



MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR  
Et de la Recherche Scientifique  
Université Abbes Laghrou Khenchela  
Faculté des sciences et technologie  
Département De Mathématiques Et informatique

N° de série :.....

**Mémoire de fin d'études**  
*Pour l'obtention du diplôme de Master (L.M.D)*

Spécialité : Informatique  
Option : STW

**Thème :**

**Proposition d'une méthode pour classifier les services IOT**

*Réalisés par :*

*- Kacha Karam Allah  
- Aissaoui Sana*

*Encadré par :*

**Dr. Hemam Sofiane**

*Présentés le : /07 / 2021*

# *Dédicaces*

## *À nos très chers parents*

Nous vous devons ce que nous sommes aujourd'hui, grâce à votre amour, votre patience et vos innombrables sacrifices. Que ce modeste travail, soit pour vous une petite compensation et reconnaissance pour tout ce que vous avez fait. Que Dieu, vous préserve et vous procure santé et longue vie afin que nous puissions à notre tour vous combler.

## *À nos très chers frères et sœurs*

Aucune dédicace ne pourrait exprimer assez profondément ce que nous ressentons envers vous. Nous vous dirons tout simplement, un grand merci, nous vous aimons.

## *À nos très chers ami(e)s*

En témoignage de l'amitié sincère qui nous lie et les bons moments passés ensemble. Nous vous dédions ce travail en vous souhaitons un avenir radieux et plein de réussites

# *Remerciement*

**E**n premier lieu nous remercions DIEU tout puissant de nous avoir donné la patience, la santé et la volonté pour achever ce travail.

**E**t Nos remerciements vont tout particulièrement à nos parents, pour leur soutien et leur patience

**N**ous aimerons adresser plus qu'un merci pour notre encadreur monsieur Hemam Sofiane. Qui a su partager son savoir faire, ses connaissances et son temps pour nous porter aide pendant et hors de ses heures de travail.

**E**nfin, nous adressons nos plus sincères remerciements à tous nos proches et amis, ils nous ont toujours soutenus et encouragés pendant la réalisation de ce mémoire.

**M**erci à vous tous.

## Résumé

L'Internet des objets (IoT) est aujourd'hui l'une des technologies à croissance rapide. Il s'agit d'une technologie grâce à laquelle des milliards d'objets ou d'appareils intelligents appelés « objets » peuvent utiliser plusieurs types de capteurs pour collecter divers types de données sur eux-mêmes et/ou sur l'environnement qui les entoure. Ils peuvent ensuite partager cela avec des parties autorisées pour servir à plusieurs fins telles que le contrôle et la surveillance des installations industrielles ou l'amélioration du service ou des fonctions de l'entreprise.

Il y a actuellement 20 milliards d'appareils connectés à Internet. Le nombre passera à 70 milliards d'ici 2025. Alors que ces appareils nous facilitent la vie.

Cette recherche propose une nouvelle approche pour la recherche services dans le grand nombre de services disponibles sur le Web en fonction des besoins de l'utilisateur, pour atteindre cet objectif, nous combinons deux algorithmes robustes : l'algorithme k-means est l'un des algorithmes d'apprentissage automatique et l'algorithme génétique est l'un des algorithmes d'optimisation.

K-means est un algorithme qui peut être utilisé pour regrouper efficacement les données en groupes en fonction des similitudes entre ces données.

Un algorithme génétique est un algorithme qui peut être utilisé pour créer le service idéal à partir des données dont nous disposons.

**Mots-clés :** Internet des objets (IoT), algorithme k-means, algorithme génétique.

## **Abstract**

The Internet of Things (IoT) is one of the fastest growing technologies today. It is a technology whereby billions of smart objects or devices called "things" can use multiple types of sensors to collect various types of data about themselves and / or the environment that surrounds them. Surrounding they can then share this with authorized parties to serve multiple purposes such as controlling and monitoring industrial facilities or improving service or business functions.

There are currently 20 billion devices connected to the Internet .The number will increase to 70 billion by 2025. While these devices make our lives easier.

This search offers a new approach to finding services from the large number of services available on the web based on user needs. To achieve this goal, we combine two robust algorithms: k-means is one of the machine learning algorithms and the genetic algorithm is one of the optimization algorithms.

K-means is an algorithm that can be used to efficiently group data into groups based on the similarities between that data.

The genetic algorithm is an algorithm that can be used to create the ideal service from the data we have.

**Keywords:** Internet of things (IoT), k-means algorithm, genetic algorithm.

Introduction générale:.....	1
-----------------------------	---

## **Chapitre 1 : Internet des Objets**

1. Introduction:.....	4
2. Définition de l'internet des objets .....	4
3. Principes d'Internet des objets .....	5
4.Évolution d'Internet et son impact dans le monde .....	5
5. Architecture de l'internet des objets.....	6
5.1 Les objets passifs .....	6
5.2 Les objets actifs .....	6
6. Domaines d'application.....	8
6.1 Les villes intelligentes (Smart Cities).....	8
6.2 La santé (Smart Health).....	8
6.3 Le Transport (Smart Transport).....	8
6.4 L'énergie.....	8
6.5 L'industrie .....	8
7. Composants De L'Internet des objets .....	9
8. Défis de l'Internet des objets .....	11
8.1 La Découverte Automatique.....	11
8.2 Disponibilité et fiabilité .....	11
8.3 Interopérabilité.....	12
8.5 Evolutivité et passage à l'échelle (Scalabilité).....	12
8.6 Politique réglementaire.....	12
8.7 Propriété intellectuelle :.....	12
Conclusion.....	13

## **Chapitre 2: les algorithmes évolutionnaire et la classification**

1. Introduction .....	15
2. Algorithmes évolutionnaires .....	15
2.1 Principe de fonctionnement .....	15
2.2 Les Algorithmes Génétiques .....	16
2.2.1 Définition .....	16
2.2.2 Vocabulaire .....	16
2.2.3 Principe .....	18

2.2.4. Algorithme 2.1 : Algorithme Génétique Standard (SGA : Standard Genetic Algorithm).....	19
2.2.5 Mise en œuvre d'un algorithme génétique : .....	19
2.2.6 Domaines d'application des algorithmes génétiques.....	20
2.2.7 Avantages et inconvénients.....	20
2.2.7.1 Avantages .....	21
2.2.7.2 Inconvénients .....	21
3. classification.....	21
3.1 Définition.....	21
3.2 Méthodes de classification.....	22
3.2.1 Classification supervisée.....	22
3.2.1.1 Méthodes de classification supervisée .....	22
3.2.1.1.1 Algorithme K-plus proches voisins .....	22
3.2.1.1.2 Algorithme 2.2: Algorithme de K-NN .....	23
3.2.1.1.3 Avantages .....	23
3.2.1.1.4 Limites .....	23
3.2.2 Classification non supervisée.....	23
3.2.2.1 K-Means .....	25
3.2.2.1.1 Principe.....	25
3.2.2.1.2 Algorithme 2.3: L'algorithme de k-means.....	26
3.2.2.1.3 Exemple .....	26
3.2.2.1.4 Avantages et contraintes de l'algorithme K-means.....	27
Conclusion.....	28

### **Chapitre 3: Étude Conceptuelle**

1. Introduction .....	30
2. Objectif du projet et méthodologie.....	30
3. Méthode de conception adoptée .....	30
3.1UML (Unified Modeling Language) .....	31
4. Architecture générale du système: .....	31
4.1Phase 1 .....	31
4.2. Phase 2:.....	33
5. Architecture détaillé du système .....	34
5.1 K-Means: .....	37
5.1.1 Organigramme de l'algorithme de k-means .....	37

5.1.2 Principe :	37
5.1.3 Notion de similarité.....	38
5.2 Algorithme génétique: .....	38
5.2.1 Organigramme de l'algorithme génétique:.....	38
5.2.1 La normalisation: .....	39
5.2.2 La fonction dévaluation: .....	40
Conclusion.....	42

## **Chapitre 4: Implémentation**

1. Introduction .....	44
2. Environnement de Travail .....	44
2.1. Environnement matériel .....	44
2.2. Environnement logiciel.....	44
2.3 Framework de programmation .....	45
2.4. Bibliothèques utilisées .....	46
3. Définition d'ensemble de données utilisé et description des variables .....	47
3.1 Définition du jeu de données utilisé .....	47
3.2. Description des variables.....	47
3. Description de l'application .....	48
3.1 Code k-means : .....	48
3.2 Code d'algorithme génétique .....	52
3.3 Le code de l'interface :.....	57
3.4 L'interaction de l'utilisateur avec notre Système .....	58
Conclusion.....	62
Conclusion Générale: .....	63

---

## Liste des figures

Figure 1.1: L'IdO connecte des objets en utilisant des capteurs et Internet.....	5
Figure1.2: L'Evolution des objets connectés .....	6
Figure 1.3:Architecture d'IdO .....	7
Figure 1.4:Différentes Domaines d'application d'Ido.....	9
Figure2.1: Principe d'un algorithme évolutionnaire standard .....	16
Figure 2.2: Exemple de croisement : Croisement à un point .....	17
Figure 2.3: Exemple de mutation : Mutation à un point .....	18
Figure 2.4:Principe de la classification .....	22
Figure 2.6 :Les deux types de clustering hiérarchique/non hiérarchique.....	24
Figure 2.7: Partition hiérarchique.....	25
Figure 2.8 : Une illustration de l'algorithme K-means. ....	26
Figure 3.1 : les principales étapes pour réaliser la première phase .....	32
Figure 3.2:L'interaction de l'utilisateur avec notre système .....	33
Figure 3.3 : Diagramme de séquence de l'interaction de l'utilisateur avec notre système .....	34
Figure 3.4 Organigramme de l'algorithme de k-means .....	37
Figure 3.4 Organigramme de l'algorithme génétique.....	38
Figure 4.1: Logo de Python.....	45
Figure 4.2: Logo d'anaconda.....	45
Figure 4.3: Logo de l'environnement de développement Spyder .....	46
Figure 4.4 : La représentation de nos données .....	49
Figure4.5: Résulta de k-means .....	50
Figure 4.6: interface qui permet a l'utilisateur de choisir le nombre de capteurs.....	59
Figure 4.7: interface qui permet à l'utilisateur de choisir les capteurs .....	59
Figure 4.8: l'interface lorsque l'utilisateur met des caractéristiques de service.....	60
Figure 4.8: l'interface du résultat qui s'affiche à l'utilisateur.....	61

## Liste des tableaux

Tableau 1.1 :Les Composants de base d'un système IOT .....	9
Tableau 3.1: Notre dataset.....	34
Tableau 3.2:Normalisation des inputs .....	36
Tableau 3.3: Normalisation des Output.....	36
Tableau 4.1: Caractéristiques du pc utilisé pour le projet .....	44
Tableau 4.2 : Description des variables d'ensemble de données .....	47

### **Introduction Générale:**

#### **a. Contexte**

Depuis les années 1960, Internet, en tant que réseau massif de réseaux locaux (LAN), joue un rôle essentiel dans la connexion des personnes, des entreprises et des organisations. Internet a fait tomber les barrières géographiques entre les gens et leur a donné un moyen de communication robuste, efficace et rentable. Maintenant, il semble que les choses soient sur le point de changer dans le monde d'Internet en raison de l'apparition d'objets intelligents qui ont la capacité de générer et de communiquer des données sur Internet de la même manière que les humains. L'Internet des objets (IoT) est le système les plus récents qui ont le potentiel de changer notre mode de vie. On peut considérer l'IoT comme une technologie composée de deux composants ; « Objets » et « Internet ». Les « objets » font référence à tout objet ou appareil qui a la capacité de percevoir ou de collecter des données sur lui-même ou sur l'environnement qui l'entoure. Selon le type, l'intelligence et les capacités de cet objet, l'objet peut être capable d'analyser et d'agir intelligemment avec d'autres objets en utilisant « Internet » comme réseau de communication.

La communication dans l'IoT ne se limite pas à la communication entre les objets IoT uniquement. Il va au-delà de cela et aborde les humains d'une manière qui peut rendre la vie plus facile.

L'IoT est omniprésent et utilisé dans presque tous les aspects de notre vie. L'IoT est utilisé par les gouvernements du monde entier pour collecter des données de différents secteurs et fournir de meilleurs services

#### **b. Objectifs**

L'objectif de ce travail d'une part, est de créer une interface qui permet à l'utilisateur de rechercher facilement et en moins de temps les services dont il a besoin.

D'autre part, nous voulons réaliser un système qui puisse classer un grand nombre de services disponibles sur le web et renvoyer les meilleurs services à l'utilisateur. Ainsi, pour atteindre cette objectif nous avons fait appel aux algorithmes d'apprentissage automatique afin de classer les services IoT, et ceci sur la base de leur similarité fonctionnelle.

#### **c. Organisation du mémoire**

Après la présentation de notre objectif et l'introduction générale, ce mémoire est organisé en quatre chapitres comme suit :

**Chapitre 1 :** concerne l'Internet des Objets, Dans ce chapitre, nous définissons l'Internet des Objets, nous présentons son principe, son architecture et nous décrivons ses différents domaines d'application.

**Chapitre 2 :** Dans ce chapitre, nous aborderons tout d'abord quelques concepts de base, puis nous exposerons les algorithmes évolutionnaires en général et nous détaillerons l'algorithme génétique sa définition, son vocabulaire, et son principe. Dans la deuxième partie de ce chapitre, nous présentons la classification et nous prenons l'algorithme des k-moyennes comme cas particulier.

