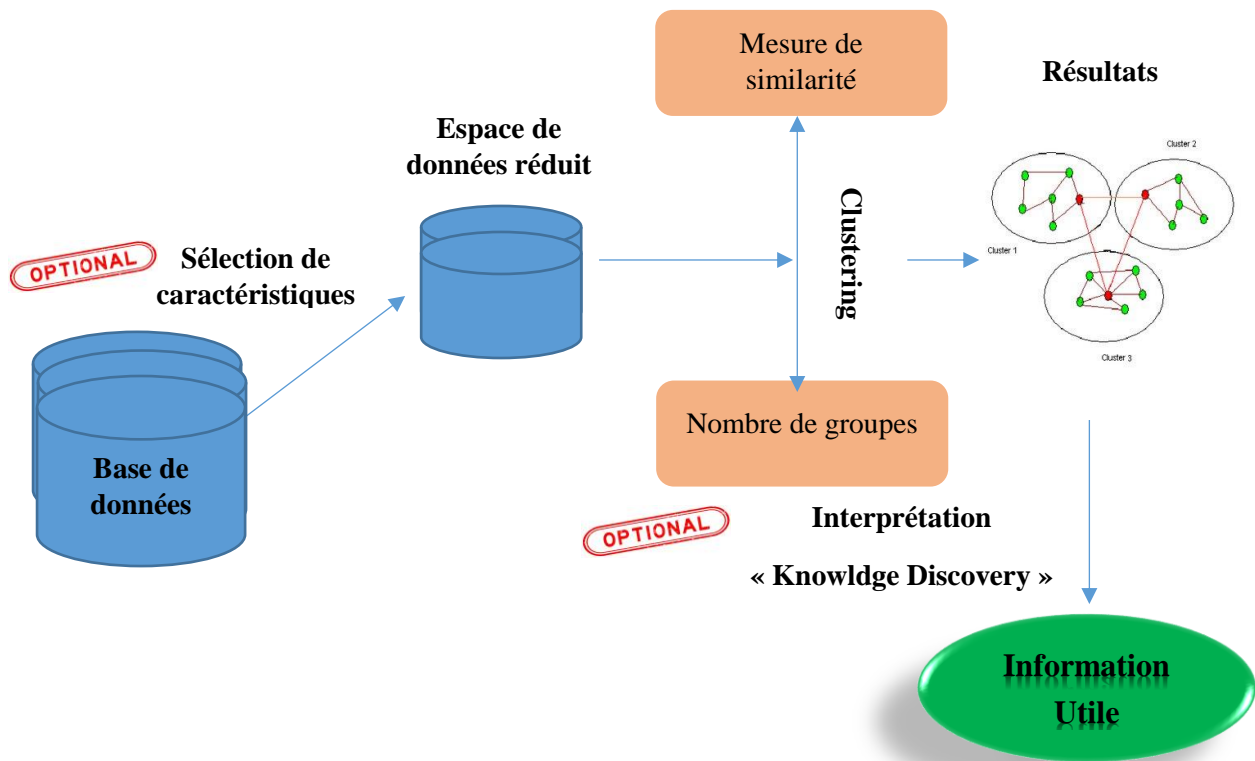


Figure 2. 7. Le processus général de clustering

b. Catégories d'algorithmes de clustering

Une multitude de méthodes de clustering ont été proposées dans la littérature. Les algorithmes de clustering peuvent être classés selon trois critères (Halkidi, 2001) :

- Le type de données d'entrée de l'algorithme.
- Le critère de clustering qui définit la similarité entre les points de données.
- La théorie et les concepts fondamentaux sur lesquels reposent les techniques d'analyse de clustering.

Généralement, les algorithmes de clustering sont classés en deux catégories (Figure 2.8) :

- *Algorithmes partitionnels* : générer différentes partitions puis évaluer-les selon certains critères ;
- *Algorithmes hiérarchiques* : créer une décomposition hiérarchique de l'ensemble des objets en utilisant un certain critère.

Chapitre 2 : L'intelligence artificielle et ses sous-domaines « clustering : unsupervised learning »

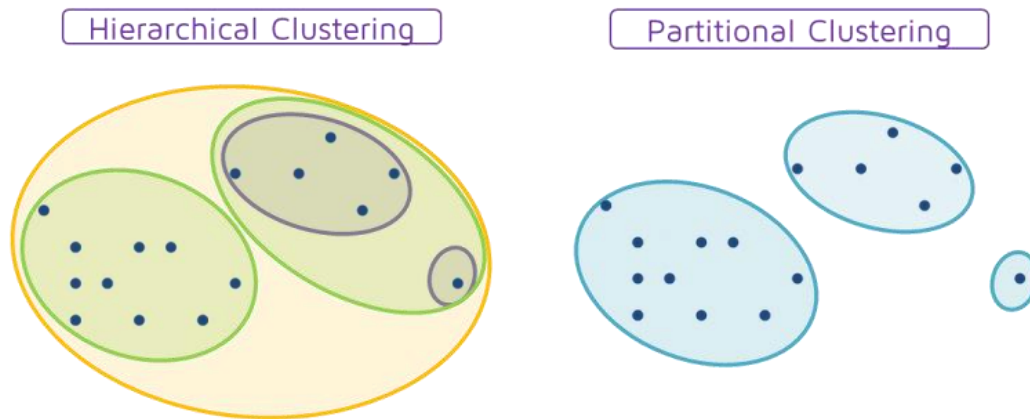


Figure 2. 8. Clustering Hiérarchique VS. Clustering partitionnel⁵

c. Mesure de proximité

Le terme proximité entre deux objets est une fonction qui mesure la proximité des caractéristiques correspondantes de ces objets. Les mesures de proximité font référence aux mesures de similarité et de dissimilarité (Tarun, 2019).

La similarité est une mesure numérique du degré de ressemblance entre les deux objets. Plus élevée pour une paire d'objets qui se ressemblent mieux. Généralement, une valeur non négative et comprise entre 0 et 1. Supposons que, $Sim(x, y)$ est la similarité entre les points x et y , alors les propriétés typiques des similarités sont les suivantes (Tarun, 2019) :

1. $Sim(x, y) = 1$ seulement si $x = y$. ($0 \leq Sim \leq 1$).
2. $Sim(x, y) = Sim(y, x)$ pour tout x et y .

Il existe plusieurs mesures de similarité dans la littérature, nous citons : Coefficient de correspondance simple, Coefficient de Jaccard, similarité du cosinus et la corrélation (Tarun, 2019).

3.3. Les algorithmes de clustering

Dans cette section, nous allons présenter quelques algorithmes de clustering les plus utilisés dans le domaine d'agriculture.

a. L'algorithme K-means

K-means est un algorithme partitionnel qui vise à générer K clusters (k paramètre donné par l'utilisateur) en partitionnant les données. Dans l'étape initiale, l'algorithme choisit

⁵ <https://quantdare.com/hierarchical-clustering/>

Chapitre 2 : L'intelligence artificielle et ses sous-domaines « clustering : unsupervised learning »

arbitrairement les centres des clusters. Puis, il distribue les éléments restants entre les clusters, en utilisant une mesure de proximité, et modifie le centre du cluster. Les étapes de l'algorithme sont résumées dans les points suivants (Vandana, 2019) :

1. **Donner une valeur pour K, le nombre de groupes ;**
2. Initialiser les centres des K clusters (aléatoire, si nécessaire) ;
3. Décider les appartenances des N objets en les affectant au centre de cluster le plus proche => ex. via une mesure de similarité ;
4. Recalculer les centres des K clusters, en supposant que les appartenances trouvées ci-dessus sont correctes ;
5. Répéter 3 et 4 jusqu'à ce qu'aucun des N objets ne change d'appartenance lors de la dernière itération.

b. *Fuzzy C-means (FCM)*

C'est une méthode qui se base sur *la logique floue*, consiste à regrouper des données multidimensionnelles, en attribuant à chaque point un degré d'appartenance à chaque cluster. Cela peut être très puissant par rapport au clustering traditionnel à seuil strict comme le K-means. Cet algorithme fonctionne en donnant une appartenance à chaque point de données correspondant à chaque cluster selon la distance entre le centre du cluster et le point de données. Plus les données sont proches du centre de cluster, plus leur degré d'appartenance est grand. La somme des membres de chaque point de données doit être égale à 1. Le processus de fuzzy c-means est résumé dans les points suivants (Aman, 2021):

1. **Supposons un nombre fixe de clusters K ;**
2. Initialiser aléatoirement les k-centres associés aux clusters et calculer la probabilité que chaque point de données soit membre d'un cluster donné k ;
3. Recalculer le centre du cluster en tant que centre pondéré étant donné les probabilités d'appartenance de tous les points de données ;
4. Itérer jusqu'à la convergence ou jusqu'à ce qu'un nombre d'itérations spécifié par l'utilisateur ait été atteint.

Le tableau ci-dessous (Tableau 2.1) résume une petite comparaison entre les deux algorithmes de clustering K-means et Fuzzy C-means (Aman, 2021).

Chapitre 2 : L'intelligence artificielle et ses sous-domaines « clustering : unsupervised learning »

Critère / Algorithme	K-means	Fuzzy C-means
Attribution à un cluster	Chaque point appartenant complètement à un seul cluster.	Chaque point a une probabilité d'appartenance à chaque cluster.
Rapidité	Rapide.	Lent.
Performance	Moins performant.	Plus performant.

Tableau 2. 1. K-Means VS. Fuzzy C-Means

c. Clustering hiérarchique

Le clustering hiérarchique représenté par une structure arborescente appelée dendrogramme. Il affiche les relations « cluster <-> sous-cluster ». Deux approches de base dans le clustering hiérarchique (Figure 2.9) sont (Vandana, 2019):

→ **Approche par division « Top-Down »** : Commence par un cluster, à chaque étape, le cluster est divisé jusqu'à la formation d'un cluster individuel. Elle est également connue sous le nom d'approche descendante. L'algorithme de clustering basé sur l'approche par division est décrit comme suit (Vandana, 2019):

1. Commencer avec l'échantillon entier comme un seul cluster ;
2. Diviser en deux sous-clusters ;
3. Répéter le processus jusqu'à ce que chaque donnée se trouve dans son propre cluster individuel.

→ **Approche agglomératif « Bottom-up »**:

Chaque point est considéré comme un cluster individuel à chaque étape, la paire de clusters la plus proche est fusionnée. Il suit une approche ascendante. L'algorithme basé sur le clustering agglomératif hiérarchique est décrit comme suit (Vandana, 2019):

1. Commencer avec n clusters car chaque élément est considéré comme un cluster ;
2. Les deux clusters les plus proches sont fusionnés en un seul cluster ;
3. A chaque étape, les deux clusters ayant la plus petite distance sont fusionnés.

Chapitre 2 : L'intelligence artificielle et ses sous-domaines « clustering : unsupervised learning »

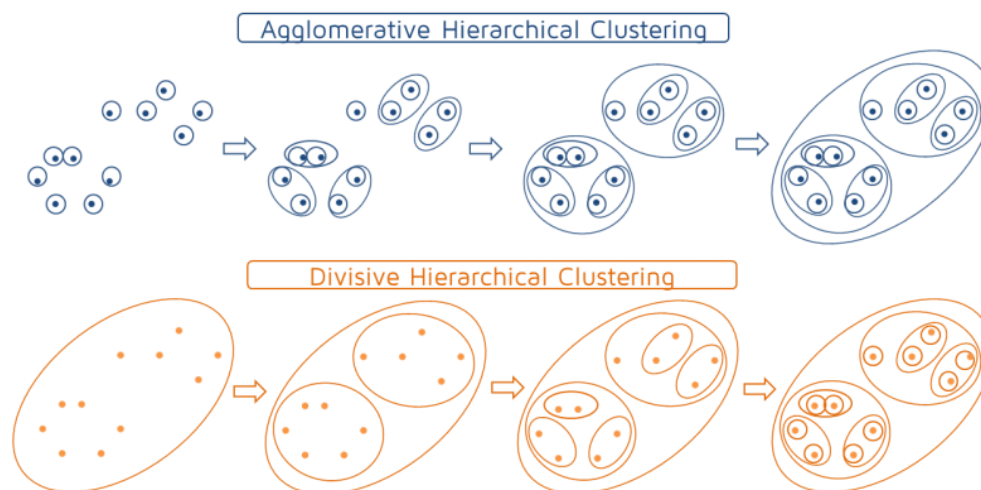


Figure 2. 9. Approche Bottom-up VS. Approche Top-Down⁶

Le tableau suivant (Tableau 2.2) récapitule les avantages et les inconvénients pour les deux catégories de clustering partitionnel et hiérarchique (Henry, 2022).

Types	Avantages	Inconvénients
Clustering partitionnel « ex. K-means »	<ul style="list-style-type: none"> • Facile à implémenter ; • Fonction objective intuitive : optimise la similarité intra-cluster ; • Relativement efficace : $O(nk)$, où n est # objets, k est # clusters, et t est # itérations ; Normalement, $k, t \ll n$. 	<ul style="list-style-type: none"> • Demande le nombre de clusters à l'avance (limité) ; • Se termine souvent à un optimum local ; • L'initialisation est importante ; • Incapable de gérer les données bruyantes et les valeurs aberrantes.
Clustering hiérarchique	<ul style="list-style-type: none"> • Pas besoin de préciser le nombre de clusters ; • Correspond bien à l'intuition humaine pour certains domaines. 	<ul style="list-style-type: none"> • Ils évoluent mal : complexité temporelle d'au moins $O(n^2)$, où n est le nombre total d'objets ; • Comme tout algorithme de recherche heuristique, le problème d'optimum local.

Tableau 2. 2. Comparaison entre le clustering partitionnel & hiérarchique

⁶ <https://quantdare.com/hierarchical-clustering/>

Chapitre 2 : L'intelligence artificielle et ses sous-domaines « clustering : unsupervised learning »

4. Le clustering dans le domaine d'agriculture

Dans le domaine d'agriculture, le clustering peut être utilisé pour agréger et analyser des données liées à des caractéristiques climatiques telles que la température, l'humidité, la vitesse et la direction du vent.

L'idée générale consiste à collecter des données sur ces variables climatiques dans les différentes régions agricoles dans lesquelles les plantes sont cultivées. Ensuite, nous pouvons appliquer des algorithmes de clustering afin de regrouper les données en fonction des conditions dans lesquelles les plantes ont besoin d'eau. Le résultat d'un regroupement permet de déterminer les bonnes conditions climatiques d'irrigation, ce qui aide les agriculteurs à mieux gérer l'irrigation des cultures et à optimiser l'utilisation des ressources en eau.

Le clustering a été largement appliqué en agriculture (Jiang, 2011) (Dileep, 2015) (Jharna, 2017) (Manjula, 2017) (Nevena, 2019) (Vandana, 2019) (Sabir, 2020) (Boris, 2021) (Asif, 2023), dans le tableau suivant (Tableau 2.3), nous allons récapituler quelques travaux connexes en précisant les méthodes utilisés et l'objectif d'application.

Travail	Méthodes utilisés	Objectif de clustering
(Jiang, 2011)	Fuzzy C-means	Définition des zones de gestion de l'irrigation.
(Dileep, 2015)	K-Means, clustering hiérarchique	Analyse de données agricoles.
(Manjula, 2017)	K-means	Regroupement des régions en fonction du niveau d'eau (à utiliser pour la prévision du rendement).
(Nevena, 2019)	K-means	Regroupement de données multivariées et corrélées dans une application d'analyse agricole, qui utilise les mesures EC du sol « conductivité électrique du sol » (ainsi que l'élévation) pour identifier les régions similaires.
(Vandana, 2019)	K-means hybride	Extraire les connaissances en agriculture de précision : identifier les données sur le niveau d'humidité du sol et les conditions climatiques liées aux cultures telles que les

Chapitre 2 : L'intelligence artificielle et ses sous-domaines « clustering : unsupervised learning »

		statistiques de température, d'humidité et de précipitations.
(Asif, 2023)	Clustering hiérarchique	Analyser les différences régionales dans l'effet des politiques publiques alternatives sur le secteur agricole.

Tableau 2. 3. Clustering en agriculture « étude des travaux connexes »

5. Conclusion

Dans ce chapitre, nous avons présenté le concept d'intelligence artificielle et ses sous domaines où nous avons focalisé sur l'apprentissage automatique non-supervisé « clustering ». Nous avons exposé le principe général de clustering en précisant les différentes catégories avec une comparaison qui récapitule les avantages et les inconvénients de chacune. Nous avons achevé le chapitre par une petite étude des travaux connexes « l'application de clustering dans le domaine d'agriculture ». Dans ce qui suit, nous allons présenter notre approche proposée pour grouper des données agricoles.

Chapitre 3 :

**Conception : Approche
proposée**

Chapitre 3 : L'approche proposée « Conception »

1. Introduction

Dans ce chapitre, nous allons présenter notre approche proposée pour grouper des données agricoles captées dans un environnement IoT et extraire les informations les plus utiles. Premièrement, nous exposerons le processus de génération des données d'évaluation en utilisant un simulateur spécialisé « TinkerCad », où nous allons trouver quelques limitations, ce qui nous a fait utiliser une base alternative. Par la suite, nous découvrirons le schéma général de l'approche en décrivant l'algorithme de clustering proposé.

2. La simulation des données : Simulateur TinkerCad

Dans cette section, nous allons découvrir le simulateur utilisé pour simuler et générer des données pertinentes.

2.1. Description

TinkerCad est un programme de modélisation 3D gratuit connu par sa facilité d'utilisation. Il est 100% basé sur le Web, ce qui le rend accessible à toute personne disposant d'une connexion Internet (AutoDeskTinkerCAD, 2023).

TinkerCad Arduino offre aux utilisateurs un environnement de développement complet pour concevoir et simuler des circuits électroniques à l'aide d'Arduino, en combinant la sélection de composants, la programmation et la simulation via un environnement de développement intégré (IDE) ⁷. Nous pouvons résumer le processus général de la création d'un circuit sous TinkerCad dans les points suivants (Figure 3.1) :

- **Création de projet** : Premièrement, l'utilisateur doit créer un nouveau projet et nommer le circuit électronique qu'il souhaite concevoir.
- **Sélection des composants** : Ensuite, l'utilisateur peut sélectionner les composants électroniques qu'il veut les utiliser, tels que des résistances, des diodes, des LED, des

⁷ <https://www.tinkercad.com/>

Chapitre 3 : L'Approche proposée « Conception »

capteurs, etc. Les composants peuvent être choisis à partir de la bibliothèque de composants TinkerCad ou ajoutés à partir de la bibliothèque en ligne d'Arduino.

- **Connexion des composants** : L'utilisateur peut faire glisser et déposer les composants sur le plan de travail et les connecter en utilisant des fils. Les fils peuvent être connectés aux broches des composants en utilisant des connexions virtuelles.
- **Programmation** : Une fois que les composants sont connectés, l'utilisateur peut ajouter le code à son projet en utilisant l'IDE Arduino intégré. Le code peut être écrit en utilisant le langage de programmation Arduino basé sur C/C++.
- **Simulation** : Après la conception et la programmation du circuit électronique, l'utilisateur peut simuler le fonctionnement de son circuit en utilisant la fonction de simulation de TinkerCad. La simulation permet à l'utilisateur de voir comment les différents composants interagissent et comment le code fonctionne.

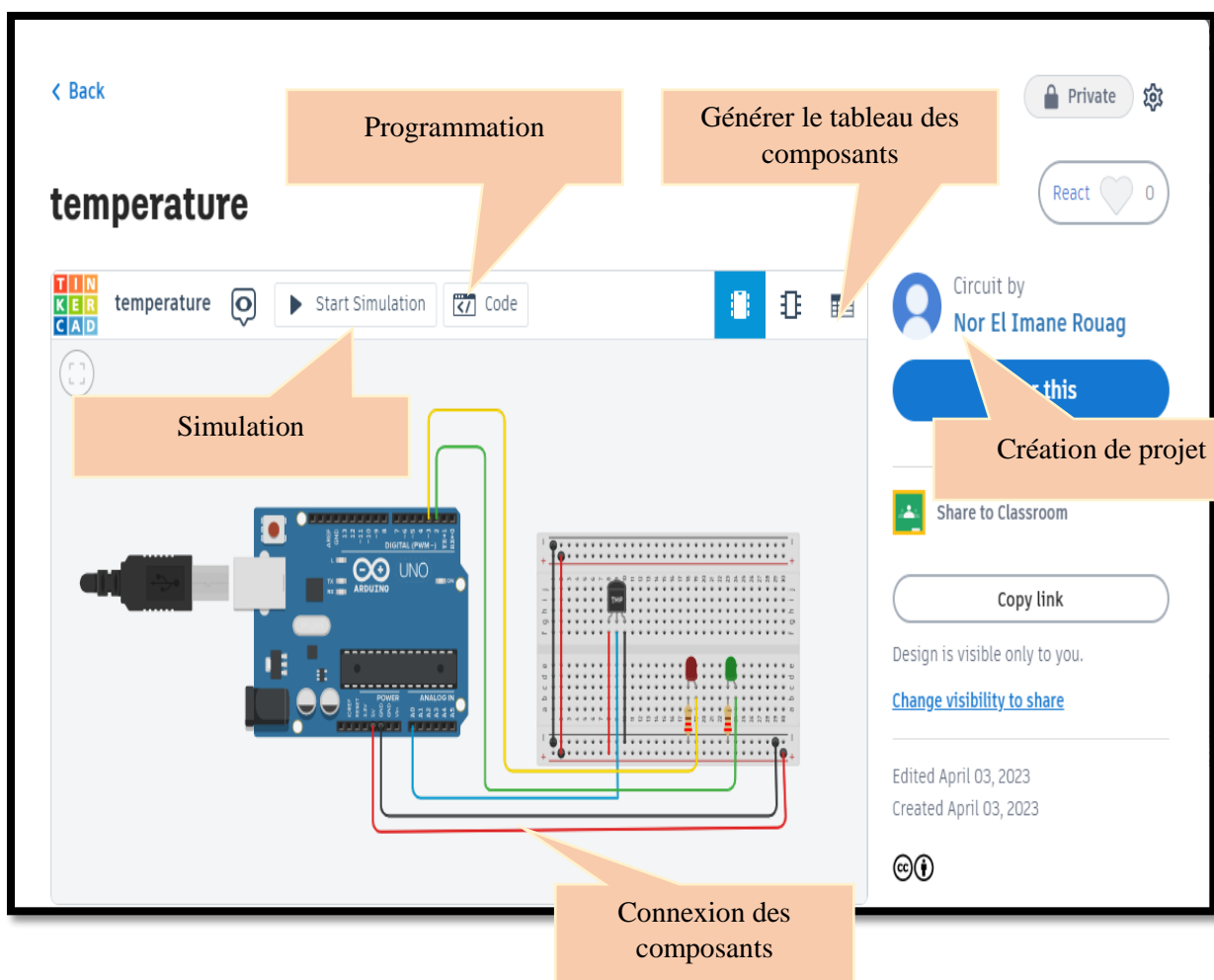


Figure 3. 1. La création d'un circuit sous TinkerCad

Chapitre 3 : L'Approche proposée « Conception »

2.2.Arduino & capteurs

Le système Arduino offre la possibilité de lier les performances de la programmation à celles de l'électronique, autrement dit, pour programmer des systèmes électroniques. Le plus grand avantage de l'électronique programmée qu'elle peut simplifier les schémas électroniques et par conséquent, optimiser le coût de la réalisation et la charge du travail à la conception d'une carte électronique (Astalaseven, 2012).