**Chapitre 3 :** Ce chapitre sera consacré à l'étude conceptuelle de notre système, une vue conceptuelle est basée sur les diagrammes UML de notre système dans ce chapitre, nous détaillerons les deux architectures (générale et détaillée) avec l'identification de la modélisation appropriée pour atteindre notre objectif.

**Chapitre 4:** Ce chapitre présent les l'environnement de développement et les outils utilisés pour développer notre système ainsi que les différents résultats atteints.

En dernier lieu, une conclusion générale clôtura ce mémoire et synthétise le travail réalisé et les perspectives envisagés.

# Chapitre 1

## Internet des Objets

## 1. Introduction:

L'internet a connu depuis son apparition, un grand essor et a touché pratiquement tous les domaines de notre vie de tous les jours : politique, économique, socioculturel,...etc. Elle est devenue la principale source d'information.

L'internet a également facilité la Communication entre les gens, en transformant le monde en une petite ville.

La technologie de l'Internet des Objets ou Internet of Things (IoT) est l'extension du réseau Internet à des choses/objets et à des lieux dans le monde physique qui repose sur l'idée que tous les objets peuvent être connectés un jour à Internet, ces objets sont adressables de manière unique. Les objets deviennent alors connectés et mis en réseau, tel que les montres connectées, bracelets connectés ou encore chaussures connectées...

L'internet des objets est en pleine croissance bénéficiant de la création du Cloud Computing et de son autonomie, de ce fait il peut être appliqué dans divers domaines.

## 2. Définition de l'internet des objets

Il n'existe pas de définition standard et unifiée de l'internet des objets, certaines définitions traitent les aspects techniques de l'IOT, alors que d'autres évoquent les usages et les fonctionnalités.

### Définition 1

L'internet des objets (IDO)<sup>1</sup> est une infrastructure dynamique d'un réseau global. Qui permet d'interconnecter des objets (physiques ou virtuels) grâce aux technologies de l'information et de la communication interopérables. [1]

### Définition 2

La technologie IOT est considérée comme l'émergence de l'Internet du futur, Certains la définissent comme des « objets ayant des identités et des personnalités virtuelles, opérant dans des espaces intelligents et utilisant des interfaces intelligentes pour se connecter et communiquer au sein de contextes d'usages variés » [2]

---

<sup>1</sup> En anglais Internet Of Things, IoT



**Figure 1.1:** L'IdO connecte des objets en utilisant des capteurs et Internet [1]

### 3. Principes d'Internet des objets

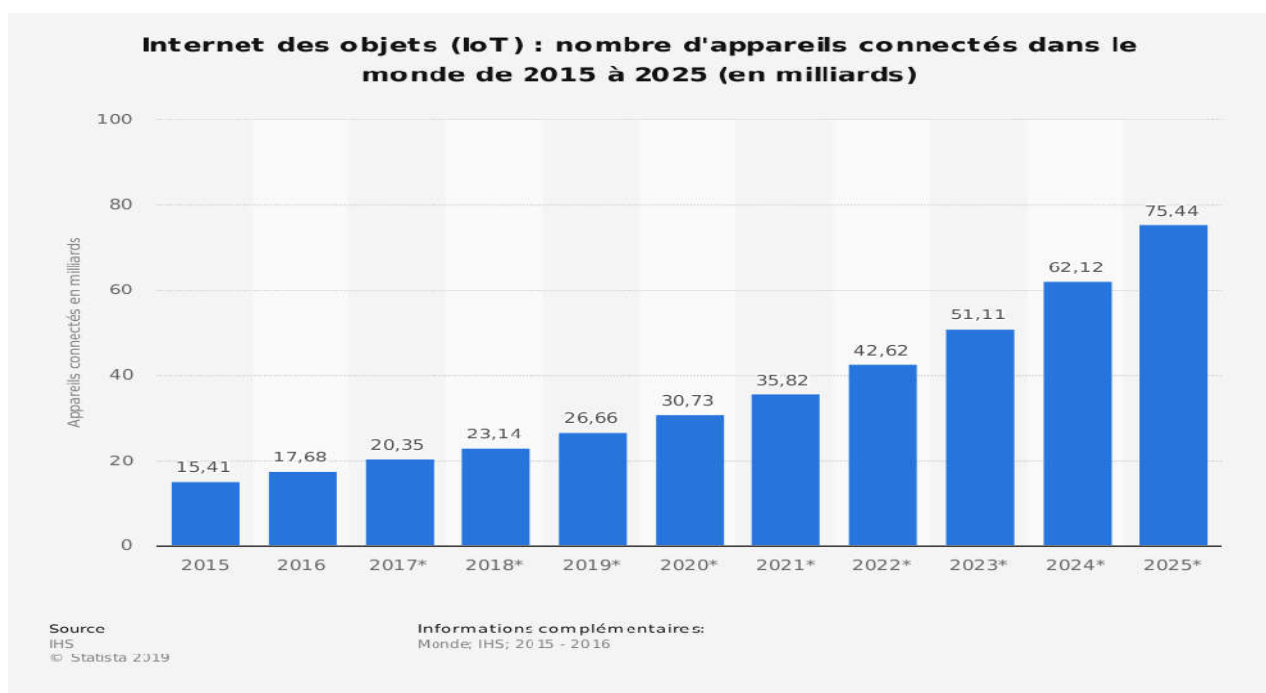
L'IOT est une combinaison d'innovations technologiques récentes et de solutions déjà existantes. Chaque objet est muni d'une identification électronique unique capable de lire et transmettre à travers un protocole dans le réseau internet. Il est nécessaire cependant de définir la nature de l'objet, ses fonctionnalités, sa position dans l'espace, l'historique de ses déplacements, etc. Pour effectuer ce lien entre physique et virtuel, le dispositif technique doit donc modéliser des contextes réels et les rendre virtuel. [3]

D'un point de vue technique, l'IdO consiste l'identification numérique directe et normalisée (adresse IP, protocole http...) d'un objet physique grâce à un système de communication sans fil (puce RFID, Bluetooth ou Wifi).[1]

### 4.Évolution d'Internet et son impact dans le monde

En 2015, la population mondiale a frôlé les 15 milliards d'individus et un demi-milliard d'appareils connectés à Internet.

L'idée de l'Internet des objets est apparue en 2009, boosté par l'apparition des Smartphones, le nombre d'appareils connectés à Internet a atteint 75,44 milliards en 2025.



**Figure1.2:** L'Evolution des objets connectés [10]

Le nombre de capteurs connectés à Internet pourrait augmenter de plusieurs millions, voire de plusieurs milliards du fait que tout ce qui existe se connecte (Animaux, lampes, maisons, personnes, chaussures, arbres,...).[4]

## 5. Architecture de l'internet des objets

Les objets de l'environnement de l'internet des objets permettent de collecter, stocker et transmettre des données issues du monde physique. Ce sont des sources de données

On distingue deux types d'objet :

### 5.1 Les objets passifs

Ils utilisent généralement un tag (puce RFID, code barre 2D). Ils ont une capacité de stockage faible (de l'ordre du kilooctet) et permettent de jouer le rôle de l'identification.

### 5.2 Les objets actifs

Ils peuvent être équipés de plusieurs capteurs, avec une grande capacité de stockage, capables d'accomplir des calculs et être en mesure de communiquer sur un réseau.

Précisons le rôle des différents processus présentés sur cette figure 1.3 :

**Capter:** permet de transformer une grandeur physique analogique en un signal numérique.

**Concentrer:** permet d'interfacier un réseau spécialisé d'objet à un réseau IP standard.

**Stocker:** permet de rassembler des données brutes, produites en temps réel, arrivant de façon non prévue.

**Présenter:** permet de collecter les informations de façon compréhensible par l'Homme, en lui offrant un moyen d'agir et/ou d'interagir.

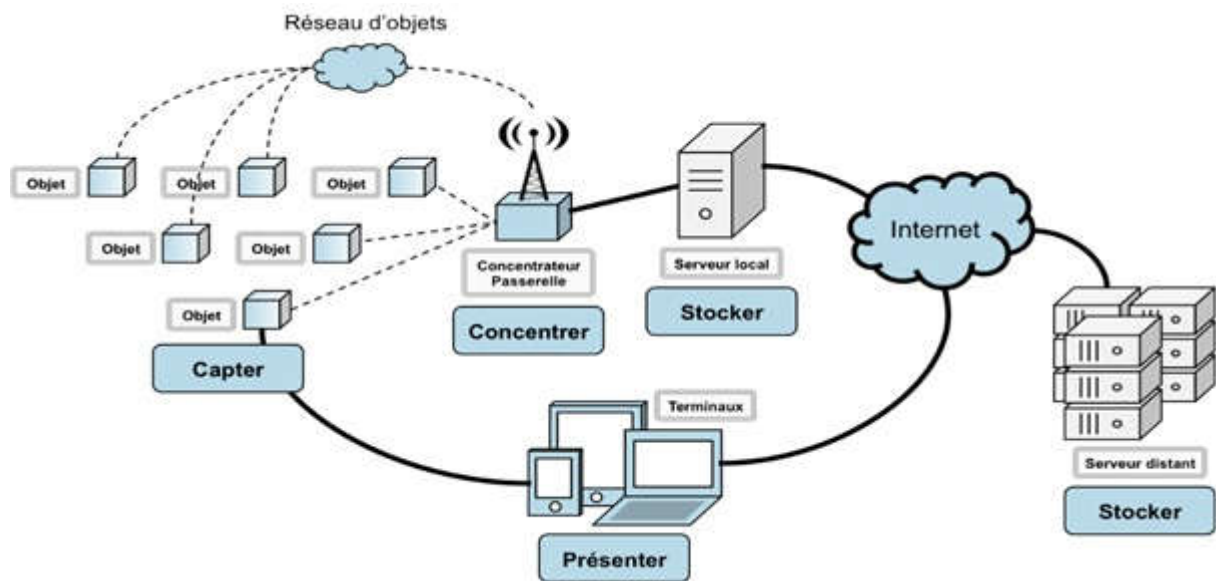


Figure 1.3: Architecture d'IdO [5]

Deux autres processus qui n'apparaissent pas sur le schéma :

**Le traitement des données :** est un processus qui peut intervenir à tous les niveaux de la chaîne depuis la capture de l'information jusqu'à sa restitution. Une stratégie pertinente, et commune quand on parle d'Internet des objets, consiste à stocker l'information. Cette stratégie est possible aujourd'hui grâce à des architectures distribuées, capables d'emmagasiner de grandes quantités d'information tout en offrant la possibilité de réaliser des traitements complexes en leur sein.

**La transmission des données :** est un processus qui intervient à tous les niveaux de la chaîne. Deux réseaux supportent des transmissions le réseau local de concentration (utilise ANT, ZigBee et Zwave..) et le réseau WAN (Wifi, réseaux cellulaires . . . .)[5]

## 6. Domaines d'application

L'internet des objets couvrira un large éventail d'applications et touchera quasiment à tous les domaines que nous affrontons au quotidien, ceci permettra l'émergence d'espaces intelligents. Parmi ces espaces intelligents, on peut citer :

### 6.1 Les villes intelligentes (Smart Cities)

L'internet des objets permettra une meilleure gestion des réseaux divers qui alimentent nos villes (eaux, électricité, gaz, etc.) en permettant un contrôle continu en temps réel et précis. Des capteurs peuvent être utilisés pour l'économie de l'eau et pour améliorer la gestion des parkings et du trafic urbain et diminuer les embouteillages et les émissions en CO<sub>2</sub>. [6]

Beaucoup de grandes villes ont été soutenues par des projets intelligents, comme Séoul, New York, Tokyo, Shanghai, Singapour, Amsterdam et Dubaï.

### 6.2 La santé (Smart Health)

Dans le domaine de la santé, L'ido permettra le déploiement de réseaux personnels pour le contrôle et le suivi des signes cliniques, notamment pour des personnes âgées, les objets connectés permettent de suivre la tension, le rythme cardiaque, la qualité de respiration ou encore la masse grasseuse. Ceci permettra ainsi de faciliter la télésurveillance des patients à domicile, et apporter des solutions pour l'autonomie des personnes à mobilité réduite. [6]

### 6.3 Le Transport (Smart Transport)

Des voitures connectées ou autonomes aux systèmes de transport/logistique intelligents, il peut sauver des vies, réduire le trafic et minimiser l'impact des véhicules sur l'environnement. [7]

### 6.4 L'énergie

L'Internet des objets permet aux innombrables appareils qui composent le réseau électrique de partager des informations en temps réel pour une distribution et une gestion plus efficaces de l'énergie [7].