Figure 3. 2. La carte Arduino (entrées-composants-sorties)(Fréou, 2023)

Une carte Arduino (Type Uno) est une petite avec les dimensions 5,33 x 6,86 cm, équipée d'un microcontrôleur (Figure 3.2). Le microcontrôleur permet, à partir d'événements captés par des capteurs, de programmer et commander des actionneurs ; alors, la carte Arduino peut définir comme une interface programmable. La figure ci-dessous (Figure 3.3) illustre quelques types des cartes Arduino les plus utilisées (Fréou, 2023).

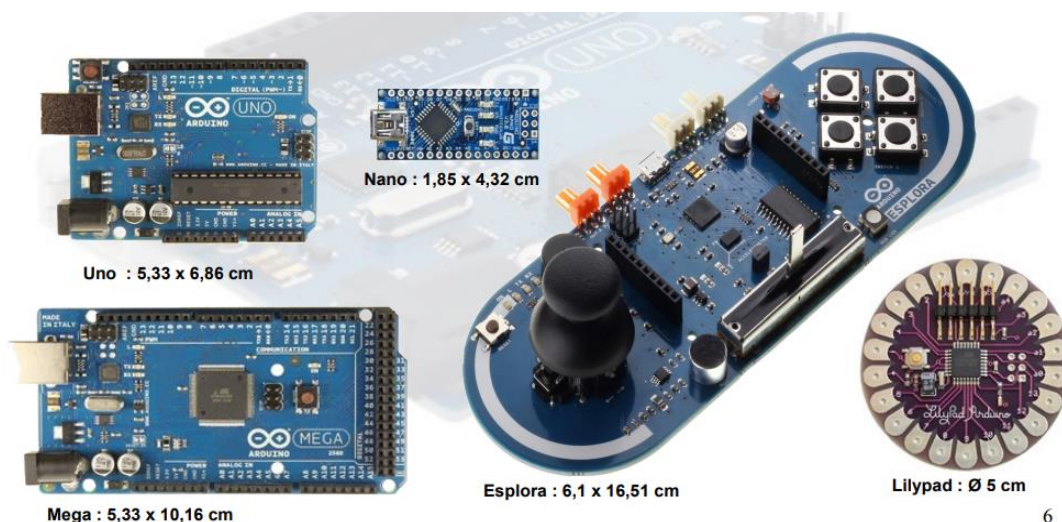


Figure 3. 3. Les types de cartes Arduino (Guy, 2023)

Les capteurs sont des objets électroniques de taille très réduite avec des ressources limitées, autonomes, capable de traiter des informations et de les transmettre, via les ondes radio, à une autre entité, ex. capteurs, unité de traitements... sur une distance limitée à quelques mètres. La figure (Figure 3.4) illustre les composants physiques d'un capteur en précisant les deux parties : le système de localisation et le mobilisateur (KAMBOUCHE, 2018).

Chapitre 3 : L'Approche proposée « Conception »

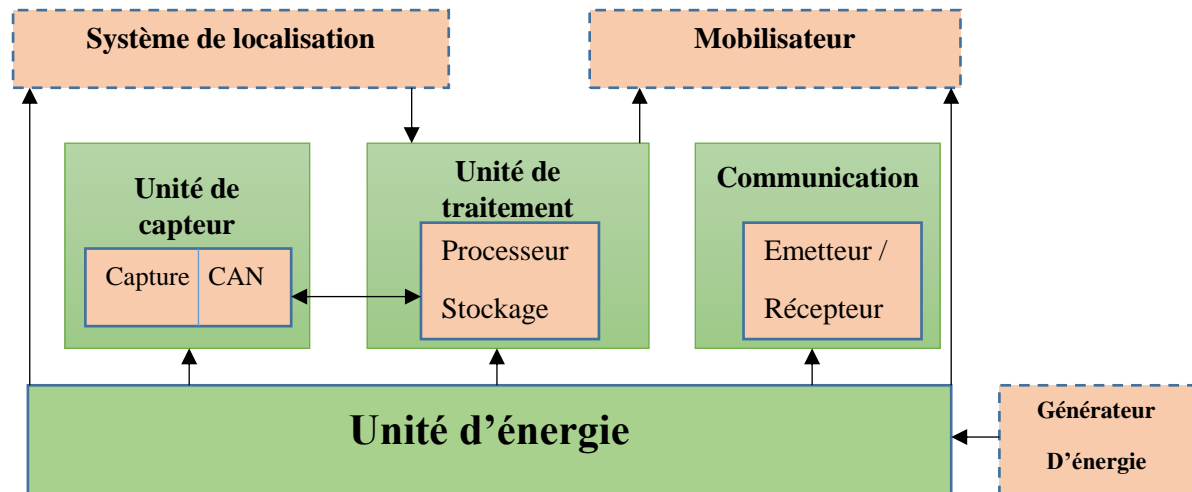


Figure 3. 4. La structure d'un capteur (KAMBOUCHE, 2018)

- *Unité de capture* : c'est l'unité de détection, usuellement composée de deux modules : le capteur, et un convertisseur analogique numérique. Le capteur a le rôle de communiquer l'information captée depuis son environnement sous forme d'un signal analogique.
- *Unité de traitement* : comprend un processeur associé à une unité de stockage et fonctionne à l'aide d'un système d'exploitation spécialisé conçu pour les microcapteurs. Cette unité gère également les procédures et les protocoles permettant la communication entre les nœuds.
- *Unité de communication* : gère toutes les émissions/réceptions au sein d'un capteur.
- *Unité d'énergie* : la gestion d'énergie est un point très important dans les réseaux de capteurs. Les batteries utilisées sont soit rechargeable ou non. Généralement, dans un environnement sensible, il est impossible de recharger ou changer une batterie, donc il faut avoir une meilleure gestion de la consommation d'énergie pour augmenter la durée de vie du réseau.

3. Génération de la base de données

3.1. En utilisant le simulateur TinkerCad

L'irrigation est un facteur important pour la culture sous serre. Mal fait, cela peut ruiner vos cultures et mettre en péril la santé de la serre de votre jardin. Donc, de nombreux critères doivent être pris en compte pour garantir la bonne irrigation dans le bon moment. Dans ce contexte, nous pouvons dire que les principaux critères (caractéristiques) sont : la température, l'humidité, la vitesse du vent et l'orientation. Afin de générer une base de données agricole via le simulateur TinkerCad, nous avons commencé par l'utilisation d'un seul capteur qui vise le critère de la température, comme un critère majeur. Les étapes de la simulation sont résumées dans les points suivants :

Chapitre 3 : L'Approche proposée « Conception »

→ La génération de la base de données sous TinkerCad commence par la création de circuits électroniques en utilisant les composants électroniques virtuels disponibles sur la plateforme. Nous avons utilisé plusieurs composants afin d'obtenir un circuit qui peut simuler des données liées à la température. Ces derniers sont récapitulés dans le tableau suivant (Tableau 3.1):

Name	Quantity	Component
U1	1	Arduino Uno R3
D1	1	Red LED
D2	1	Green LED
R1 R2	2	220 Ω Resistor
U2	1	Temperature Sensor [TMP36]

Tableau 3.1. Les composants du circuit

→ Par la suite, nous pouvons ouvrir le TinkerCad et accéder à la bibliothèque de composants électroniques (tableau précédent) afin d'obtenir la structure suivante :

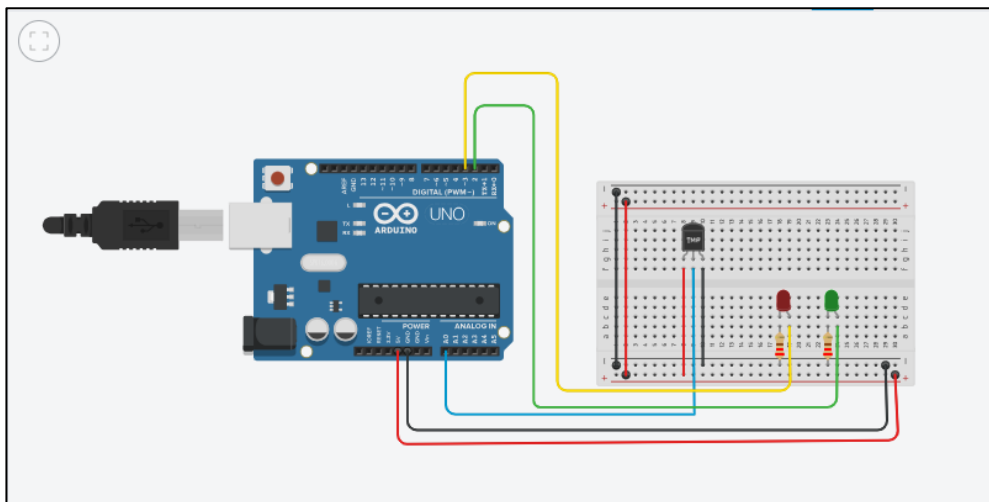
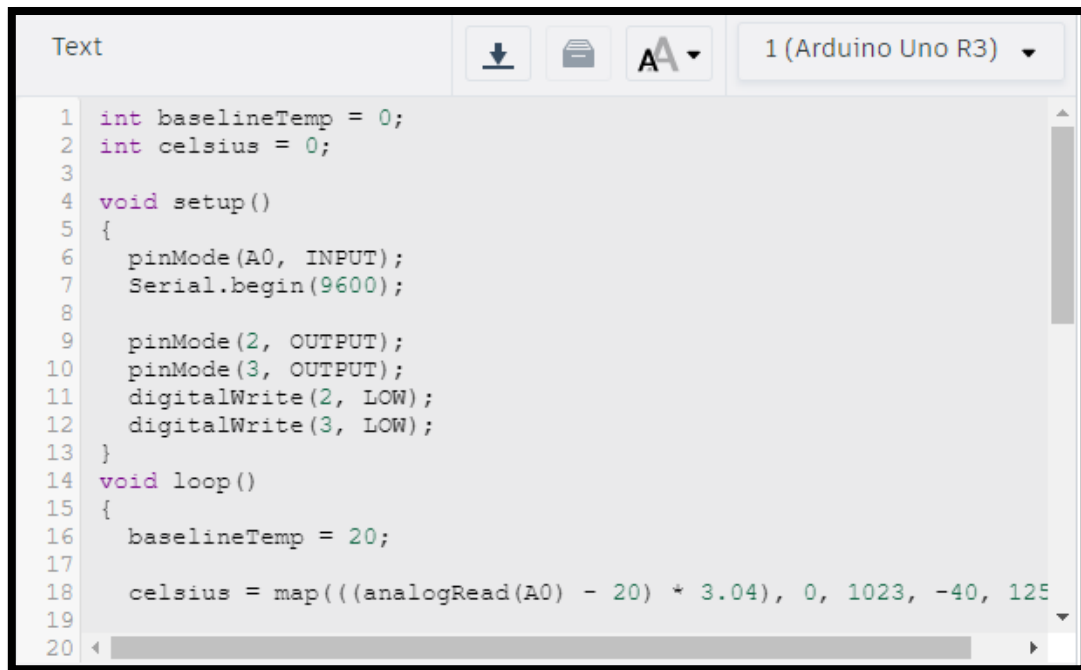


Figure 3. 5. Le circuit obtenu

→ La génération du circuit fournit immédiatement le code suivant (Figure 3.6)

Chapitre 3 : L'Approche proposée « Conception »



```
Text [Download] [Save] [Font] 1 (Arduino Uno R3)
1 int baselineTemp = 0;
2 int celsius = 0;
3
4 void setup()
5 {
6   pinMode(A0, INPUT);
7   Serial.begin(9600);
8
9   pinMode(2, OUTPUT);
10  pinMode(3, OUTPUT);
11  digitalWrite(2, LOW);
12  digitalWrite(3, LOW);
13 }
14 void loop()
15 {
16   baselineTemp = 20;
17
18   celsius = map((analogRead(A0) - 20) * 3.04), 0, 1023, -40, 125);
19
20
```

Figure 3. 6. Code source pour la génération du circuit

→ Enfin, nous avons exécuté la simulation pour surveiller et visualiser les valeurs de la température en temps réel, le schéma suivant illustre une petite démonstration des résultats obtenus (figure 3.7) :

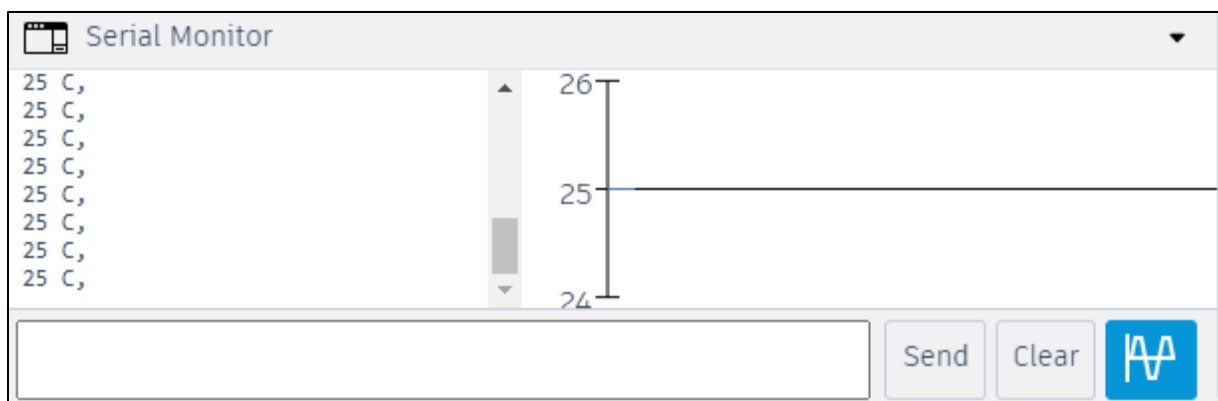


Figure 3. 7. Résultat de la simulation

D'après la figure au-dessus, nous pouvons remarquer que nous avons obtenu une seule valeur de la température (25°C) à cause de la durée de test, ce sont des valeurs générées dans **≈ 30 minutes**, et c'est la contrainte principale. Comme nous avons dit précédemment, le TinkerCad est un outil populaire et facile à utiliser pour créer des différents modèles et simuler des données dans un domaine particulier. En revanche, dans le domaine d'agriculture, la simulation de données afin d'avoir une large quantité de données pertinentes nécessite un temps


Chapitre 3 : L'Approche proposée « Conception »

important (jusqu'à des mois, à différents moments de la journée). A cette raison, nous avons utilisé une base de données alternative dans le même contexte.

3.2. Une base de données alternative

La base de données que nous avons utilisée pour évaluer notre approche est extraite à partir de la plateforme Meteoblue⁸, Meteoblue permet de fournir des données météorologiques et climatiques brutes et offre des données les plus précises du marché, adaptées aux exigences spécifiques de divers cas d'utilisation. La base de données est constituée de 10825 échantillons pris en se basant sur les 4 critères (caractéristiques) : la température (°C), l'humidité (%), la vitesse du vent (km/h) et la direction du vent (°). Les données ont capté entre 1/1/2022 (00 :00) et 29/3/2023 (1 :00), chaque jour, et à des moments différents (Figure 3.8).

La même journée, à des moments différents



timestamp	Temperature	Humidity	Speed	Direction
2022-01-01T00:00:00	6,4902453	93,82894	5,0911684	188,1301
2022-01-01T01:00:00	6,0602455	94,19782	3,9763298	174,80557
2022-01-01T02:00:00	5,5602455	94,83262	4,32	180
2022-01-01T03:00:00	4,630245	96,47203	3,2599385	173,6598
2022-01-01T04:00:00	3,6602454	97,1187	3,5455887	156,0375
2022-01-01T05:00:00	3,8802452	96,16575	3,671294	168,69006
2022-01-01T06:00:00	3,6502452	96,980194	4,3498964	155,55605
2022-01-01T07:00:00	3,1102452	97,79321	5,1919937	146,30994
2022-01-01T08:00:00	2,7302456	98,201126	5,3999996	143,1301
2022-01-01T09:00:00	2,5102453	98,19411	5,3517847	137,72632
2022-01-01T10:00:00	2,4402454	97,83999	5,495161	148,3925
2022-01-01T11:00:00	4,2502456	95,36762	6,638072	167,47119
2022-01-01T12:00:00	6,5302453	92,20724	5,8048253	150,25511
2022-01-01T13:00:00	8,290245	88,76688	6,439876	153,43495
2022-01-01T14:00:00	10,090245	83,10312	5,95906	154,98311
2022-01-01T15:00:00	11,3102455	79,97462	5,991594	147,26477
2022-01-01T16:00:00	12,100245	77,46256	5,692099	145,30484
2022-01-01T17:00:00	10,530245	84,3071	6,28713	156,37062
2022-01-01T18:00:00	7,3002453	93,68196	5,6920996	161,56505
2022-01-01T19:00:00	5,9902453	95,78908	6,0347	162,64597
2022-01-01T20:00:00	5,8002453	95,445526	5,937272	165,96376
2022-01-01T21:00:00	5,170245	94,87827	5,315336	151,69923
2022-01-01T22:00:00	4,4902453	95,85379	5,495161	148,3925
2022-01-01T23:00:00	3,9102454	95,96225	5,495161	148,3925
2022-01-02T00:00:00	3,5702453	98,1573	6,193674	144,46231
2022-01-02T01:00:00	3,4202452	98,012985	6,6087217	150,64224
2022-01-02T02:00:00	2,8802452	97,087776	6,28713	156,37062

Figure 3. 8. Structure de la base de données

⁸ <https://www.meteoblue.com/fr/meteo/semaine>

Chapitre 3 : L'Approche proposée « Conception »

4. L'approche proposée

Usuellement, les données captées dans le domaine d'agriculture sont hétérogènes, peuvent contenir des informations redondantes ou non pertinentes. Dans ce contexte, nous allons proposer une architecture qui vise à extraire **un groupe d'intérêt qui contient les informations (échantillons) les plus pertinentes**, en se basant sur un nouvel algorithme de clustering qui utilise Markov Blanket Approximation et s'appuie sur le coefficient de corrélation d'information.

Les principales étapes dans le workflow de l'approche proposée sont (Figure 3.9) :

1. Préparation de la base de données ;
2. Définir le vecteur d'intérêt qui représente les valeurs idéales des caractéristiques pour avoir la bonne irrigation ;
3. Appliquer l'algorithme de clustering proposé ;
4. Générer les k groupes avec les k centres ;
5. En utilisant une mesure de similarité « Jaccard », extraire le groupe d'intérêt.

Chapitre 3 : L'Approche proposée « Conception »

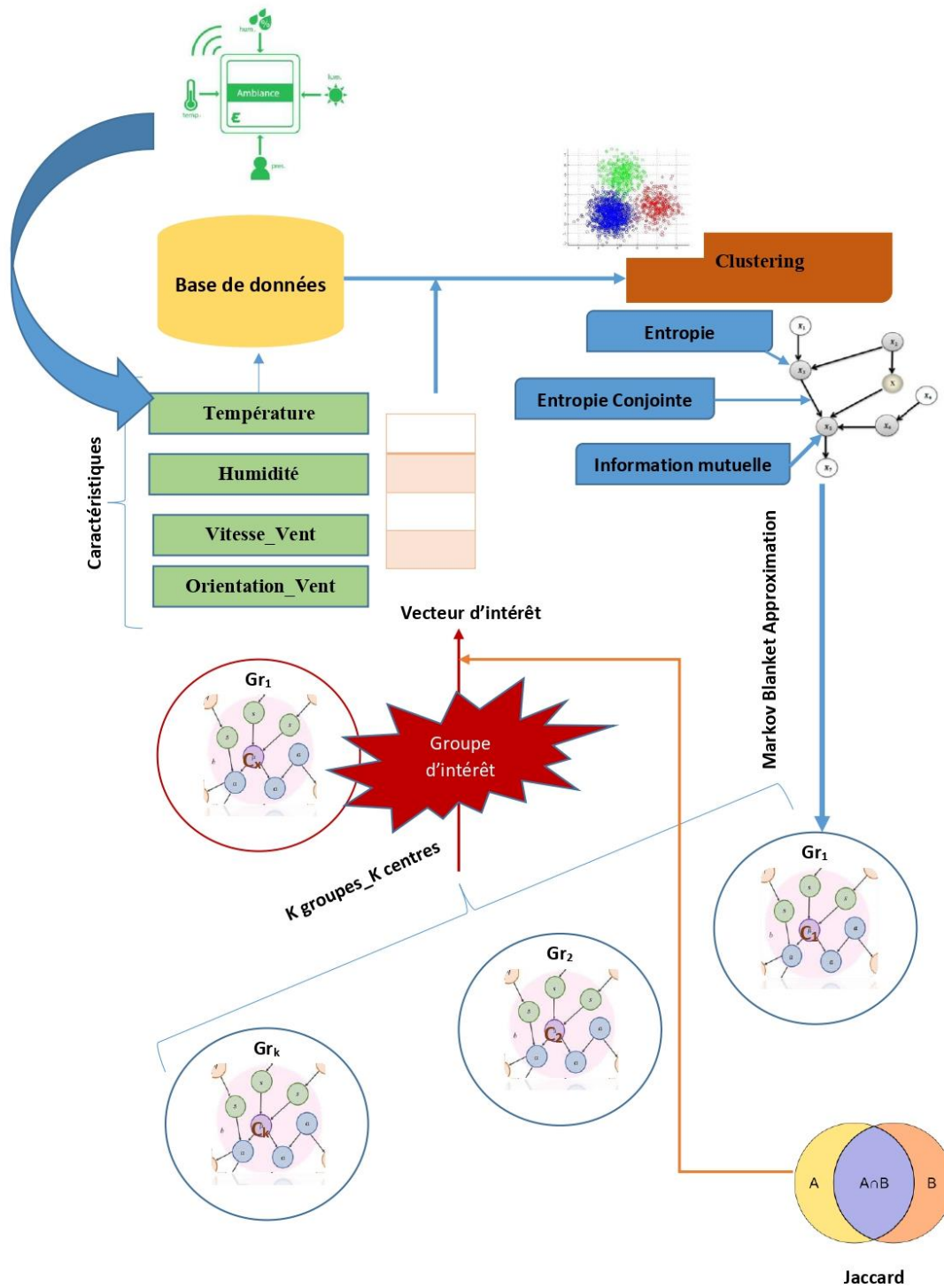


Figure 3. 9. La structure générale de l'approche proposée

Chapitre 3 : L'Approche proposée « Conception »

5. Clustering basé sur l'approximation d'une couverture de Markov

Comme nous avons dit, l'algorithme de clustering vise à grouper les données avec une similarité maximale au sein du même groupe et dissimilarité maximale entre les groupes, deux échantillons sont similaires s'ils ont la même pertinence au problème à résoudre. Dans ce contexte, nous avons proposé un nouvel algorithme pour grouper les échantillons, l'algorithme est **non supervisé** c'est-à-dire ne demande pas le nombre de cluster comme paramètre initial pour distribuer les échantillons sur ces clusters selon certaine métrique. L'algorithme commence par un seul groupe contient un seul échantillon (centre), ensuite un processus itératif est lancé pour évaluer la pertinence d'échantillons avec le centre de ce groupe, par conséquent, une insertion à ce groupe ou une création d'un nouveau groupe est réalisée.

Ce clustering est basé sur le concept de l'approximation d'une couverture de Markov en utilisant les deux métriques de la théorie de l'information qui sont *l'information mutuelle* et *l'entropie conjointe*.

→ *L'entropie* $H(x)$ permet de mesurer l'incertitude dans une variable aléatoire X avec une probabilité de distribution marginale $p(x)$ comme suit (Rendas, 2006):

$$H(X) = - \sum_x p(x) \log p(x)$$

→ *l'entropie conjointe* $H(x,y)$ permet de mesurer l'information contenue dans un système de deux variables aléatoires par exemple X et Y , elle est calculée en utilisant la probabilité jointe $p(x,y)$ comme suit (Rendas, 2006):

$$H(X, Y) = - \sum_{x,y} p(x,y) \log p(x,y)$$

Dans la théorie des probabilités et la théorie de l'information, l'information mutuelle de deux variables aléatoires est une quantité mesurant la dépendance statistique de ces variables (Figure 3.10).