### 6.5 L'industrie

La technologie Ido permettra un suivi total des produits, de la chaîne de production, jusqu'à la chaîne logistique et de distribution en supervisant les conditions d'approvisionnement. Cette traçabilité de bout en bout permet aux usines d'améliorer l'efficacité de ses opérations, d'optimiser la production et d'améliorer la sécurité des employés. [6]

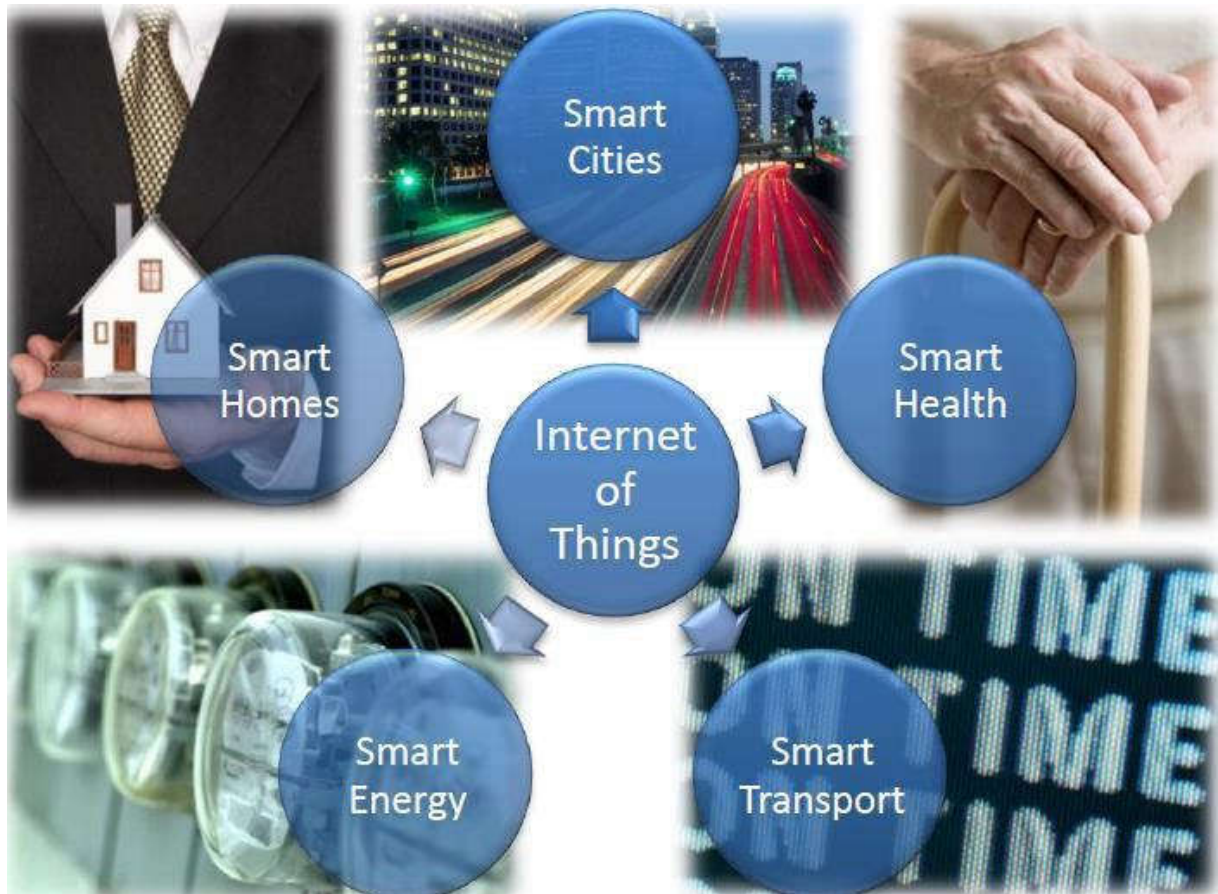


Figure 1.4: Différentes Domaines d'application d'IoT [6]

### 7. Composants De L'Internet des objets

Le tableau suivant présente les composants principaux de l'IOT:

جدول 1: **Tableau 1.1** : Les Composants de base d'un système IOT [8]

Composants internet des objets	La description
Objets physiques	Un objet connecté est un objet physique équipé de capteurs ou d'une puce qui lui permettent de transcender son usage initial pour proposer de nouveaux services. Il s'agit d'un matériel électronique capable de communiquer avec un ordinateur, un

	Smartphone ou une tablette via un réseau sans fil (Wifi, Bluetooth, réseaux de téléphonie mobile, réseau radio à longue portée de type Sigfox ou LoRa, etc.), qui le relie à Internet ou à un réseau local.
Capteurs	Ils sont installés sur les objets connectés, ils sont plus ou moins intelligents, selon qu'ils intègrent ou non eux-mêmes des algorithmes d'analyse de données, et qu'ils soient pour certains auto-adaptatifs. Les capteurs connus sont : Capteurs de température et thermostats, Capteurs de pression, Humidité / niveau d'humidité, Détecteurs d'intensité lumineuse, Capteurs d'humidité, Détection de proximité, Étiquettes RFID....
Gens	Exemple : Les humains peuvent contrôler l'environnement via des applications mobiles
Prestations de service	Exemple : Services Cloud - peuvent être utilisés pour: <ul style="list-style-type: none"> <li>• Traiter les Big Data et les transformer en informations précieuses</li> <li>• Construire et exécuter des applications innovantes</li> <li>• Optimiser les processus métier en intégrant les données de l'appareil.</li> </ul>

Plateformes	<p>Elle est considérée comme un type d'intergiciel utilisé pour connecter les composants Ido (objets, personnes, services, etc.) à l'environnement l'Ido. Elle fournit de nombreuses fonctions :</p> <ul style="list-style-type: none"> <li>• Accès aux appareils</li> <li>• Assurer une installation / un comportement correct de l'appareil</li> <li>• Analyse des données</li> <li>• Connexion interopérable avec le réseau local, le cloud ou d'autres périphériques.</li> </ul>
Réseaux	<p>Les composants Ido sont liés entre eux par des réseaux, utilisant diverses technologies, normes et protocoles sans fil et filaire .</p>

## 8. Défis de l'Internet des objets

Il y a des défis à l'application de l'Internet des objets de point de vu coût de la mise en œuvre du fait que l'attente de la technologie doit être disponible à faible coût avec un grand nombre d'objets. De plus, l'Ido est également face à de nombreux autres défis [8]:

### 8.1 La Découverte Automatique

Dans les environnements dynamiques les services appropriés pour les objets doivent être identifié automatiquement. Ceci nécessite des moyens sémantiques appropriés pour décrire leur fonctionnalité afin de les exploiter d'une manière efficace. [8]

### 8.2 Disponibilité et fiabilité

La méthode de collecte et de transmission des informations influence fortement la qualité des données fournies [9].

### **8.3 Interopérabilité**

L'hétérogénéité et la diversité des environnements logiciels et matériels des objets [9].

### **8.4 Sécurité et confidentialité**

Nécessité de sécuriser et cloisonner les données échangées [9].

### **8.5 Evolutivité et passage à l'échelle (Scalabilité)**

Trouver des solutions flexibles pour le passage à l'échelle dans un scénario d'objets dispersés et nombreux [9].

### **8.6 Politique réglementaire**

La réglementation n'est pas adaptée pour des applications Iod spécifiques. Par exemple, les entreprises investissent énormément dans ce domaine, mais l'autorisation de circulation des voitures autonomes n'est toujours pas claire du point de vue réglementaire [9].

### **8.7 Propriété intellectuelle :**

Une compréhension commune des droits de propriété entre les parties prenantes devrait être clairement définie pour libérer tout le potentiel de l'Ido. La question demeure ouverte, par exemple dans les dispositifs médicaux implantés dans le corps d'un patient, la question du droit sur les données générées, le patient ou le fabricant de l'appareil [9].

**Conclusion**

Dans ce chapitre on a présenté une étude détaillée sur l'internet des objets, sa définition, leurs principe et son architecture leurs domaine d'application puis les principaux composant de Ido et enfin ses défis, Parmi les défis majeurs de l'IOT la découverte automatique qui est l'une des techniques assurant l'identification automatique des objets, on va préciser ce défi dans les chapitres suivants.

## Chapitre 2

### Les algorithmes évolutionnaire et la classification

## 1. Introduction

Ce chapitre comporte deux parties, une partie des algorithmes évolutionnaires qui sont un domaine de l'informatique, qui appliquent les principes de recherche heuristique inspirés par l'évolution naturelle proposée par Darwin, en 1929 à une variété de domaines différents, particulièrement aux problèmes d'optimisation des paramètres, ainsi que des problèmes considérés dans l'Intelligence Artificielle. Toute au long de ce partie, nous décrivons ces approches un intérêt particulier est porté aux algorithmes génétiques.

La deuxième partie, quant à elle, est la classification qui permet de regrouper des objets en groupes ou classes d'objets plus homogènes. Les objets regroupés ont des caractéristiques communes, ils sont similaires, mais se distinguent clairement des objets des autres classes.

## 2. Algorithmes évolutionnaires

Les algorithmes évolutionnaires ou l'informatique évolutionnaire est un domaine de l'informatique [11]. Ils appliquent les principes de recherche heuristique inspirés par l'évolution naturelle proposée par Darwin, en 1929 à une variété de domaines différents, particulièrement aux problèmes d'optimisation des paramètres, ainsi que des problèmes considérés dans l'Intelligence Artificielle. Les algorithmes évolutionnaires rassemblent des approches différentes dans le nom, l'origine et l'implémentation [12][13][14][15][16][17][18] Tout au long de ce chapitre, nous décrivons ces approches. Un intérêt particulier est porté aux algorithmes génétiques.

### 2.1 Principe de fonctionnement

Le principe d'un algorithme évolutionnaire (AE) est schématisé à travers la (figure 2.1) Considérant un problème d'optimisation combinatoire, une population est un ensemble de points de l'espace des solutions, chacun de ces points est appelé individu. Un individu est constitué d'un patrimoine génétique qui le caractérise et le différencie des autres individus; concrètement les gènes sont les blocs élémentaires caractérisant une solution. Habituellement, un individu est pris comme une liste d'entiers s'il s'agit de problèmes combinatoires, un vecteur de nombres réels pour des problèmes numériques dans des espaces continus ou une

chaîne de nombres binaires pour des problèmes booléens. Au besoin, ces représentations peuvent être combinées dans des structures complexes [13].

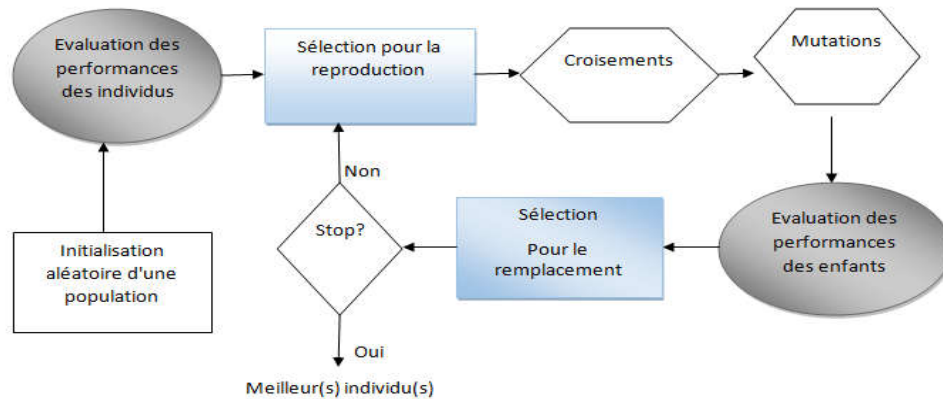


Figure2.1: Principe d'un algorithme évolutionnaire standard [13]

## 2.2 Les Algorithmes Génétiques

### 2.2.1 Définition

Plusieurs définitions ont été données pour les algorithmes génétiques. Parmi ces définitions, nous citons celle de Koza, en 1992. Ce dernier fournit une très bonne définition :

"L'algorithme génétique est un algorithme mathématique qui transforme un ensemble d'individus (population d'objets mathématiques), chacun avec une valeur de Fitness associée, en une nouvelle population (la génération suivante) en utilisant des opérations inspirées du principe darwinien de reproduction, et après des opérations génétiques naturelles". [19]

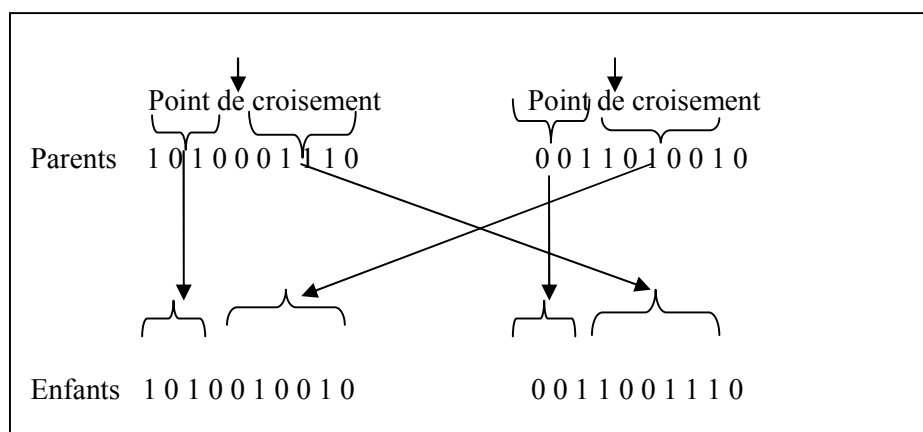
Leur but est d'obtenir une approximation de la solution à un problème NP-complet par un mécanisme d'optimisation.

### 2.2.2 Vocabulaire

Dans cette section nous introduisons quelques vocabulaires utilisés dans la mise en œuvre des algorithmes génétiques

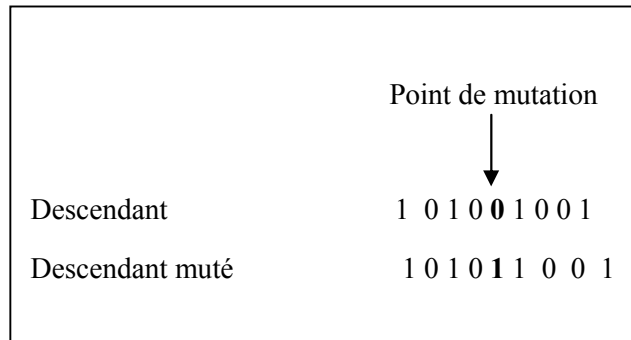
- **Gène** : Un gène est une unité d'information génétique. on appellera gène la suite de symboles qui codent la valeur d'une variable. Dans le cas général, un gène correspond à un seul symbole (0 ou 1 dans le cas binaire).
- **Génotype**: Correspond à l'ensemble des valeurs des gènes.

- **Individu:** Il représente le codage d'une solution potentielle (un élément de l'espace de recherche).
- **Chromosome:** Un chromosome est constitué de gènes, l'ensemble des chromosomes d'un individu regroupe l'intégralité de son patrimoine génétique. Généralement un AG utilise un chromosome par individu.
- **Performance:** Elle représente la mesure de la qualité (fitness) des individus basée sur l'objectif de l'optimisation et permettant de comparer les individus entre eux afin d'en déterminer plus et moins aptes.
- **Evaluation d'un individu:** Elle représente le calcul de la performance de l'individu.
- **Population:** Un ensemble fini (de taille N) d'individus.
- **Evolution:** Représente un processus d'optimisation itératif de recherche d'un (ou plusieurs) individu(s).
- **Génération:** Correspond à l'itération, mais ce terme signifie parfois la population en une certaine itération.
- **Croisement:** Il s'agit d'un opérateur de reproduction (Cross over) appliqué avec la probabilité point de croisement (pc) et qui correspond à un brassage d'information entre les individus de la population. Il consiste à échanger des parties composantes (gènes) entre deux ou plusieurs individus.



**Figure 2.2:** Exemple de croisement : Croisement à un point

- **Mutation:** Opérateur de modification d'un ou plusieurs gènes appliqué avec la probabilité point de mutation (pm) dans le but d'introduire une nouvelle variabilité dans la population.



**Figure 2.3:** Exemple de mutation : Mutation à un point

- **Sélection:** Processus du choix des individus pour la reproduction basé sur leur performance.
- **Remplacement:** Il représente le processus de formation d'une nouvelle population à partir des ensembles de parents et d'enfants, effectué le plus souvent sur la base de leur performance.

### 2.2.3 Principe

Pour mettre en œuvre un algorithme génétique, il est nécessaire de disposer :

1. Une représentation génétique du problème, c'est-à-dire un codage de solutions utilisé sous la forme de chromosomes.
2. Un mécanisme de génération de la population initiale. Ce mécanisme est indispensable pour construire une population d'individus non homogène.
3. Une fonction qui permet d'évaluer l'adaptation d'un chromosome à son environnement, ce qui offre la possibilité de comparer des individus. Cette fonction est construite à partir du critère que l'on désire optimiser. L'application de cette fonction à un élément de la population donne sa fitness.
4. Un mode de sélection des chromosomes à reproduire. Cette sélection est basée sur la reproduction et sur le codage génétique, qui stocke les informations décrivant l'individu sous forme de gènes.
5. D'opérateurs de croisement et de mutation permettant de diversifier la population au cours des générations et d'explorer l'espace d'état.
6. De paramètres qu'utilise l'algorithme : taille de la population, probabilité de croisement et de mutation, nombre total de générations. [19]

Enfin, il suffit d'appliquer l'algorithme 2.1, pour lequel nous avons les notations suivantes :

- a. -  $POP_t$  : population à la génération  $t$ ,
- b. -  $POP_{sel}$  : population sélectionnée.

---

**2.2.4. Algorithme 2.1 : Algorithme Génétique Standard (SGA : Standard Genetic Algorithm)**

---

**Début**

$t = 0$  ;

Initialisation de la population  $POP_t$  ;

Evaluation des individus de la population  $POP_t$  ;

**Tant que** (Condition de terminaison non satisfaite) **faire**

$t \leftarrow t + 1$  ;

    Sélection et copies des parents de la population  $POP_{t-1}$  dans  $POP_{sel}$  ;

    Application des opérateurs génétiques sur  $POP_{sel}$  ;

$POP_t = POP_{sel}$  ;

    Evaluation des individus de la population  $POP_t$  ;

**Fin Tant que**

Donner la meilleure solution ;

**Fin**

---

**2.2.5 Mise en œuvre d'un algorithme génétique :**

La mise en œuvre d'un algorithme génétique est réalisée suivant les étapes suivantes [20] :

1. Création d'une population initiale.
2. Evaluation des individus de la population
3. Sélection des meilleurs individus.
4. Reproduction (Croisement et mutation).
5. Formation d'une nouvelle génération

### 2.2.6 Domaines d'application des algorithmes génétiques

Les algorithmes génétiques ont été appliqués dans un grand nombre de domaines, notamment la science, l'ingénierie, les affaires, les jeux et la robotique:

- Optimisation : optimisation de fonctions, planifications, etc.
- Exemples de problèmes en général : tels que les exemples numériques et mathématiques, le design industriel tel que le problème de la déchiqueteuse, la planification des achats, des exemples de qualité audio et vidéo.
- Programmation automatique : Les algorithmes génétiques ont été utilisés pour développer des programmes informatiques pour effectuer des tâches spécifiques et pour concevoir d'autres structures informatiques, telles que des réseaux de tri.
- Enseignement des robots et des machines (Apprentissage): Les algorithmes génétiques ont été utilisés dans de nombreuses applications d'apprentissage automatique, y compris la classification et la prédiction, des algorithmes génétiques ont été utilisés dans la conception de réseaux neuronaux.
- Modèles économiques : Des algorithmes génétiques ont été utilisés pour modéliser les mécanismes d'innovation et de développement de stratégies d'appel d'offres et dans le domaine de l'émergence des marchés économiques.
- L'interaction entre évolution et apprentissage : Elle a été utilisée pour étudier l'influence mutuelle entre l'apprentissage individuel et l'évolution des espèces.
- En tant que modèles pour les systèmes sociaux: Ils ont été utilisés pour étudier des aspects de l'évolution des systèmes sociaux, tels que l'évolution de la coopération et l'évolution de la communication, le comportement de suivi de piste chez les fourmis. [22-23]

### 2.2.7 Avantages et inconvénients

Les algorithmes génétiques possèdent plusieurs avantages qui permettent de les utiliser dans des domaines multiples mais ils ont aussi des inconvénients qui imposent des restrictions dans leur utilisation, nous citons quelques-uns.

### 2.2.7.1 Avantages

- En général les algorithmes génétiques accélèrent le parcours d'un vaste ensemble de solutions. Et les mauvaises solutions ne sont pas prises en compte, elles sont éliminées afin de ne pas affecter la solution optimale.
- Grande efficacité pour résoudre tous les problèmes d'optimisation qui peuvent être décrit avec le codage de chromosomes.
- Traitent des problèmes très complexes d'une manière très efficace et permettant facilement de trouver les bonnes solutions, cette caractéristique a beaucoup aidé dans les domaines avec un grand nombre de paramètres et dont le temps de réponse doit être réduit (obtenir de bonnes solutions en quelques itérations seulement)

### 2.2.7.2 Inconvénients

- Trouver les bonnes solutions pour des problèmes avec un grand nombre de paramètres n'est pas toujours assuré par les algorithmes génétiques car il est très lent.
- les algorithmes génétiques peut converger vers les optimums locaux et ne pas trouver un optimum global. Il s'agit d'une technique d'intelligence artificielle, donc l'algorithme génétique ne peut pas assurer toujours les mêmes résultats en terme de temps. Cette propriété limite leur utilisation dans les applications temps réel. [19]

## 3. classification

### 3.1 Définition

La classification consiste à organiser un ensemble de données en un ensemble fini de classes (clusters) selon un ou plusieurs critère(s) de classification à l'aide d'un classifieur [24], de sorte que les éléments à l'intérieur d'une même classe soient similaires, et les éléments appartenant à deux classes différentes soient différents.

La figure (2.5) illustre le principe de la classification. Les données sont représentées par des points (vecteurs) dans un espace à  $n$  dimensions (dans notre exemple  $n = 2$ ). En sortie de la classification, nous obtenons  $m$  classes (ou familles) de points selon les critères donnés et le classifieur choisi (dans notre exemple,  $m = 3$ ).

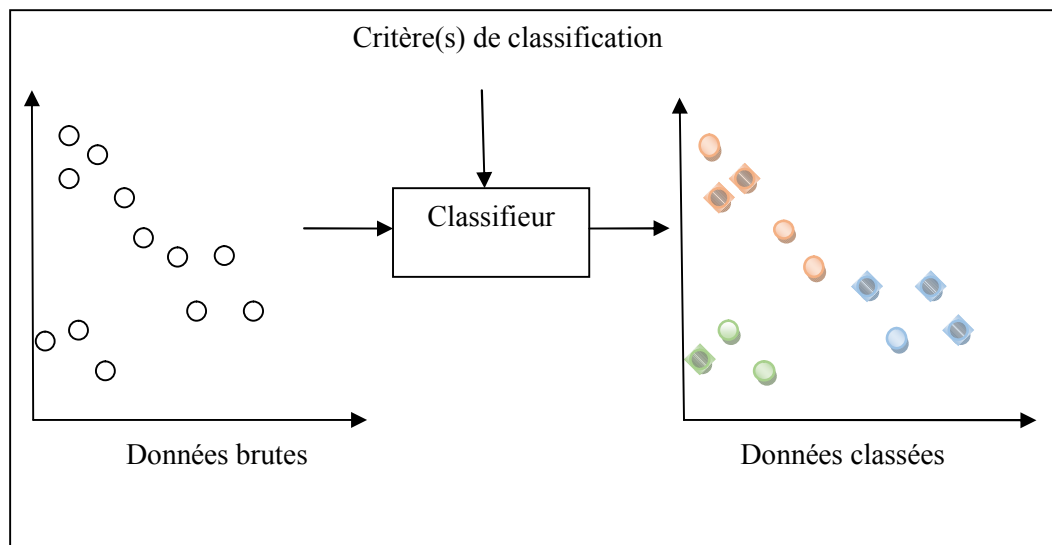


Figure 2.4: Principe de la classification [24]

## 3.2 Méthodes de classification

### 3.2.1 Classification supervisée

Classification supervisée fournit une vue de groupée selon des catégories prédéfinies. C'est-à-dire l'assignation des éléments aux classes. Cela signifie que nous devons d'abord déterminer le nombre et la nature des classes que nous voulons utiliser.

#### 3.2.1.1 Méthodes de classification supervisée

De nombreux algorithmes d'apprentissage sont adaptés au problème de la classification supervisée, on peut les citer : les Séparateurs à Vastes Marges (SVM), les réseaux de neurones, les méthodes des  $k$  plus proches voisins, les arbres de décision ... etc.

Parmi les plus célèbres algorithmes dans classification supervisée on cite  $K$ -plus proches voisins.

##### 3.2.1.1.1 Algorithme $K$ -plus proches voisins

L'algorithme des  $k$ -plus proches voisins est un des algorithmes de classification les plus simples. Le seul outil dont on a besoin est une distance entre les éléments que l'on veut classifier. Si on représente ces éléments par des vecteurs de coordonnées, il y a en général pas mal de choix possibles pour ces distances, partant de la simple distance usuelle (euclidienne) en allant jusqu'à des mesures plus sophistiquées pour tenir compte si nécessaire de paramètres

non numériques comme la couleur, la nationalité, etc. Afin de trouver les K plus proches d'une donnée à classer on peut choisir la distance euclidienne. [25]

---

### 3.2.1.1.2 Algorithme 2.2: Algorithme de K-NN

---

Début Algorithme

Données en entrée :

- un ensemble de données D.
- une fonction de définition distance d.
- Un nombre entier K

Pour une nouvelle observation X dont on veut prédire sa variable de sortie y Faire :

1. Calculer toutes les distances de cette observation X avec les autres observations du jeu de données D
2. Retenir les K observations du jeu de données D les proches de X en utilisation le fonction de calcul de distance d
3. Prendre les valeurs de y des K observations retenues :
  - a. Si on effectue une régression, calculer la moyenne (ou la médiane) de y retenues
  - b. Si on effectue une classification, calculer le mode de y retenues
4. Retourner la valeur calculée dans l'étape 3 comme étant la valeur qui a été prédite par K-NN pour l'observation X.