Chapitre 3 : L'Approche proposée « Conception »

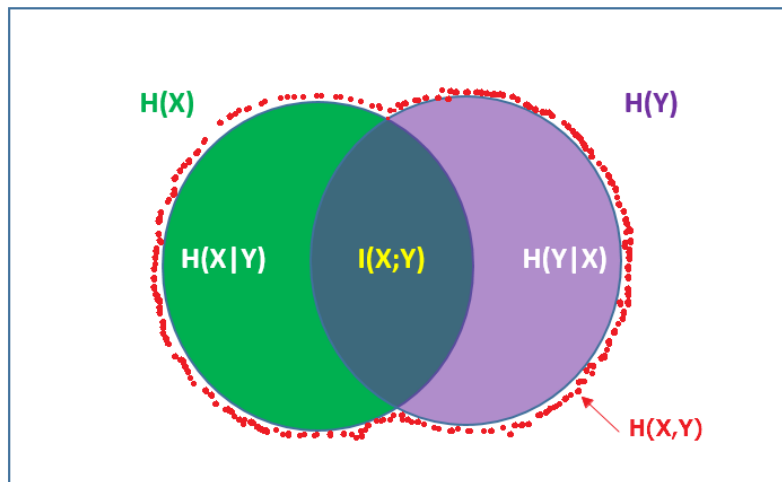


Figure 3. 10. Entropies individuelles ($H(X)$, $H(Y)$), jointes ($H(X, Y)$), l'information mutuelle $I(X; Y)$ ⁹.

5.1.L'approximation d'une couverture de Markov

En apprentissage automatique, la couverture de Markov pour un nœud X d'un réseau bayésien est l'ensemble des nœuds ∂X composés des parents de X , de ses enfants et des parents de ses enfants. Dans un réseau de Markov, la couverture de Markov d'un nœud est l'ensemble de ses nœuds voisins¹⁰, voir la figure 3.11.

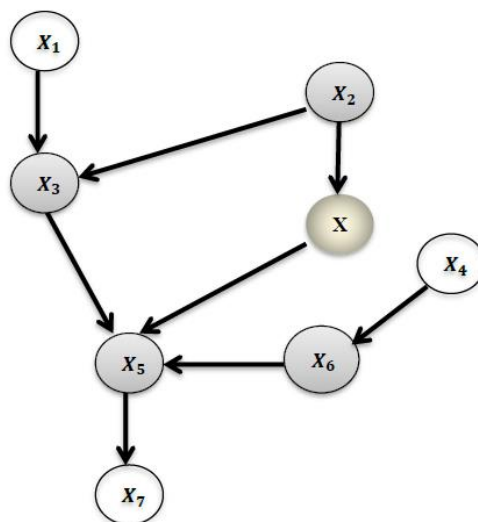


Figure 3. 11. Exemple d'une couverture de Markov dans un réseau Bayésien, le Markov Blanket de X , $M(X)=\{X_2, X_3, X_5, X_6\}$ (Sid, 2014)

⁹ <https://fr.linkedin.com/pulse/entropie-entropie-crois%C3%A9e-informations-n%C3%A9gatives-et-ahmad-abdallah>

¹⁰ https://fr.wikipedia.org/wiki/Couverture_de_Markov

Chapitre 3 : L'Approche proposée « Conception »

L'approximation d'une couverture de Markov que nous avons manipulé dans ce travail est basée sur un coefficient nommé le *coefficient de corrélation de l'information*, défini comme une normalisation de l'information mutuelle par l'entropie conjointe comme suit (Yuan, 2013):

$$ICC(X, Y) = \frac{MI(X, Y)}{H(X, Y)} \quad \text{Eq.1}$$

Sachant que, $MI(X, Y)$ est l'information mutuelle qui permet de mesurer la dépendance entre les deux variables (X et Y), estimée empiriquement par (Sid, 2014):

$$MI(X, Y) = H(X) + H(Y) - H(X, Y) \quad \text{Eq.2}$$

Soit :

- $icc(X, C)$ le coefficient de corrélation qui représente la pertinence entre l'échantillon X et le vecteur d'intérêt C,
- $icc(Y, C)$ le coefficient de corrélation qui représente la pertinence entre l'échantillon Y et le vecteur d'intérêt C,
- $icc(X, Y)$ le coefficient de corrélation qui représente la pertinence entre les deux échantillons X et Y.

→ X Est une approximation d'une couverture de Markov de Y si et seulement si :

$$\begin{cases} icc(X, C) \geq icc(Y, C) \\ icc(X, Y) \geq icc(Y, C) \end{cases} \quad \text{Eq.3}$$

→ Si X est une approximation d'une couverture de Markov de Y, alors ils ont la même pertinence au problème à résoudre donc nous pouvons les regrouper dans le même groupe.

5.2.L'algorithme proposé

Le pseudocode suivant illustre notre algorithme de clustering proposé pour grouper des données agricoles. Afin de générer le groupe d'intérêt, c'est-à-dire le groupe qui contient tous les échantillons climatiques représentant les conditions climatiques pour l'irrigation optimale, nous avons utilisé le coefficient de Jaccard comme métrique de similarité selon l'équation suivante (Sid, 2014) :

$$Sim(X, Y) = \frac{|X \cap Y|}{|X \cup Y|} \quad \text{Eq.4}$$

Chapitre 3 : L'Approche proposée « Conception »

Entrées : n : nombre des échantillons $X = (x_1; x_2; \dots; x_n)$;
C: Vecteur d'intérêt $C = (c_1; c_2; \dots; c_m)$;

Sorties : k : nombre de groupes $\{Gr_1, r_2, \dots, Gr_k\}$;
Cr: vecteur de centres ;

Début

- Attribuer le premier échantillon de X comme centre du premier groupe Gr_1 ,
 $Cr(1) = X(1)$;
- $k=1$;

// Calculer les clusters

Pour $i=2$ à $length(X)$ **faire**

- pour chaque échantillon $X(i)$;
Si \exists un groupe Gr_k , le centre de ce groupe est une approximation de couverture de Markov de cet échantillon, selon les équations Eq.1 et Eq.3, **alors**
 - Regrouper $X(i)$ à Gr_k ;

Sinon

- $k=k+1$;
- Affecter $X(i)$ à un nouveau groupe, $Cr(k) = X(i)$;

Finsi

Finpour

// Résultat de clustering

- Donner les k groupes $\{Gr_1, Gr_2, \dots, Gr_k\}$ et vecteur Cr de k centres ;

// Générer le groupe d'intérêt

Pour $i=1$ à k **faire**

- Calculer la similarité $S(Cr_i, C) = / i=1, \dots, k$ selon l'équation Eq.4 ;

Finpour

- Trier les valeurs de S obtenues de manière décroissante, on obtient la liste des groupes ordonnés selon S ;

-Sélectionner le premier groupe, groupe d'intérêt ;

Fin

Chapitre 3 : L'Approche proposée « Conception »

6. Conclusion

Nous avons présenté dans ce qui précède l'approche proposée pour le clustering de données agricoles captées dans un environnement IoT afin d'extraire les données les plus pertinentes, en utilisant Markov Blanket Approximation qui se base principalement sur le coefficient de corrélation d'information. Dans ce qui suit, nous allons implémenter l'approche, synthétiser les résultats et les discuter.

Chapitre 4 :

Implémentation & Résultat

Chapitre 4 : Implémentation & Résultats

1. Introduction

Le processus d'implémentation de l'algorithme de clustering proposé et expliqué dans le chapitre précédent sera présenté dans ce chapitre. Nous commencerons par la présentation des outils utilisés dans le développement, par la suite, nous présenterons quelques parties du programme, pour terminer par une démonstration et discussion des résultats obtenus.

2. Les outils et les bibliothèques de développement

Dans cette section, nous allons exposer les différents outils de programmation utilisés pour implémenter l'approche proposée.

2.1. Le langage Python

Python est un langage de programmation open-source qui possède plusieurs caractéristiques : simple, lisible, doté d'une vaste bibliothèque de base et offrant une grande variété de bibliothèques supplémentaires. Il est portable et prend en charge la programmation orientée objet, ce qui en fait un choix populaire pour de nombreux développeurs (Paris-Sud, 2016).



2.2. Anaconda Navigator

Anaconda Navigator est une interface utilisateur graphique (GUI) de bureau incluse dans *Anaconda® Distribution*, permet de lancer des applications et de gérer des packages, des environnements et des canaux conda sans utiliser les lignes de commandes. Navigator peut rechercher des packages sur Anaconda.org ou dans un référentiel Anaconda local. Il est disponible pour Windows, macOS et Linux ¹¹.

¹¹ <https://docs.anaconda.com/free/navigator/index.html>

Chapitre 4 : Implémentation & Résultats

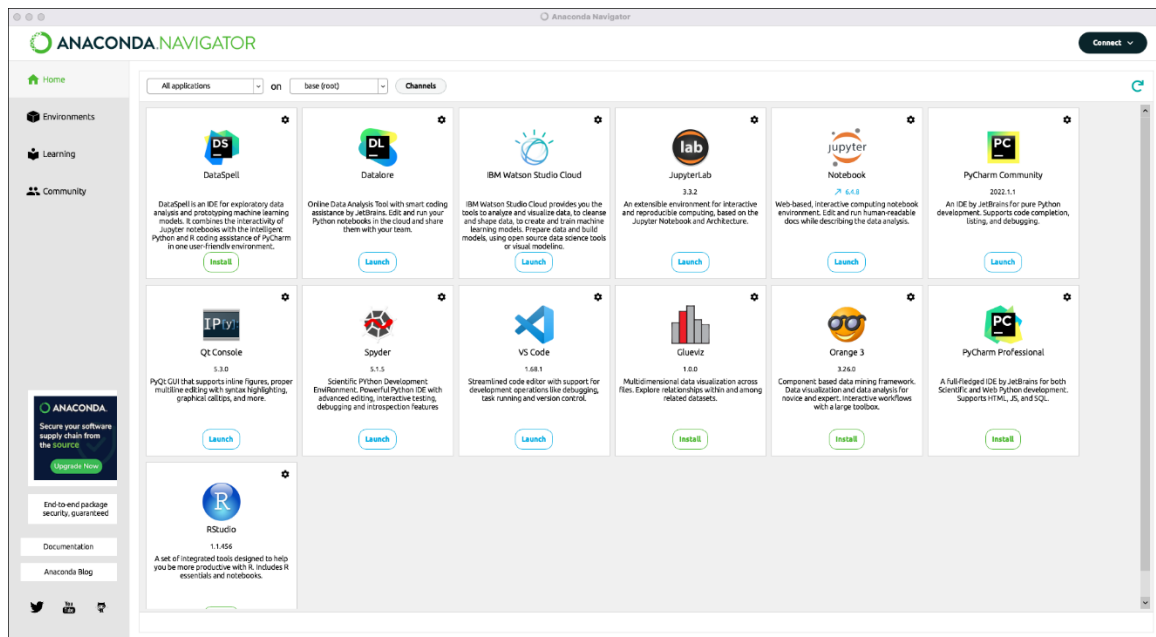


Figure 4. 1. L'interface de l'Anaconda Navigator ¹¹

2.3.L'environnement de programmation Spyder (sous Anaconda)

Spyder est un environnement de développement intégré (IDE) pour Python qui est disponible sur plusieurs plateformes comme Windows, MacOS, GNU et Linux, Il intègre de nombreuses bibliothèques scientifiques couramment utilisées, telles que Matplotlib, NumPy et Panda, ce qui facilite l'exploration, la visualisation et l'analyse statistique de données ¹².

¹² <https://fr.wikipedia.org/wiki/Spyder>

Chapitre 4 : Implémentation & Résultats

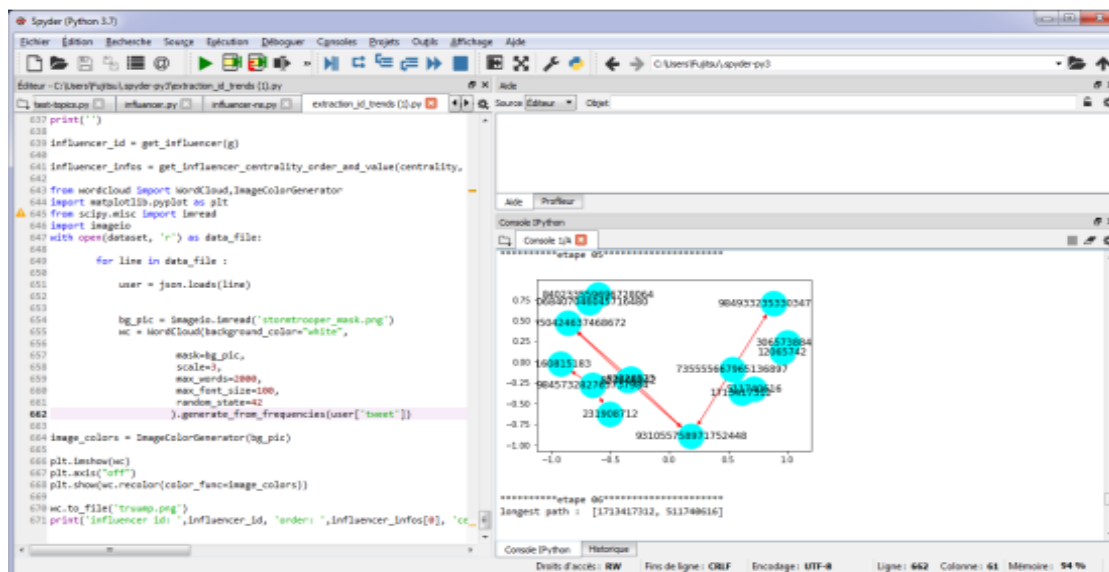


Figure 4. 2. L'interface de Spyder 12

2.4.NumPy

Numpy fournit à Python une bibliothèque mathématique riche pour effectuer des calculs numériques de manière efficace, permettant de travailler avec des tableaux multidimensionnels et des différents structures de données¹³.



2.5.Pandas

Pandas est une bibliothèque Python largement utilisée pour la manipulation et l'analyse de données. Elle facilite l'importation et l'exportation de données à partir de divers formats de fichiers, tels que CSV, Excel, SQL, JSON, etc. ce qui permet aux utilisateurs de charger et de manipuler facilement les données¹⁴.



2.6.Pygubu

Est un outil qui permet de développer rapidement et facilement des interfaces. Il génère automatiquement le code Python correspondant. Cela permet aux développeurs de gagner le temps et de se concentrer sur leurs applications¹⁵.



¹³ <https://courspython.com/apprendre-numpy.html>

¹⁴ <https://fr.wikipedia.org/wiki/Pandas>

¹⁵ <https://pypi.org/search/?q=Pygubu+%&o=>

Chapitre 4 : Implémentation & Résultats

3. Les implémentations

L'implémentation de notre approche proposée dans le chapitre précédant est divisée en plusieurs parties.

3.1. La préparation de données

La base de données que nous avons utilisée pour élaborer notre algorithme est téléchargée à partir de la plateforme Meteoblue contient 10860 échantillons selon quatre caractéristiques sous le format CSV (comma-separated values). La figure suivante illustre le code python généré pour télécharger les données d'évaluation.

```
14     def loadData(path = 'data.csv'):  
15         Gr = pd.read_csv(path).values  
16         return Gr
```

Figure 4. 3. Code Python pour télécharger Dataset.

3.2. Le calcul de coefficient de corrélation

Afin de calculer le coefficient de corrélation « ICC », premièrement, nous avons calculé l'entropie via la fonction **entropie (x)**, l'entropie conjointe via la fonction **joint_entropy (x, y)** et l'information mutuelle via la fonction **Mutual_info (x, y)**.

Par la suite, nous avons calculé le coefficient de corrélation d'informations (ICC) en fonction de l'information mutuelle et l'entropie conjointe via la fonction **ICC (x, y)**. La figure suivante (Figure 4.4) montre le code python développé pour calculer l'ICC.

Chapitre 4 : Implémentation & Résultats

```
18 def entropy(X):
19     _, counts = np.unique(X, return_counts=True)
20     p = counts / len(X)
21     return -np.sum(p * np.log2(p))
22 def joint_entropy(X, Y):
23     # combine X and Y into a single 2D array
24     XY = np.column_stack((X, Y))
25
26     # calculate the joint probability distribution
27     p_XY = np.histogram2d(XY[:,0], XY[:,1], bins=(np.unique(X), np.unique(Y)))
28
29     # calculate the joint entropy
30     H_XY = -np.sum(p_XY * np.log2(p_XY + (p_XY == 0)))
31
32     return H_XY
33
34 def Mutual_info(X,Y):
35     return entropy(X)+entropy(Y)-joint_entropy(X, Y)
36     :
37
38 def ICC(X,Y):
39     return (Mutual info(X,Y)/joint_entropy(X,Y))
40
```

Figure 4. 4. Code Python pour calculer l'ICC

3.3.Le calcul de l'approximation d'une couverture de Markov & Clustering

Pour calculer l'approximation d'une couverture de Markov nous avons utilisé le code montré dans la figure ci-dessous (Figure 4.5). Où *vecteur_Interet* représente un vecteur, nommé le vecteur d'intérêt, avec une taille de 4 correspond aux caractéristiques climatiques de notre base de données d'évaluation. Le choix des valeurs de ce vecteur se fait selon une étude bibliographique systématique concernant la bonne condition pour l'irrigation optimale des tomates dans une serre intelligente.

La tomate se développe parfaitement à une température de 30-33 °C et une bonne humidité. Cependant, si l'humidité relative tombe sous 50 %, la culture devient stressée et n'est plus capable de convertir toute la lumière qu'elle reçoit en production supplémentaire¹⁶. En outre, une température en dessous à 15°C, la plante peut présenter des symptômes de carence en eau et en phosphore¹⁷. A partir de cela, nous pouvons déduire que les bonnes conditions pour une irrigation optimale sont : température < 15 °C et humidité <50%. Tandis que, les valeurs de deux autres caractéristiques la vitesse du vent (km/h) et la direction du vent (°) sont

¹⁶ <https://hortimedia.ma/tomate-dete-sous-serre-les-cles-de-la-reussite/>

¹⁷ <https://caldor.fr/cultures/tomates/>

Chapitre 4 : Implémentation & Résultats

sélectionnées d'une manière aléatoire en se basant sur la logique d'irrigation, en raison du manque d'informations sur ces deux caractéristiques. Ex. *vecteur_Interet* = [13,20,13,2.5].

```
42 def Markov_blanket_approximation(x,y,vecteur_Interet):
43     # x is the Markov_blanket_approximation of y
44     if( ICC(x,vecteur_Interet) >= ICC(y,vecteur_Interet) and ICC(x,y) >= ICC(y,vecteur_Interet) ):
45         return True
46     else:
47         return False
```

Figure 4. 5. Code Python pour calculer Markov Blanket Approximation

En utilisant le Markov Blanket Approximation nous avons groupé les échantillons via le code montré dans la figure suivante (Figure 4.6).

```
48 def Grouping(G,Y):
49
50     MI=[]
51     for x in G:
52         MI=np.append(MI,Mutual_info(x,Y))
53
54     # mis en order des
55     indices = np.argsort(MI)[::-1]
56     G_ = G[indices]
57
58
59     # Assign the first ech in G' as a center of the first group Gr1,
60     first_value = G_[0]
61
62     Centers=np.empty((0,4))
63     Centers = np.append(Centers, np.array([first_value]), axis=0)
64
65     #Centers=np.r_[Centers,[[first_value]]]
66
67     k=0
68
69     Gr=[]
70     Gr.append([])
71     Gr[k].append(first_value.tolist())
72
73     for x in G_[1:]:
74
75         if(Markov_blanket_approximation(x,Centers[k-1],Y)):
76             Gr[k].append(x.tolist())
77         else:
78             k=k+1
79             Centers = np.append(Centers, np.array([x]), axis=0)
80             Gr.append([])
81             Gr[k].append(x.tolist())
```

Figure 4. 6. Code Python pour le clustering

Chapitre 4 : Implémentation & Résultats

3.4. Générer le groupe d'intérêt « information utile »

Après le regroupement des échantillons, les résultats sont les groupes avec ses centres. En utilisant le vecteur d'intérêt décrit précédemment, nous avons généré un groupe nommé le groupe d'intérêt contient tous les échantillons qui représentent les bonnes conditions pour l'irrigation optimale « information utile », cette génération se fait via le coefficient Jaccard pour calculer la similarité entre les centres des groupes et le vecteur d'intérêt *vecteur_Interet*. Le code Python suivant (Figure 4.7) illustre ce processus.

```
90     def jaccard_similarity(list1, list2):
91         set1 = set(list1)
92         set2 = set(list2)
93         intersection = len(set1.intersection(set2))
94         union = len(set1.union(set2))
95         similarity = intersection / union
96         return similarity
97
98     def g_similarity(centers, vecteur_Interet):
99         similarity=[]
100        for row in centers:
101            similarity.append(jaccard_similarity(row,vecteur_Interet))
102
103        return similarity
104
105    def find_max_index(lst):
106        max_index = lst.index(max(lst))
107        return max_index
108
109    def Group_Interet(Centres,Gr,vecteur_Interet):
110        return Gr[find_max_index(g_similarity(Centres,vecteur_Interet))]
```

Figure 4. 7. Code Python pour générer le groupe d'intérêt

4. Résultats & Discussion

Afin de manipuler notre implémentation, nous avons créé une interface simple et facile à utiliser, illustré dans la figure 4.8. Elle est constituée d'un :

- Bouton pour importer les données afin de les visualiser ;
- Les champs pour introduire le vecteur d'intérêt ;
- Bouton pour lancer le clustering ;

Chapitre 4 : Implémentation & Résultats

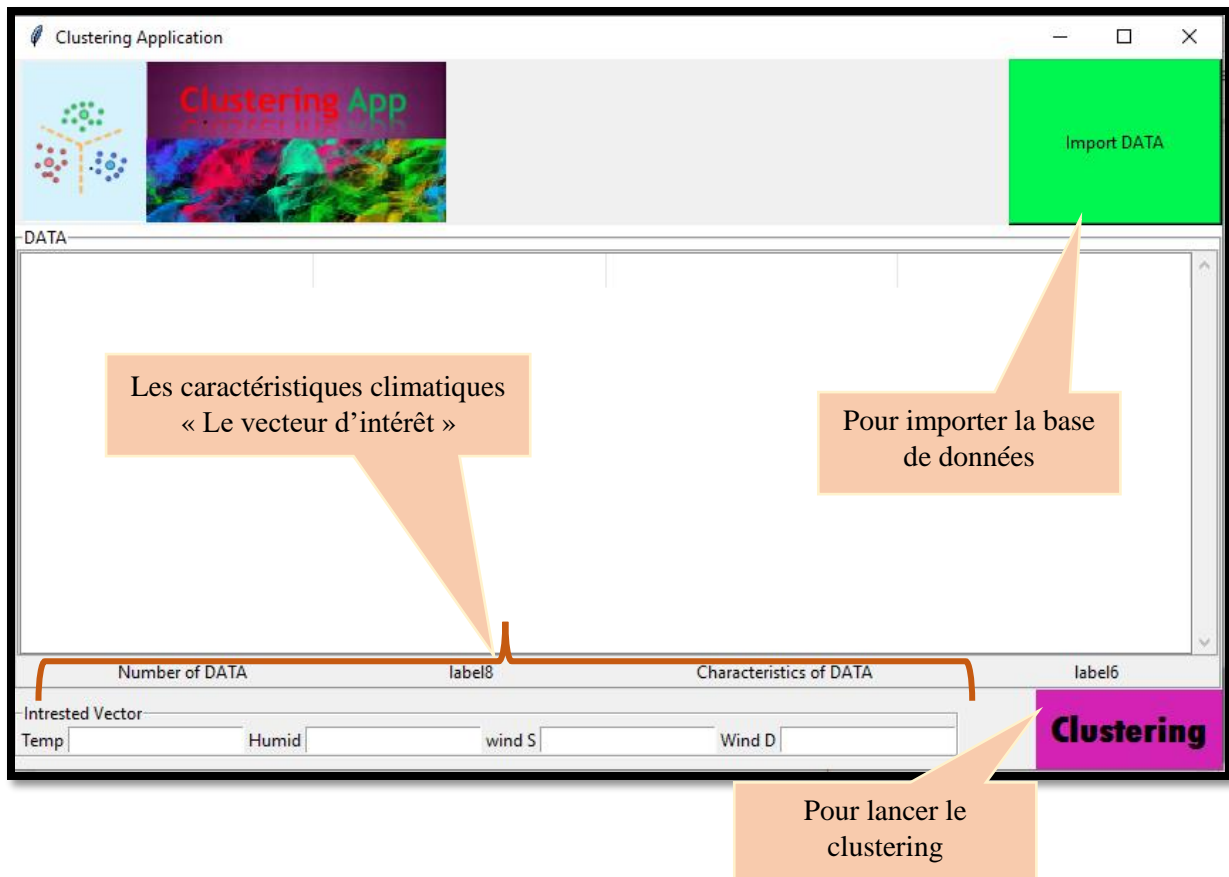


Figure 4. 8. L'interface principale

4.1.Scénario d'évaluation 1 : Importer les données

La figure ci-dessous (Figure 4.9) présente le résultat de l'importation de notre base de données en montrant la taille de cette dernière (10825 * 4).