Fin Algorithme

---

#### 3.2.1.1.3 Avantages

- Simplicité, pas d'apprentissage d'un modèle
- Bonnes performances en général

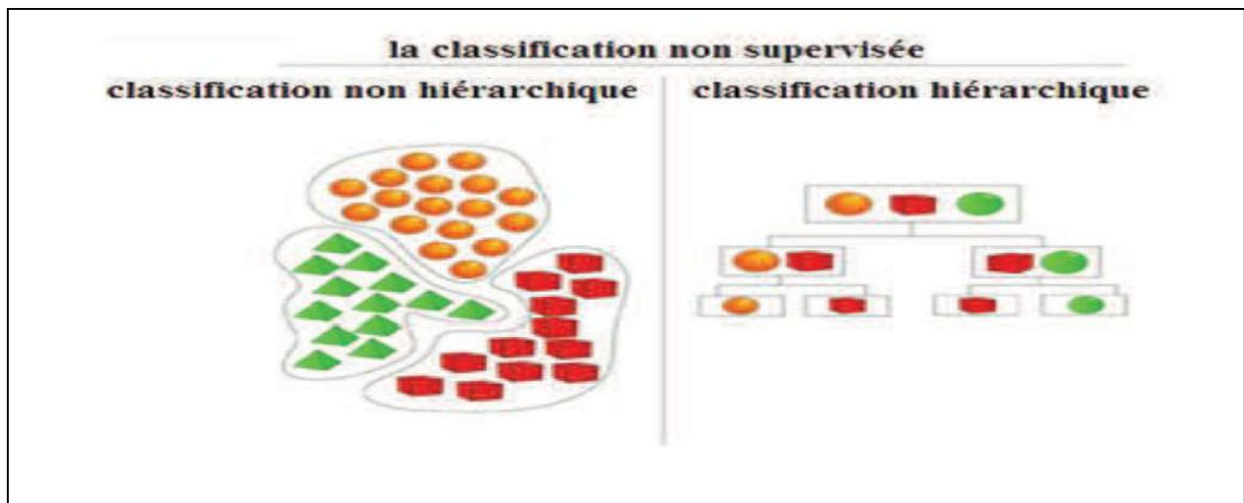
#### 3.2.1.1.4 Limites

- Paramétrage difficile (choix de la taille du voisinage)
- Nécessité de garder sous la main la base de données

## 3.2.2 Classification non supervisée

Regroupant des éléments ayant les mêmes propriétés statistiques, géométriques ou linguistiques. Elle utilise un critère de regroupement qui peut être basé sur des distances entre objets (K-Means) ou sur des appartenances floues (FCM, PCM et FPCM). Dans ce type de classification le nombre de classes, inconnu a priori, est déduit directement des données.

On distingue aussi les approches de classification non hiérarchiques et les méthodes de classification hiérarchiques.



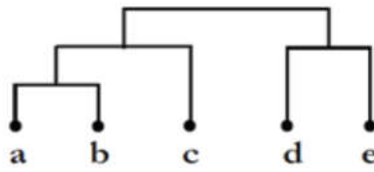
**Figure 2.6 :**Les deux types de clustering hiérarchique/non hiérarchique [26]

### a. Méthode non hiérarchique :

Cette méthode produit directement une partition en un nombre fixé de classes. Regrouper  $n$  individus en  $k$  classes de telle sorte que les individus d'une même classe soient le plus semblables possible et que les classes soient bien séparées. [26]

### b. Méthode hiérarchique :

La classification hiérarchique consiste à effectuer une suite de regroupements en classes de moins en moins fines en agrégeant à chaque étape les objets ou les groupes d'objets les plus proches. Elle fournit ainsi un ensemble de partitions de l'ensemble d'objets [27]. Cette approche utilise la notion de distance, qui permet de refléter l'homogénéité ou l'hétérogénéité des classes. Ainsi, on considère qu'un élément appartient à une classe s'il est plus proche de cette classe que de toutes les autres. La figure suivante est une illustration du principe des méthodes hiérarchiques [26]. Dans cette figure suivante, on représente la suite de partition d'un ensemble  $\{a, b, c, d, e\}$



**Figure 2.7:** Partition hiérarchique[26]

Dans le cadre de ce travail nous n'utiliserons que la méthode de classifications : K-means

### 3.2.2.1 K-Means

L'algorithme des K-Means appelé aussi algorithme des centres mobiles est l'algorithme de clustering le plus connu et le plus utilisé, tout en étant très efficace et simple, il a été introduite par J. McQueen [28] en 1971 et mise en œuvre sous sa forme actuelle par E. Forgy [29], il est un outil de classification classique qui permet de répartir un ensemble de données en  $K$  classes homogènes.

#### 3.2.2.1.1 Principe

On suppose qu'il existe  $K$  classes distinctes. On commence par désigner  $K$  centres de classes  $\mu_1, \dots, \mu_K$  parmi les individus. Ces centres peuvent être soit choisis par l'utilisateur pour leur "représentativité", soit désignés aléatoirement. On réalise ensuite itérativement les deux étapes suivantes :

- Pour chaque individu qui n'est pas un centre de classe, on regarde quel est le centre de classe le plus proche. On définit ainsi  $K$  classes  $C_1, \dots, C_k$ , où  $C_i = \{\text{ensemble des points les plus proches du centre } \mu_i\}$ .
- Dans chaque nouvelle classe  $C_i$ , on définit le nouveau centre de classe  $\mu_i$  comme étant le barycentre des points de  $C_i$ .

L'algorithme s'arrête suivant un critère d'arrêt fixé par l'utilisateur qui peut être choisi parmi les suivants : soit le nombre limite d'itérations est atteint, soit l'algorithme a convergé, c'est à dire qu'entre deux itérations les classes formées restent les mêmes, soit l'algorithme a "presque" convergé, c'est à dire que l'inertie intra-classe ne s'améliore quasiment plus entre deux itérations.

### 3.2.2.1.2 Algorithme 2.3: L'algorithme de k-means

**Etape 1 :** Choisir aléatoirement les  $C$  centres initiaux  $v_1, v_2, \dots, v_c$  à partir des points  $\{x_1, x_2, \dots, x_N\}$

**Répéter**

**Etape 2 :** Assigner  $x_j, j=1,2,\dots, N$  au cluster  $Z_i, i=1,2,\dots,C$

Si

$\|x_j - v_i\| \leq \|x_j - v_p\|, p = 1,2,\dots, C$  avec  $j \neq p$

Choix aléatoire si conflit

**Etape 3:** Calculer les nouveaux centres :

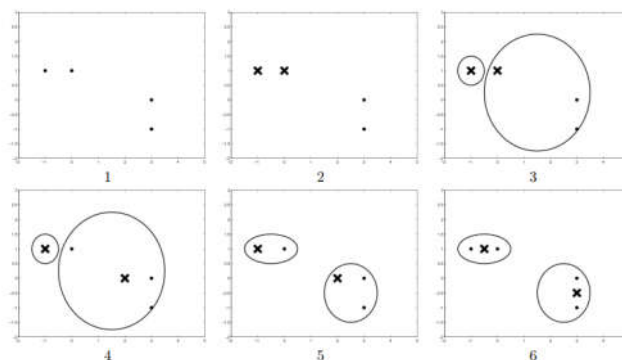
$$v_i^* = \frac{1}{N_i} \sum_{x_j \in Z_i} x_j, i = 1, 2, \dots, C$$

$N_i$  est le nombre d'éléments dans le cluster  $Z_i$

**Jusqu'à ce que**  $v_i^* = v_i, \forall i = 1, 2, \dots, C$ .

### 3.2.2.1.3 Exemple

La figure 2.8 illustre l'algorithme sur un exemple où quatre points  $a (-1,1), b (0,1), c (3,0)$  et  $d (3,-1)$  doivent être classés en 2 classes. On remarque sur cet exemple que bien que à l'initialisation les centres de classes sont mal répartis, l'algorithme a convergé en retrouvant les "vraies" classes.



**Figure 2.8 :** Une illustration de l'algorithme K-means[28].

- (1) On dispose de 4 points à classer en deux classes.
- (2) A l'initialisation, deux de ces points sont choisis comme centres de classe.
- (3) Deux classes sont créées en regroupant les autres points en fonction du centre de classe le plus proche.
- (4) On définit les nouveaux centres de classe comme étant le barycentre des classes nouvellement créées.
- (5) On regroupe à nouveau les points.
- (6) On définit les nouveaux centres de classes. A l'étape suivante rien ne change, l'algorithme a convergé.

#### **3.2.2.1.4 Avantages et contraintes de l'algorithme K-means**

L'avantage de cet algorithme est qu'il a une :

- Bonne résistance aux données erronées.
- Flexibilité avec tout type de distance.

Cet algorithme a plusieurs points faibles [30] :

- Sensibilité à l'initialisation du fait que les centres initiaux des différentes classes sont choisis aléatoirement.
- K-means est fortement sensible au bruit.
- Convergence vers un minimum local.

### **Conclusion**

Dans ce chapitre, nous avons présenté de manière générale les algorithmes évolutionnaires, et en particulier les algorithmes génétiques. et dans la deuxième partie nous avons présenté de manière générale la classification, et en particulier le k-means.

Dans le chapitre suivant nous montrerons comment combiné les deux algorithmes sus cités afin d'obtenir une solution optimale et qui répond aux exigences de l'utilisateur.

# Chapitre 3

## Étude Conceptuelle

## 1. Introduction

Après avoir défini les concepts théoriques de base sur l'IdO, les algorithmes génétique et le k-means dans les deux premiers chapitres, nous passons à la deuxième partie de notre travail qui consiste à présenter progressivement la conception et l'implémentation du système. Dans ce chapitre nous décrivons une approche de conception et de développement d'applications ou de systèmes d'information à base de services IdO. La première section présente l'approche proposée globale, et la deuxième section présente une approche proposée détaillée de ce travail.

## 2. Objectif du projet et méthodologie

Afin de bénéficier des avantages des applications IdO, nous souhaitons créer une interface permettant à l'utilisateur d'interagir avec les services IdO.

Notre objectif est de trouver le service parfait pour l'utilisateur en fonction de ses besoins spécifiques. Le système à concevoir permet à ce service d'être fourni par le groupe de services Internet des objets. Afin de réaliser ce système, nous avons deux parties principales. Dans la première partie, nous appliquons l'un des algorithmes de classification, qui est le K-means pour trouver des services qui ont la même fonctionnalité. Et dans la deuxième partie, nous appliquons l'un des algorithmes évolutionnaire qui est l'algorithme génétique, nous appliquons ce l'algorithme sur chaque cluster que nous avons afin de trouver la service optimal.

Et le deuxième objectif de ce travail est de créer une interface qui garantit à l'utilisateur de saisir ses besoins qui sont traités par notre système pour rendre aux utilisateurs le meilleur service

## 3. Méthode de conception adoptée

Les méthodes de conception définissent une représentation souvent graphique qui permet d'une part de manipuler facilement les modèles, et d'autre part de communiquer et d'échanger des informations entre les différents objets. La meilleure représentation doit être claire et facile à comprendre, pour cela nous choisissons des diagrammes UML pour la représentation des différents modèles de notre système, car UML offre différentes représentations du système selon différentes vues grâce aux diagrammes.

### 3.1 UML (Unified Modeling Language)

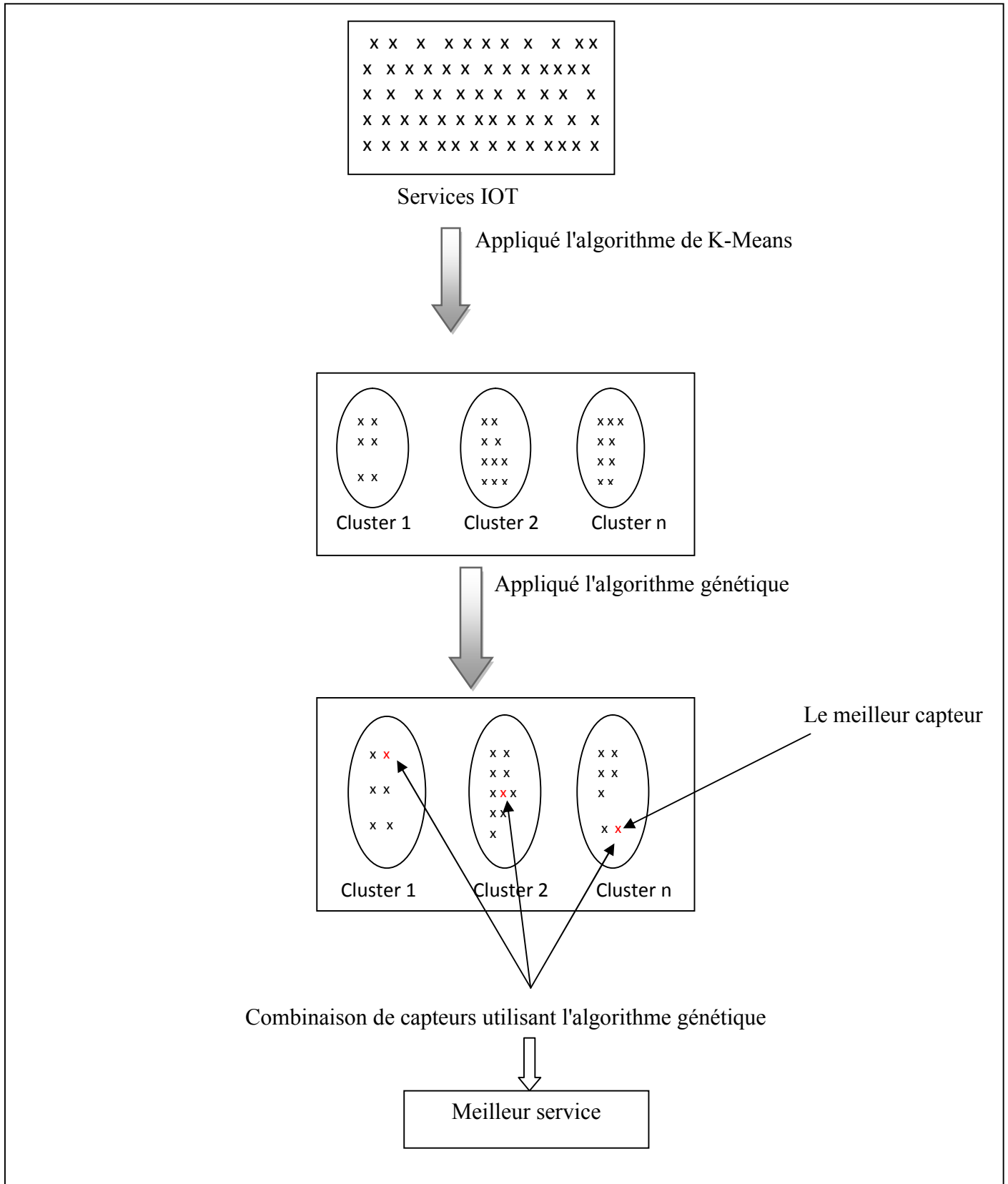
UML est un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, à spécifier et documenter des systèmes, à esquisser des architectures logicielles. Il s'articule autour de plusieurs types de diagrammes, chacun d'eux étant dédié à la représentation des concepts particuliers d'un système logiciel mais, on va représenter seulement le diagramme qui est utilisés dans notre projet :

**Diagrammes de séquence** : représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation.

## 4. Architecture générale du système:

### 4.1 Phase 1

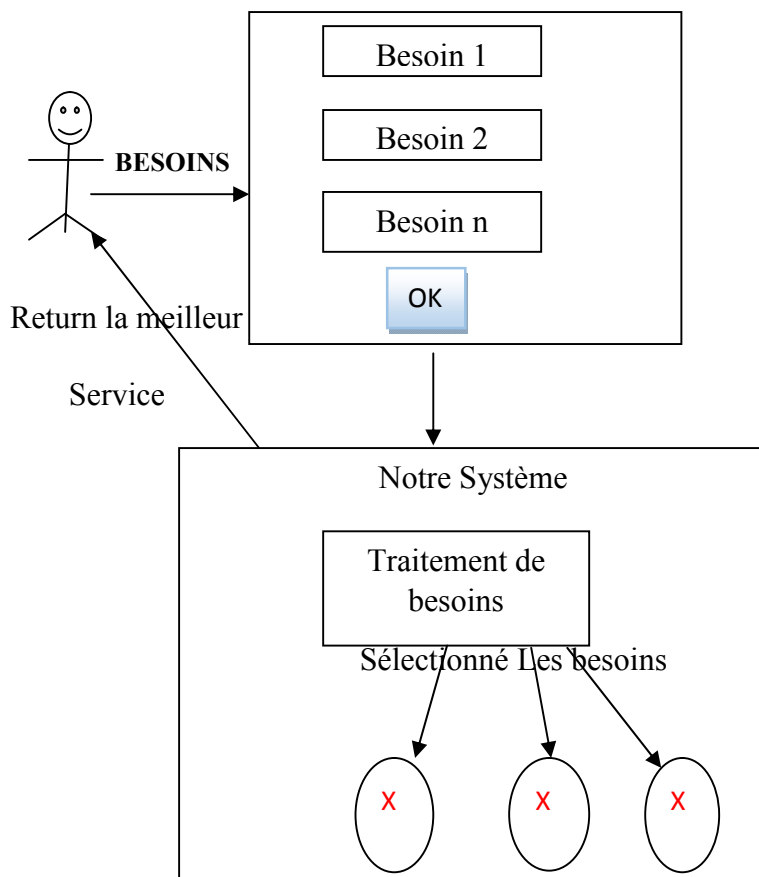
Dans cette section nous présentons l'architecture générale de notre système, que nous avons un ensemble de services ido, nous appliquons l'algorithme k-means sur ces services en raison de trouver les services qui ont la même fonctionnalité, puis nous appliquons l'algorithme génétique sur les clusters s'obtient par k-means.



**Figure 3.1 :** les principales étapes pour réaliser la première phase

#### 4.2. Phase 2:

Dans cette phase nous présentons une architecture générale de l'interaction qui se produit entre l'utilisateur et notre système où l'utilisateur met dans l'interface son besoin et notre système lui rend le service dont il a besoin:



**Figure 3.2:**L'interaction de l'utilisateur avec notre système

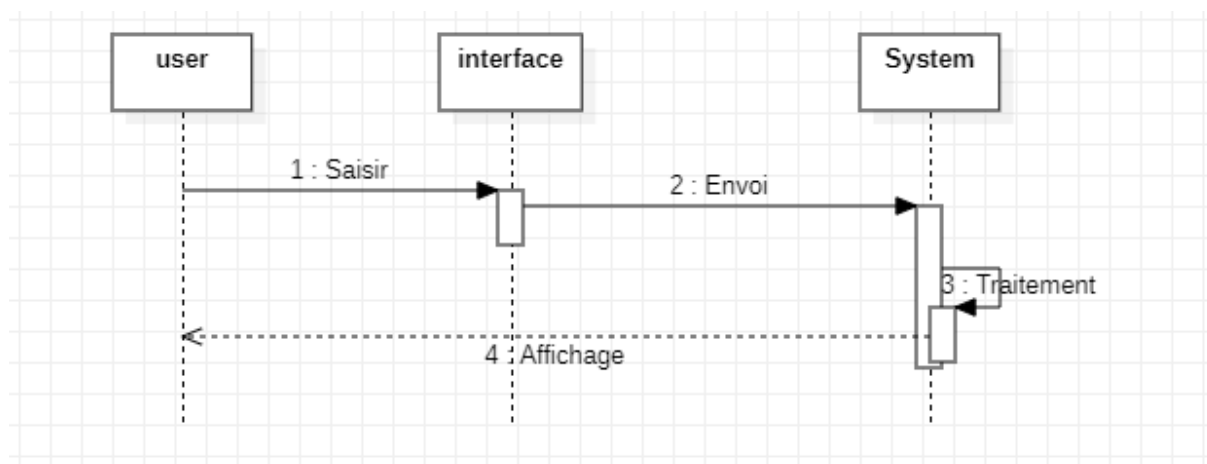


Figure 3.3 : Diagramme de séquence de l'interaction de l'utilisateur avec notre système

### 5. Architecture détaillé du système

Dans cette approche nous avons plus expliqué les deux algorithmes que nous utilisons en objectif pour réaliser ce thème.

Nous avons un ensemble de données qui contient 100 services IoT :

Tableau 3.1: Notre dataset

Id service	Temps de réponse	Succès	prix	Batterie	Description	Input	Output
001	302.75	90	75	9000	CAPTEUR DE TEMPERATURE MCP9808 HAUTE PRECISION I2C	Etiquette	Photo
002	482	95	70	9000	CAPTEUR DE TEMPERATURE ET D'HUMIDITE SI7021	Etiquette	Alarme
...							
026	256	58	73	8000	CAPTEUR DE LUMIERE AMBIANTE TEMT6000	Photo rgb	Photo
027	408	99	100	9500	CAPTEUR DE LUMIÈRE ANALOGIQUE ALS-PT19	Photo rgb	Alarme

028	173	84	40	4000	CAPTEUR DE LUMIERE/LUX I2C - VEML7700	Photo rgb	Rapport visuel
029	320.48	47	70	8000	CAPTEUR DE LUMIERE LUMINOSITE HAUTE PRECISION - TSL2591	Photo rgb	Vidéo
...							
051	227	85	83	8500	DÉTECTEUR DE MOUVEMENT -EE 820-140°	35	Photo
052	511.25	96	85	9000	DÉTECTEUR DE MOUVEMENT Theben 1010505 theLuxa S180 WH	35	Alarme
...							
081	130.33	100	67	6500	CAPTEUR QUALITÉ D'AIR Figaro TGS2612- D00	46	Photo
100	146.08	43	92	9500	CAPTEUR QUALITÉ D'AIR cmx	49	Rapport imprimé

Avant d'appliquer l'algorithme k-means, nous effectuons une normalisation sur notre dataset:

**Exemple de normalisation des inputs :****Tableau 3.2:**Normalisation des inputs

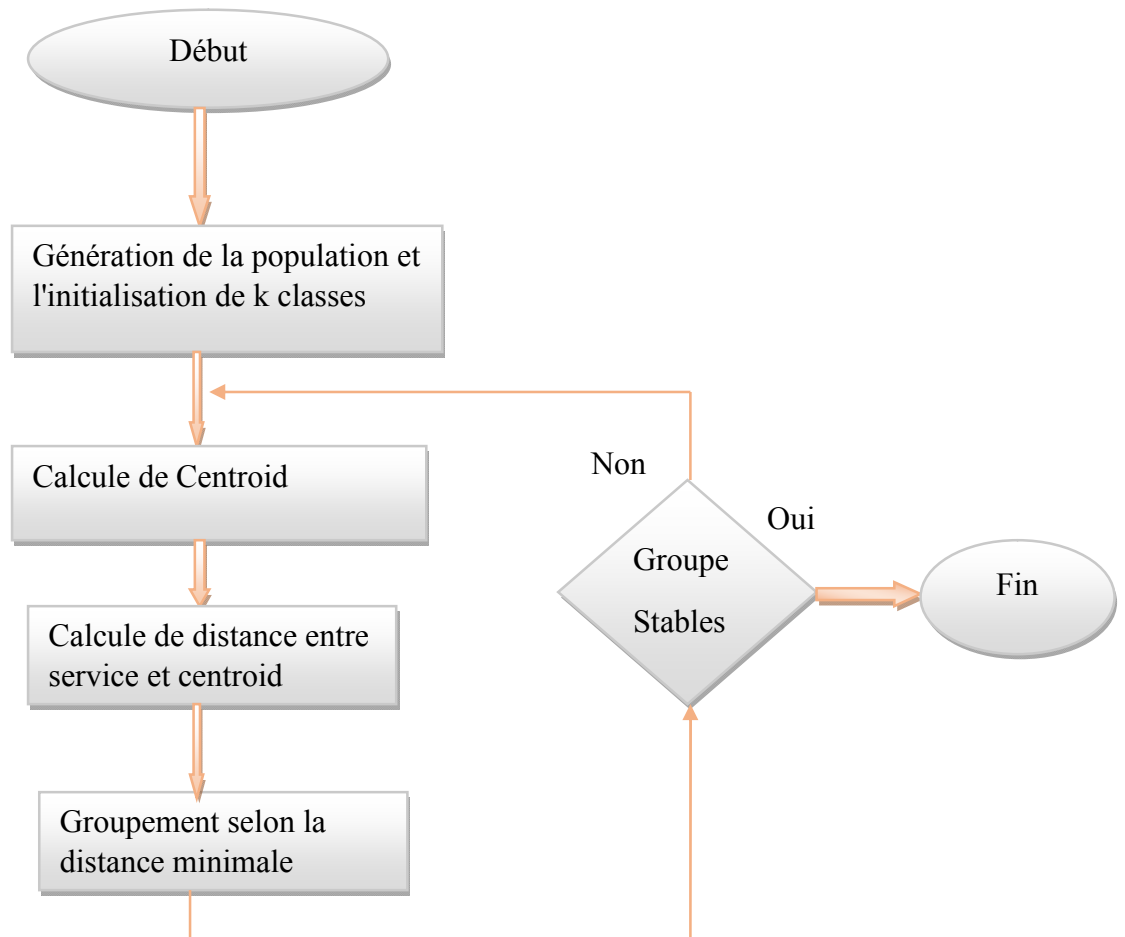
Input	Normalisation
Etiquette	1
Rapport imprimé	2
Matière liquide	3
Matériau solide	4
Photo	5
Photo rgb	25
Video rgb	26
Photo png	27
Camera	28

**La normalisation des Output :****Tableau 3.3:** Normalisation des Output

Output	Normalisation
Alarme	6
Rapport visuel	7
Rapport imprimé	2
Photo	5
Vidéo	8

## 5.1 K-Means:

### 5.1.1 Organigramme de l'algorithme de k-means



**Figure 3.4** Organigramme de l'algorithme de k-means

### 5.1.2 Principe :

- L'algorithme consiste à grouper les services selon leur similarité.
- Il permet de regrouper en K clusters distincts les services.
- l'algorithme choisit arbitrairement k points comme centres «initiaux » des k groupes.
- L'étape suivante consiste à calculer la distance entre chaque service (point) et les k centres, la plus petite distance est retenue pour inclure ce service dans le groupe ayant le centre le plus proche.
- Une fois tous les services groupés, on aura k sous-nuages (cluster) disjoints du nuage total.
- Pour chaque groupe, l'algorithme calcule le nouveau centre de gravité.
- L'algorithme s'arrête lorsque les groupes construits deviennent stables.

### 5.1.3 Notion de similarité

Pour pouvoir regrouper un jeu de données en cluster distincts, l'algorithme K-Means a besoin d'un moyen de comparer le degré de similarité entre les différentes observations. Ainsi, deux données qui se ressemblent, auront une distance de dissemblance réduite, alors que deux objets différents auront une distance de séparation plus grande.