Chapitre 4 : Implémentation & Résultats

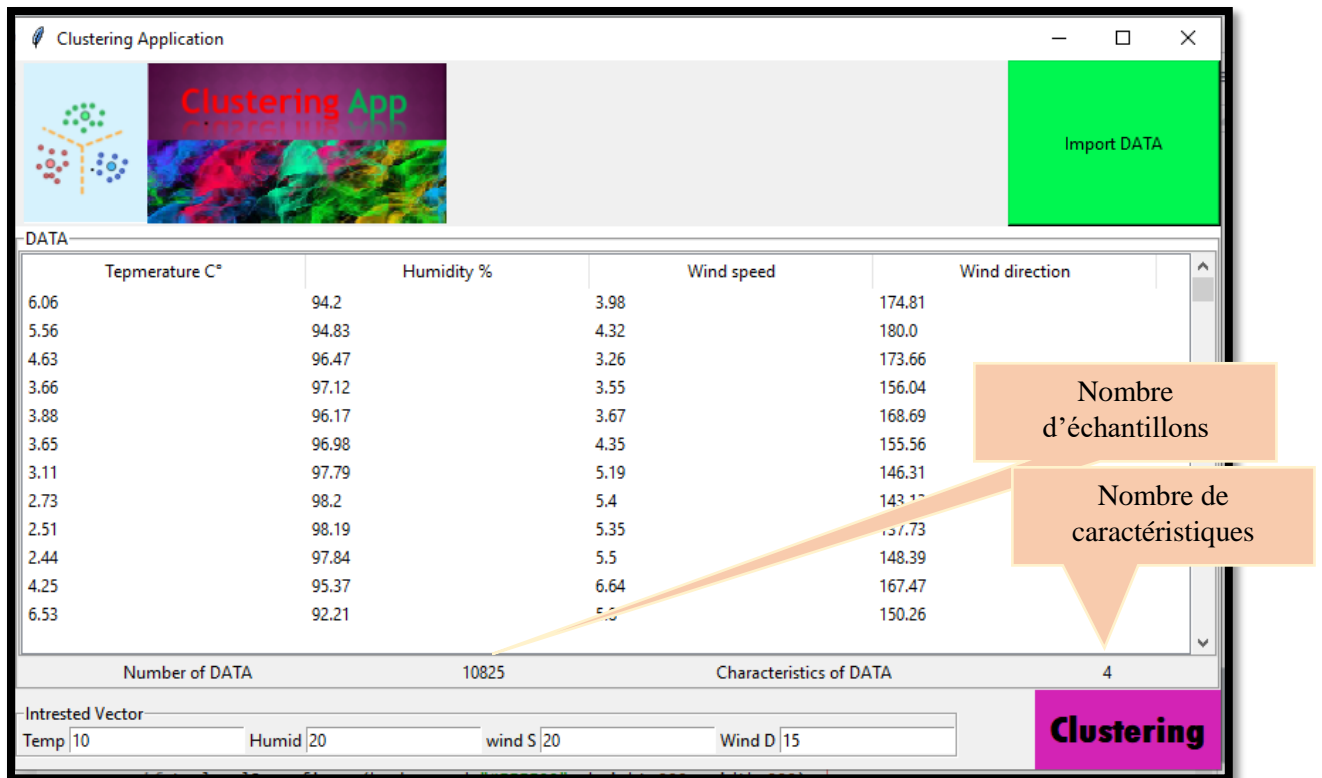


Figure 4. 9. Importer les données

4.2.Scénario d'évaluation 2 : Le clustering

En utilisant trois différents vecteurs d'intérêt, les résultats sont récapitulés dans le tableau 4.1, pour chaque vecteur, nous avons montré le nombre de groupes et la taille du groupe d'intérêt.

Vecteur d'intérêt	Nombre de Groupe	Taille groupe_intérêt
[10, 20, 20, 15]	116	173
[15, 50, 6, 50]	555	58
[13.5, 50, 4, 3]	4	388

Tableau 4. 1.Les résultats de clustering selon des différents vecteurs d'intérêt

D'après les résultats montrés dans le tableau au-dessus, nous pouvons remarquer la **sensibilité** de notre algorithme, c'est-à-dire une petite modification dans les valeurs des caractéristiques influe immédiatement sur les résultats d'une manière remarquable.

Chapitre 4 : Implémentation & Résultats

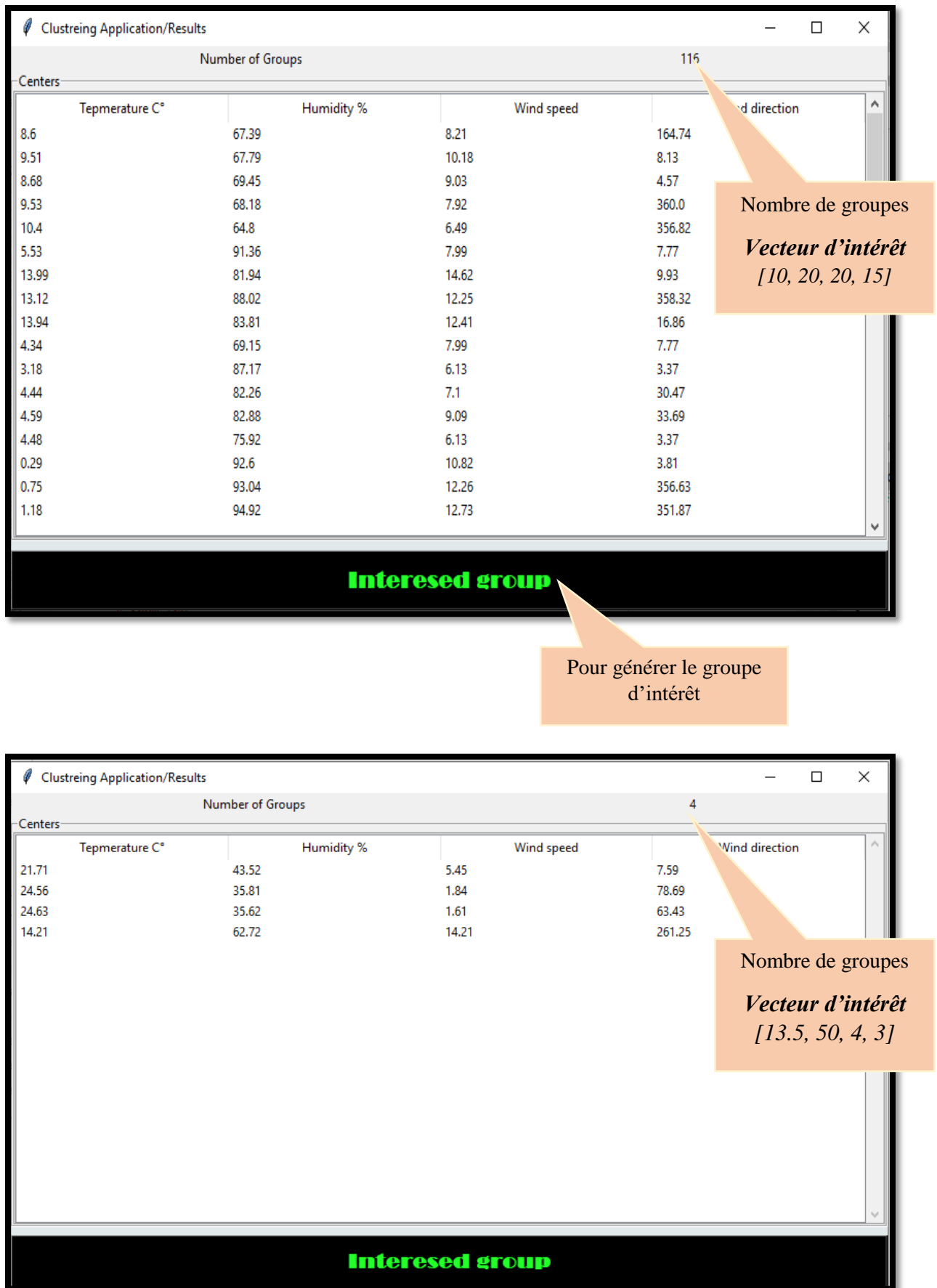


Figure 4. 10. Le résultat de clustering

Chapitre 4 : Implémentation & Résultats

4.3.Scénario d'évaluation 3 : Générer le groupe d'intérêt

La dernière étape dans notre processus d'évaluation est la génération du groupe d'intérêt qui représente l'information utile, le résultat est illustré dans la figure 4.11.