Les littératures mathématiques et statistiques regorgent de définitions de distance, dans notre application nous utilisons La distance Euclidienne :

- **La distance Euclidienne** : C'est la distance géométrique qu'on apprend au collège. Soit une matrice  $\mathbf{X}$  à  $n$  variables quantitatives. Dans l'espace vectoriel  $E^n$ . La distance euclidienne  $D$  entre deux observations  $\mathbf{x}_1$  et  $\mathbf{x}_2$  se calcule comme suit :

$$D(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{\sum_{j=1}^n (x_{1j} - x_{2j})^2}$$

## 5.2 Algorithme génétique:

### 5.2.1 Organigramme de l'algorithme génétique:

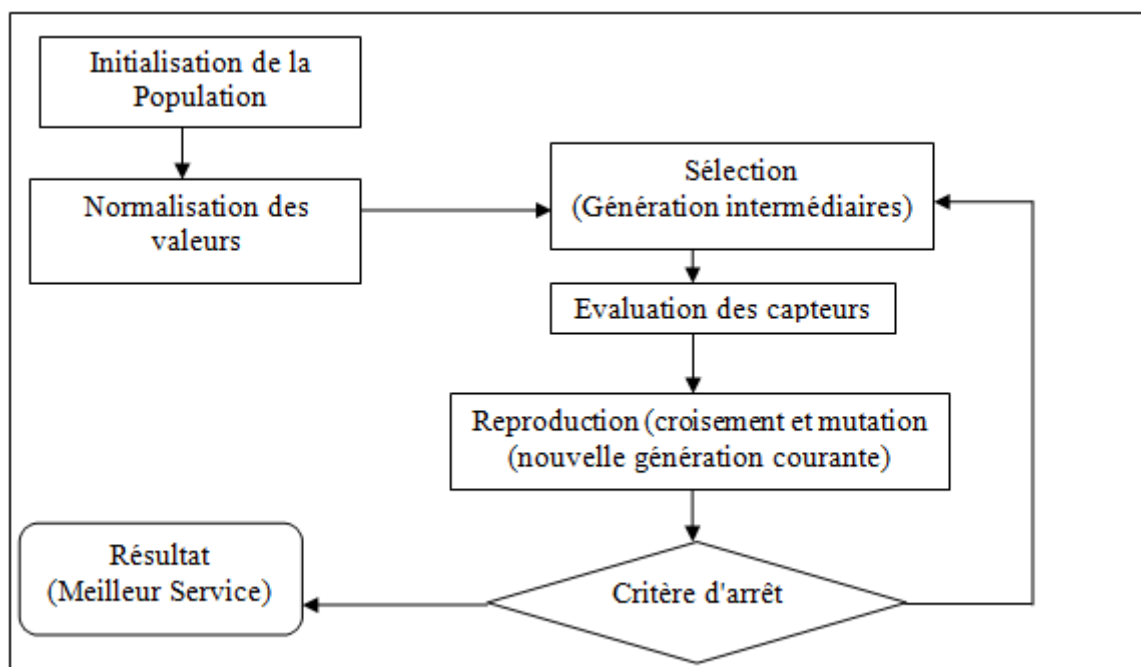


Figure 3.4 Organigramme de l'algorithme génétique

### 5.2.1 La normalisation:

Pour appliquer l'algorithme génétique dans notre travail, nous devons d'abord faire une normalisation de nos caractéristiques pour les faire entre 0 et 1.

Cette formule nous permet de faire la normalisation:

$$Nqos_p^k = \begin{cases} 1, & \text{If } (qos_p^{max} - qos_p^{min}) = 0 \\ \text{Else} \\ \frac{qos_p^{max} - qos_p^k}{qos_p^{max} - qos_p^{min}} & \text{if } p \text{ is a negative attribute} \\ \frac{qos_p^k - qos_p^{min}}{qos_p^{max} - qos_p^{min}} & \text{if } p \text{ is a positive attribute} \end{cases}$$

#### Notions:

P : Capteur.

K: le nombre de cluster.

$qos_p^{max}$  : La valeur maximale de chaque capteur.

$qos_p^{min}$  : La valeur minimale de chaque capteur.

p est un attribut négatif : la valeur est bonne quand elle est élevée.

p est un attribut positif : la valeur est bonne quand elle est faible.

#### Exemple de normalisation:

- Nous avons ce service :  
000,408.21, 98, 60, 6500, CAPTEUR DE TEMPERATURE SANS CONTACT  
MLX906166 8V, 5,2
- L'indice de ce service est : 000
- La normalisation de temps de réponse:
  - max de temps de réponse dans le cluster 1 est : 3321.4
  - min de temps de réponse dans le cluster 1 est : 50.0
  - Le max - min = 3271.4
  - la valeur du temps de réponse de noter capteur est : 408.21
  - la valeur de normalisation est : 0.10949746285993764

- La normalisation de succès:
  - Max de succès dans le cluster 1 est : 100
  - Min de succès dans le cluster 1 est : 72
  - Le max - min = 28
  - La valeur du succès de noter capteur est : 98
  - La valeur de normalisation est : 0.07142857142857142
  
- La normalisation de prix:
  - Max de prix dans le cluster 1 est : 85
  - Min de prix dans le cluster 1 est : 30
  - Le max – min = 55
  - la valeur du prix de noter capteur est : 60
  - la valeur de normalisation est : 0.5454545454545454
  
- La normalisation de la batterie:
  - Max de batterie dans le cluster 1 est : 9000
  - Min de batterie dans le cluster 1 est : 3000
  - Le max - min = 6000
  - La valeur de la batterie de noter capteur est : 6500
  - La valeur de normalisation est : 0.4166666666666667

### 5.2.2 La fonction dévaluation:

Pour calculer la fitness nous utilisons cette formule :

$$fu(CS_i^k) = \sum_{p=1}^q w_p * Nqos_{ip}^k$$

#### Notions:

$w_p$ : Le poids que l'on multiplie dans les capteurs tel que :  $\sum w_p = 1$

Le poids de tempe de réponse est : 0.2

Le poids de succès est : 0.1

Le poids de prix est : 0.4

Le poids de la batterie est : 0.3

**Exemple de calculer la fitness :**

Calculer la fitness de capteur 000:

Tempe de réponse:  $0.10949746285993764 * 0.2$

Succès :  $0.07142857142857142 * 0.1$

Prix :  $0.5454545454545454 * 0.4$

Batterie :  $0.4166666666666667 * 0.3$

Fitness du capteur est : 0.37222416789666285

**Exemple d'application de l'algorithme génétique:**

L'utilisateur a besoin d'un service qui a quatre capteurs

- Temps de réponse : 800
- Succès : 87%
- Prix : 300
- Batterie : 20 000
- La fitness de ce service est : 0,508945
- Le résultat donné par l'algorithme génétique ce sont des meilleurs services:
  - 022, 132, 72, 58, 5500, CAPTEUR DE TEMPERATURE NUMERIQUE ET D'HUMIDITE AM2560, 5,6
  - 043, 170, 100, 65, 3500, CAPTEUR DE LUMIERE AMBIANTE TEMT8000, 28,7
  - 058,430.5, 95, 95,7000, DÉTECTEUR DE MOUVEMENT Goobay 95172, 36,6
  - 084,47.27, 85,78, 4000, CAPTEUR QUALITÉ D'AIR Dioxyde de carbone ,46,8
- Total:779.77, 88%, 296,20000 fitness total = 0.514725

**Conclusion**

Tout au long de ce chapitre, nous avons exposé les différentes fonctionnalités auxquels doit répondre le système à réaliser ainsi que la solution que nous avons adoptée pour la satisfaction de ces exigences stipulées. Dans le chapitre suivant, nous allons aborder l'implémentation de cette solution proposée tout en évoquant l'ensemble des choix techniques et logiciels qui nous ont aidés dans la réalisation de cette tâche.

# Chapitre 4

## Implémentation

## 1. Introduction

Notre travail à réaliser comporte deux parties, une partie théorique, où les notions et les concepts de base du thème sont présentés, et une partie pratique où les différentes implémentations requises sont détaillées. Dans le chapitre précédent, nous avons présenté les composants et les idées que nous allons mettre en place d'un point de vue conceptuel. Dans ce chapitre, nous présentons d'abord une étude technique dans laquelle nous définissons l'environnement logiciel utilisé pour construire notre application l'implémentation et la réalisation de notre système. Nous présentons également les différents outils et les technologies utilisés pour le développer. Enfin, nous présentons et expliquons les différents résultats acquis.

## 2. Environnement de Travail

### 2.1. Environnement matériel

Notre application est testée sur un ordinateur ayant les caractéristiques suivantes :

**Tableau 4.1:** Caractéristiques du pc utilisé pour le projet

Configuration De la machine	Processeur	Intel(R) Core(TM) i3-5005U CPU@ 2.00GHz 2.00 GHz
	Mémoire RAM	4.00 Go
	Nom du pc	Lenovo
Système d'exploitation	Windows 10 Enterprise © Microsoft Corporation, 2017 Edition 64bit	

### 2.2. Environnement logiciel

Python est un langage de programmation puissant et facile à apprendre. Il dispose de structures de données de haut niveau et permet une approche simple mais efficace de la programmation orientée objet. Il a été créé par Guido van Rossum entre 1985 et 1990.

Python est un langage idéal pour l'écriture de scripts et le développement rapide d'applications dans de nombreux domaines et sur la plupart des plateformes. [31]

La figure 4.1 illustre le logo du langage Python :



**Figure 4.1:** Logo de Python

### 2.3 Framework de programmation

Nous avons utilisé le framework Anaconda car c'est un distributeur libre et open source du langage de programmation Python appliqué au développement d'applications dédiées à la science de données et à l'apprentissage automatique [31]. Le principal avantage de l'utilisation d'anaconda est que, anaconda est comme un point central pour les bibliothèques qui auraient besoin pour le traitement de données, l'analyse prédictive et les calculs scientifiques. [32]

La figure 4.2 illustre le logo du framework de programmation Anaconda :



**Figure 4.2:** Logo d'anaconda

Nous avons utilisé l'environnement de développement Spyder (nommé Pydee dans ses premières versions). C'est un environnement de développement pour Python. Libre (Licence MIT) et multiplateforme (Windows, Mac OS, GNU/Linux) [33] qui contient nombreuses bibliothèques d'usage scientifique : Matplotlib, NumPy, SciPy et IPython.

La figure 4.3 présente le logo de l'environnement de développement Spyder :



**Figure 4.3:** Logo de l'environnement de développement Spyder

## 2.4. Bibliothèques utilisées

**NumPy** : est le paquet du traitement de tableau [34] pour les nombres, les chaînes de caractères, les enregistrements et les objets.

**Pandas**: est une bibliothèque libre, écrite pour le langage de programmation Python permettant la manipulation et l'analyse des données [35]. Elle propose en particulier des structures de données et des opérations de manipulation de tableaux numériques.

**Matplotlib** : Matplotlib est une bibliothèque complète pour créer des visualisations statiques, animées et interactives en Python

**Scikit-learn** : Librairie Python pour effectuer de l'apprentissage automatique. Il inclut la plupart des méthodes de classification. [34]

### 3. Définition d'ensemble de données utilisé et description des variables

#### 3.1 Définition du jeu de données utilisé

Il s'agit d'un ensemble de données sur les services IoT, il créé par nous à partir d'ensemble de services web, il se compose de plusieurs qualité de ces services, description, entrée et sortie de chaque service.

Les variables sont les suivants :

1. Temps de réponse
2. Coût
3. Batterie
4. Succès
5. Description
6. Entrée

#### 3.2. Description des variables

**Tableau 4.2** : Description des variables d'ensemble de données

<b>Variable</b>	<b>Description</b>
Temps de réponse	Temps pris pour envoyer une demande et recevoir une réponse (ms)
Coût	Le prix du service
Batterie	La batterie du service IoT.
Succès	Nombre de réponse / nombre de messages de requête (%)
Description	Présentation de service
Entrée	Informations qui entrent dans le service
Sortie	Le résultat des entrées

### 3. Description de l'application

#### 3.1 Code k-means :

Avant d'appliquer l'algorithme k-means, nous devons importer un ensemble de bibliothèques : Pandas, NumPy et Pyplot.

```

1 import pandas as pd
2 import numpy as np
3 from sklearn.cluster import KMeans
4 import matplotlib.pyplot as plt
5
6 Y = pd.read_csv('data1.csv')
7

```

La ligne `y = pd.read_csv ("data1.csv")` : permet de lire notre dataset.