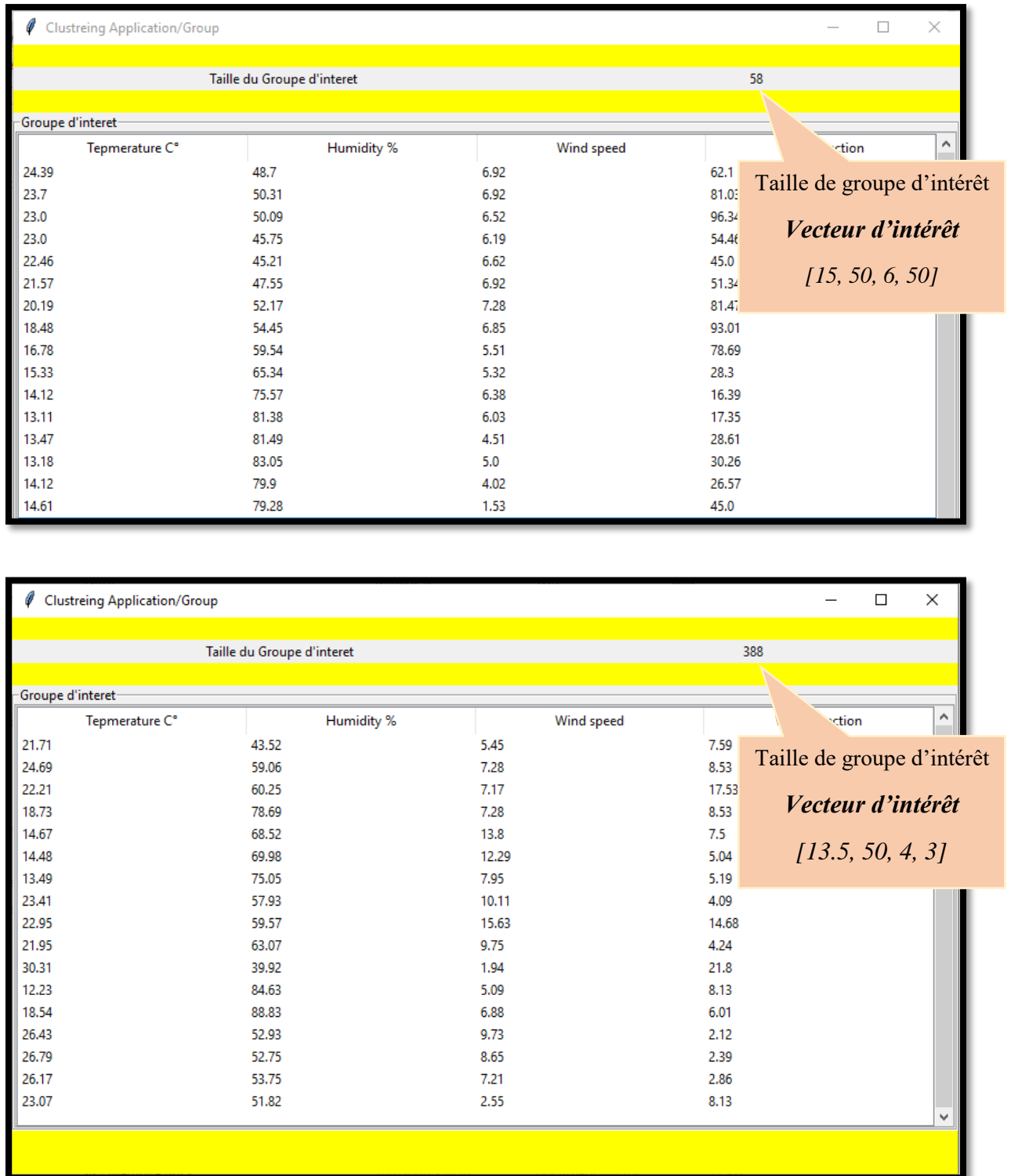


Figure 4. 11. Extraction de l'information utile

Chapitre 4 : Implémentation & Résultats

5. Conclusion

Dans ce chapitre, nous avons présenté le processus suivi afin d'implémenter et évaluer notre algorithme proposé. D'abord, nous avons montré les différents outils et bibliothèques utilisés, par la suite, nous avons exposé des parties de codes développés, pour terminer par une présentation des résultats selon des différents scénarios d'évaluation.

Conclusion Générale

Conclusion Générale

Dans ce travail, nous avons proposé et implémenté un nouvel algorithme de clustering basé sur l'approximation d'une couverture de Markov. L'algorithme est structuré en deux étapes principales, la première consiste à grouper les données en utilisant le coefficient de corrélation de l'information, et la deuxième vise à extraire le groupe d'intérêt qui représente les informations utiles. L'algorithme est appliqué sur des données agricoles captées dans un environnement IoT. D'abord, nous avons utilisé un simulateur spécialisé TinkerCad pour générer notre base de données d'évaluation, mais nous sommes tombés dans le problème du temps, pour avoir une base bien structurée via TinkerCad nous avons besoin des mois. Afin de surmonter ce problème, nous avons utilisé une base de données alternative extraite à partir de la plateforme Meteoblue constituée de 10825 échantillons pris en se basant sur les 4 variables (caractéristiques) : la température (°C), l'humidité (%), la vitesse du vent (km/h) et la direction du vent (°).

Afin d'implémenter notre proposition, nous avons utilisé un ensemble des outils et bibliothèques, tels que Spyder NumPy Pandas Pygubu. Selon une étude systématique, nous avons choisi un vecteur d'intérêt pour l'irrigation optimale, ce vecteur est utilisé pour évaluer notre application. L'évaluation se fait selon des différents scénarios qui montrent le bon fonctionnement et la sensibilité de notre algorithme.

Comme perspectives de ce travail, nous espérons d'élaborer l'algorithme sur des données réelles (captées dans un environnement IoT réel) ; hybrider l'algorithme avec des algorithmes d'apprentissage supervisé et profond pour améliorer les résultats de la prédiction dans l'agriculture de précision.

Références Bibliographiques

Références Bibliographiques

Ah-Pine, J., 2019. Apprentissage automatique., s.l.: s.l. : Université Lyon 2 M2..

Aman, G., 2021. Fuzzy C-Means Clustering (FCM) Algorithm.

Asif, R. D. A., 2023. Heterogeneity in US Farms: A New Clustering by Production Potentials. Agriculture , 13(2).

Astalaseven, E., 2012. Arduino pour bien commencer en électronique et en programmation. s.l.:Licence Creative Commons BY-NC-SA 2.0.

AutoDeskTinkerCAD, 2023. AUTODESK TINKERCAD. [En ligne] Available at: <https://www.lesnumeriques.com/telecharger/autodesk-tinkercad-49632> [Accès le 6 6 2023].

Babita, B. e. a., 2023. Role of IoT in Agriculture in India in Water Irrigation. International Journal for Research in Applied Science and Engineering Technology, Volume 11, pp. 6520-6524.

Bairam, S. & C. S., 2022. IOT IN AGRICULTURE. preprint paper.

Boris, I. G.-E., 2021. A review on the methods for big data analysis in agriculture. IOP Conference Series Materials Science and Engineering .

Chen, H. e. a., 2018. The rise of deep learning in drug discovery.. Drug discovery today, Volume 23, pp. 1241-1250.

Deepak, K. V., 2022. Role of IOT in introducing Smart Agriculture. International Research Journal of Engineering and Technology, 9(6).

Dileep, K. Y., 2015. A COMPARATIVE ANALYSIS OF CLUSTERING ALGORITHMS FOR AGRICULTURAL DATA. International Journal of Current Research, 7(7), pp. 18361-18364.

DJEBRI, L. L. M., 2018. Prévion de stationnement dans un environnement IoT. s.l.:Mémoire Master 2-Informatique-.

Références Bibliographiques

- Dutech, A. & S. M., 2003. Un algorithme d'apprentissage par renforcement pour les processus décisionnels de markov partiellement observés : apprendre une extension sélective du passé. *Revue d'Intelligence Artificielle.*, 17(4), pp. 559-589.
- Fathy, C. & A. H., 2023. A Secure IoT-Based Irrigation System for Precision Agriculture Using the Expedient Cipher. *Sensors.*, Volume 23.
- Fréou, C. & G., 2023. Découverte des cartes Arduino, s.l.: s.n.
- Gaganpreet, S. H. e. a., 2023. Exploring Barriers to the Adoption of Internet of Things-Based Precision Agriculture Practices. *Agriculture*, 13(163).
- Guy, S., 2023. Découverte de la carte Arduino, s.l.: s.n.
- Halkidi, M. & B. Y. V. M., 2001. On Clustering Validation Techniques. 17(2-3).
- Hausman, I., 2019. Le Clustering: Définition et Top 5 Algorithmes.
- Henry, L., 2022. Clustering 15-381 Artificial Intelligence, s.l.: s.n.
- Hizkia, 2019. Iot Components: Internet of Things. [En ligne] Available at: <https://fr.scribd.com/document/407409421/Internet-of-Things#> [Accès le 2023 6 07].
- International Journal of Engineering Applied Sciences and Technology, 2. V. 5. I. 7. I. N. 2.-2. P. 1.-1. P. O. N. 2. i. I. (. M. L. I. A. T. A. A. N., s.d.
- ISMAILI, Z., 2019. Apprentissage Supervisé Vs. Non Supervisé. [En ligne] Available at: <https://brightcape.co/apprentissage-supervise-vs-non-supervise/> [Accès le 30 5 2023].
- Jharna, M. S. N. & S. A., 2017. A review on the methods for big data analysis in agriculture. *IOP Conference Series Materials Science and Engineering.*
- Jiang, Q. F. & W., 2011. Study on delineation of irrigation management zones based on management zone analyst software. *Computer and Computing Technologies in Agriculture.*
- KAMBOUCHE, S. & A. I., 2018. Conception et réalisation d'un système d'agriculture. s.l.:Mémoire de Master, Centre Universitaire Belhadj Bouchaib d'Ain-Temouchen.
- Manjula, E. & D. S., 2017. A Model for Prediction of Crop Yield. *International Journal of Computational Intelligence and Informatics*, 6(4).

Références Bibliographiques

- Marco, L., 2021. Internet of Things: A General Overview between Architectures, Protocols and Applications. *Information*, 12(87).
- Meftah, Z., 2020. Une approche cloud computing basée IoT pour le smart House.. s.l.:Université de Biskra, Thèse de Doctorat. .
- Merima, K., 2021. A survey on Machine Learning-based Performance Improvement of Wireless Networks: PHY, MAC and network layer. *Electronics*. .
- Negi, H., 2021. Iot-Based Smart Farming System Using Soil Properties. *Elementary Education Online*.
- Neha, Y. e. a., 2020. MACHINE LEARNING IN AGRICULTURE: TECHNIQUES AND APPLICATIONS. *International Journal of Engineering Applied Sciences and Technology*, 5(7), pp. 118-122.
- Nevena, G. C. K. & W., 2019. A scalable system for executing and scoring K-means clustering techniques and its impact on applications in agriculture. *Int. J. Big Data Intelligence*, 6(2/3).
- Paris-Sud, U., 2016. Introduction à la programmation en langage Python. s.l.:Université Paris-Sud.
- Rendas, M., 2006. Théorie de l'Information. s.l.:SIC-SICOM.
- Rwan, M. e. a., 2015. Internet of Things (IoT) Security: Current Status, Challenges and Prospective Measures. *10th International Conference for Internet Technology and Secured Transactions (ICITST)*.
- Sabir, A. e. a., 2020. BlockChain with IoT, an Emergent Routing Scheme for Smart Agriculture. *International Journal of Advanced Computer Science and Applications*, 11(4).
- Safa, A. e. a., 2018. Overview for Internet of Things: Basics, Components and Applications. *Journal of University of Anbar for Pure Science*. , 12(3).
- Shadi, A. e. a., 2023. IoT-Enabled Precision Agriculture: Developing an Ecosystem for Optimized Crop Management. *Information*, 14(205).
- Sid, K. & B. M., 2014. Classification et prédiction en bioinformatique, s.l.: Thèse de magister, Université d'Oum El Bouaghi. .

Références Bibliographiques

- Sid, K. & B. M., 2018. Ensemble learning for large scale virtual screening on apache spark. Computational Intelligence and Its Applications: 6th IFIP TC 5 International Conference, CIIA 2018, Oran, Algeria, May 8-10, Proceedings 6. Springer International Publishing.
- Sid, K. & B. M., 2021. Distributed heterogeneous ensemble learning on Apache Spark for ligand-based virtual screening. International Journal of Data Mining, Modelling and Management, 13(1-2), pp. 160-191.
- Sid, K. B. M. & M. C., 2020. Analyse des Big Data médicales pour la découverte de médicaments, Constantine 2 university: Thèse de doctorat..
- Sid, K. Z. S. & M. C., 2022. DeepD_DrugC: Deep and distributed workflow to predict drug-candidates. 4th International Conference on Pattern Analysis and Intelligent Systems (PAIS). IEEE.
- Talal, S., 2019. INTERNET OF THINGS-IOT: DEFINITION, ARCHITECTURE AND APPLICATIONS.. Egypt. J. of Appl. Sci., 34(1).
- Tarun, G., 2019. Measures of Proximity in Data Mining & Machine Learning.
- Vandana, B. K., 2019. Hybrid K Mean Clustering Algorithm for Crop Production Analysis in Agriculture. International Journal of Innovative Technology and Exploring Engineering, 9(2).
- Veerpal, K. R. K., 2019. Role of IoT in agriculture. Journal of Pharmacognosy and Phytochemistry, Volume 8, pp. 422-425.
- Yuan, T. e. Z. L., 2013. Feature selection and prediction with a Markov blanket structure learning algorithm.. Proceedings of the 12th Annual UT-ORNL-KBRIN Bioinformatics Summit 2013.