**data1** : c'est le nom du dataset.

La sortie de cette ligne dans la console est :

id	sponse Tir	iccessabili	prix	batterie	Description	input	output
0	408.21	98	60	6500	CAPTEUR DE TEMPERATURE SANS CONTACT MLX906166 8V	5	2
1	302.75	90	75	9000	CAPTEUR DE TEMPERATURE MCP9808 HAUTE PRECISION I2C	1	5
2	482	95	70	9000	CAPTEUR DE TEMPERATURE ET D'HUMIDITE SI7021	1	6
3	3321.4	96	65	9000	CAPTEUR DE TEMPERATURE ET D'HUMIDITE SENSIRON SHT31-D	1	7
4	126.17	100	85	9000	CAPTEUR DE TEMPERATURE THERMISTANCE EPOXY 10K - 3950 NTC	1	8
5	107	95	70	8000	CAPTEUR DE TEMPERATURE ETANCHE DS18B20	1	2

Le code qui permet de construire le k-means:

```

10 s={}
11 for i in range(1,101):
12     s[i]= i
13     print(s[i])
14 data = pd.DataFrame(Y,index=s)
15
16
17
18 X = Y.iloc[:,6:8]
19 # k means
20 km = KMeans(n_clusters=4)
21 km.fit(X)
22

```



Résultat après application de l'algorithme k-means:

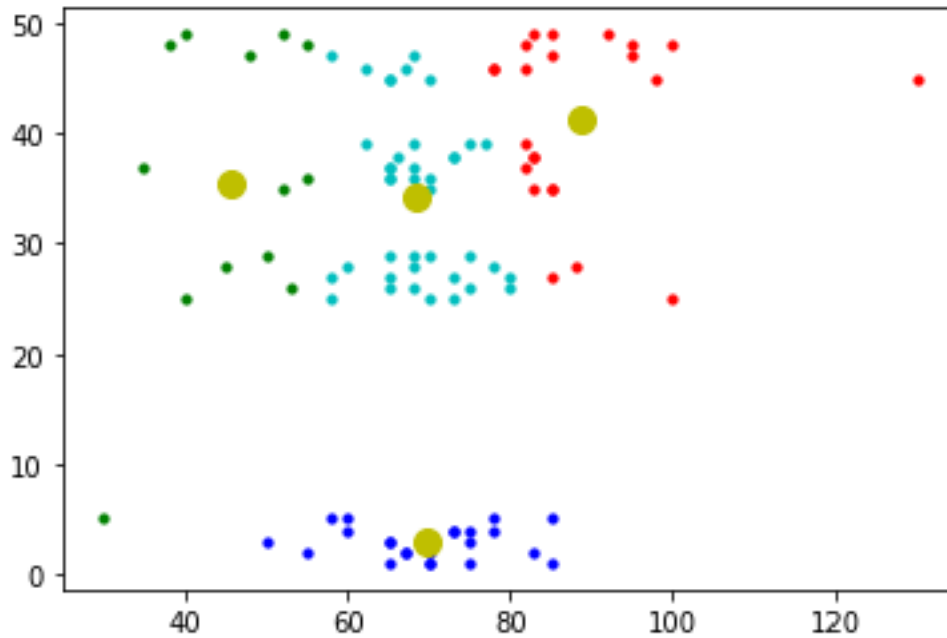


Figure4.5: Résulta de k-means

Puis on met l'étiquette des k-means dans Y:

```
# put the labels in the Y dataframe  
Y['labels'] = km.labels_
```

Le résultat de ce code est comme suite :

Y - DataFrame

Index	Description	input	output	labels
0	FEUR DE TEMPERATURE SANS CONTACT 306166 8V	5	2	1
1	FEUR DE TEMPERATURE MCP9808 HAUTE PRECISION I2C	1	5	1
2	FEUR DE TEMPERATURE ET D'HUMIDITE SI7021	1	6	1
3	FEUR DE TEMPERATURE ET D'HUMIDITE SIRON SHT31-D	1	7	1
4	FEUR DE TEMPERATURE THERMISTANCE EPOXY - 3950 NTC	1	8	1
5	FEUR DE TEMPERATURE ETANCHE DS18B20	1	2	1
6	FEUR DE TEMPERATURE ET D'HUMIDITE HTU21D-	2	5	1
7	FEUR DE TEMPERATURE THERMISTANCE - CTN	2	7	1
8	FEUR DE TEMPERATURE TMP36	2	8	1

ici nous mettons notre cluster en tuple (cluster):

```

39
40 cluster__={}
41 i=0
42 n=4
43 while (i<n):
44     variabel = Y[Y['labels'] == i]
45     cluster__[i]=variabel.iloc[:,1:5]
46     i=i+1
47
48

```

Nos clusters:

Key	Type	Size	Value
0	DataFrame	(25, 4)	Column names: Response Time, Successability, prix, batterie
1	DataFrame	(25, 4)	Column names: Response Time, Successability, prix, batterie
2	DataFrame	(25, 4)	Column names: Response Time, Successability, prix, batterie
3	DataFrame	(25, 4)	Column names: Response Time, Successability, prix, batterie

L'affichage du contenu de chaque cluster :

```
{0:      Response Time  Successability  prix  batterie
50      179.00          72      62      6500
51      227.00          85      83      8500
52      511.25          96      85      9000
53      173.00          78      70      7000
54      645.00          68      85      9000
55      300.12          86      52      5500
56     1035.00          97      55      6000
57      135.00          97      65      6000
58      430.50          95      68      7000
59      205.33          40      65      7000
60      262.50          97      70      7000
61      451.00          97      68      7000
62      207.00          42      82      8500
63      115.00          86      65      6500
64      184.00          83      35      3500
65      119.33          100     65      6500
66      166.67          95      73      7000
67      241.50          75      83      8500
68      301.80          79      83      8500
69      300.60          100     73      7000
70      227.60          100     66      6500
71      617.00          67      68      7000
72      151.33          96      77      7500
73      115.00          99      82      8000
74       42.50          84      75      7500,
1:      Response Time  Successability  prix  batterie
0       408.21          98      60      6500
```

### 3.2 Code d'algorithme génétique

Ce code permet de calculer le max – min de capteurs :

```
68     def QosMaxMin(Namlabel,numCluster):
69         var = cluster__[numCluster]
70         return var[Namlabel].max() - var[Namlabel].min()
71
```

Ce code permet de normaliser les attributs négatifs:

```
81     def QosNigative(Namlabel,numCluster,indexGenom):
82
83         var = cluster__[numCluster]
84         max=var[Namlabel].max()
85         qosn=var.loc[indexGenom,Namlabel]
86
87         return(max-qosn)/QosMaxMin(Namlabel, numCluster)
88
89
```

Ce code permet de normaliser les attributs positifs:

```

89
90     def QosPositive(Namlabel,numCluster,indexGenom):
91
92         var = cluster__[numCluster]
93
94         min=var[Namlabel].min()
95         qosp=var.loc[indexGenom,Namlabel]
96
97         return(qosp-min)/QosMaxMin(Namlabel, numCluster)
98

```

La fonction qui permet de calculer la fitness:

```

100     def fitCapteur(numCluster,index_captuer):
101         namlabel=['Response Time','Successability','prix','batterie']
102         cof=[0.2,0.1,0.4,0.3]
103         negPos=["pos","neg","pos","neg"]
104         fit = 0
105         var=cluster__[numCluster]
106         genom=var.iloc[index_captuer,:]
107         print(genom)
108         for i in range(4):
109
110             if (negPos[i]=="pos"):
111                 fit = fit + ( QosPositive(namlabel[i],numCluster,index_captuer) * cof[i] )
112                 print (namlabel[i], " : " ,QosPositive(namlabel[i],numCluster,index_captuer)
113             else:
114                 fit = fit + ( QosNigative(namlabel[i],numCluster,index_captuer) * cof[i] )
115                 print (namlabel[i], " : " ,QosNigative(namlabel[i],numCluster,index_captuer)
116         return fit
117

```

Résultat de la méthode fitness:

```

Response Time      408.21
Successability     98.00
prix               60.00
batterie           6500.00
Name: 0, dtype: float64
Response Time : 0.10949746285993764 * 0.2
Successability : 0.07142857142857142 * 0.1
prix : 0.5454545454545454 * 0.4
batterie : 0.4166666666666667 * 0.3
fitness du captuer est : 0.37222416789666285

```

Le code pour choisir la population initiale:

```

121
122     population_initial = {}
123     for i in range(8):
124         genom= cluster__[i]
125         for j in range(4):
126             v= cluster__[j]
127         population_initial[i]=genom.sample()
128

```

Résultat de ce code :

```
In [125]: population_initial
Out[125]:
{'genom0': [8, 29, 90, 73],
 'genom1': [12, 33, 97, 76],
 'genom2': [23, 29, 85, 64],
 'genom3': [2, 38, 78, 54],
 'genom4': [23, 32, 79, 68],
 'genom5': [9, 40, 81, 54],
 'genom6': [5, 35, 90, 57],
 'genom7': [17, 21, 86, 66]}
```

La fonction crossover permet de faire croisement entre les capteurs:

```
139     def crossover(a: Genome, b: Genome, numCluster):
140         if len(a) != len(b):
141             raise ValueError("Les génomes a et b doivent être de même longueur")
142
143         longueur = len(a)
144         if longueur < 4:
145             return a, b
146
147         p = randint(1, longueur - 1)
148         return a[0:p] + b[p:], b[0:p] + a[p:]
149
```

Résultat de croisement :

```
parants : {'genom0': [8, 29, 90, 73], 'genom1': [12, 33, 97, 76]}
Crossover : [8, 29, 97, 76]
```

La fonction mutation permet de faire mutation :

```
163
164     def mutation(genome: Genome, numCluster):
165         index = randrange(len(genome))
166         genome[index] = cluster__[numCluster].sample()
167         return genome
168
```

Le code de l'algorithme génétique :

```
146 def alg_gen(population_in,besoin):
147     fitness_besoin =fitCapteur(besoin)
148     fitness = 0
149     n_gen = 0
150     i =0
151     max_fitnes = 0
152     generatin[i]=population_in
153     while (fitness < besoin & n_gen < 250 ) :
154
155         parents = generatin[0:2]
156
157
158         parents = generatin[i]
159
160         for j in range(int(len(population) / 2) - 1):
161             fitness = fitCapteur
162             if ( fitness > max_fitnes ):
163                 max_fitnes=fitness
164
165
166
167         offspring_a, offspring_b = crossover(parents[0], parents[1])
168         offspring_a = mutation(offspring_a)
169         offspring_b = mutation(offspring_b)
170         next_generation += [offspring_a, offspring_b]
171
172         population = next_generation
173
174         print("fitness de meilleure genom dans la genration ",i," est : " ,max_fitnes
175         i=i+1
176         n_gen =n_gen+1
177         return population, i ,max_fitnes
178
```

Le code d'affichage:

```
if (n_gen < 250) :
    print("La meilleure solution est " ,population)
    print("fitness de meilleure solution est : " ,max_fitnes)
    print("Meilleure solution trouvée après ",i,"génération")
```

Résultat de l'algorithme génétique :

```

fitness de meilleure genom dans la generation 0 est : 0.076725
fitness de meilleure genom dans la generation 1 est : 0.089725
fitness de meilleure genom dans la generation 2 est : 0.102725
fitness de meilleure genom dans la generation 3 est : 0.115725
fitness de meilleure genom dans la generation 4 est : 0.128725
fitness de meilleure genom dans la generation 5 est : 0.14172500000000002
fitness de meilleure genom dans la generation 6 est : 0.15472500000000003
fitness de meilleure genom dans la generation 7 est : 0.16772500000000004
fitness de meilleure genom dans la generation 8 est : 0.18072500000000005
fitness de meilleure genom dans la generation 9 est : 0.19372500000000006
fitness de meilleure genom dans la generation 10 est : 0.20672500000000008
fitness de meilleure genom dans la generation 11 est : 0.21972500000000001
fitness de meilleure genom dans la generation 12 est : 0.23272500000000001
fitness de meilleure genom dans la generation 13 est : 0.24572500000000001
fitness de meilleure genom dans la generation 14 est : 0.25872500000000001
fitness de meilleure genom dans la generation 15 est : 0.27172500000000001
fitness de meilleure genom dans la generation 16 est : 0.28472500000000001
fitness de meilleure genom dans la generation 17 est : 0.297725000000000013
fitness de meilleure genom dans la generation 18 est : 0.310725000000000014
fitness de meilleure genom dans la generation 19 est : 0.323725000000000015
fitness de meilleure genom dans la generation 20 est : 0.336725000000000016
fitness de meilleure genom dans la generation 21 est : 0.34972500000000002
fitness de meilleure genom dans la generation 22 est : 0.36272500000000002
fitness de meilleure genom dans la generation 23 est : 0.37572500000000002
fitness de meilleure genom dans la generation 24 est : 0.38872500000000002
fitness de meilleure genom dans la generation 25 est : 0.40172500000000002

```

```

fitness de meilleure genom dans la generation 22 est : 0.36272500000000002
fitness de meilleure genom dans la generation 23 est : 0.37572500000000002
fitness de meilleure genom dans la generation 24 est : 0.38872500000000002
fitness de meilleure genom dans la generation 25 est : 0.40172500000000002
fitness de meilleure genom dans la generation 26 est : 0.414725000000000023
fitness de meilleure genom dans la generation 27 est : 0.439725000000000025
fitness de meilleure genom dans la generation 28 est : 0.46472500000000003
fitness de meilleure genom dans la generation 29 est : 0.48972500000000003
fitness de meilleure solution est : 0.514725
Meilleure solution trouvée après 30 générations
La meilleure solution est [ 22 43 58 48 ]

```

captuer 22 : CAPTEUR DE TEMPERATURE NUMERIQUE ET D'HUMIDITE

```

Response Time      132.0
Successability     72.0
prix               58.0
batterie           5500.0

```

captuer 43: CAPTEUR DE LUMIERE AMBIANTE TEMT8000

```

Response Time      170.0
Successability     100.0
prix               65.0
batterie           3500.0

```

capteur 58 : DÉTECTEUR DE MOUVEMENT Goobay 95172

```

Response Time      430.5
Successability     95.0
prix               95.0
batterie           7000.0

```

CAPTEUR QUALITÉ D'AIR Dioxyde de carbone

```

Response Time      47.27
Successability     85.00
prix               78.00
batterie           4000.00

```

### 3.3 Le code de l'interface :

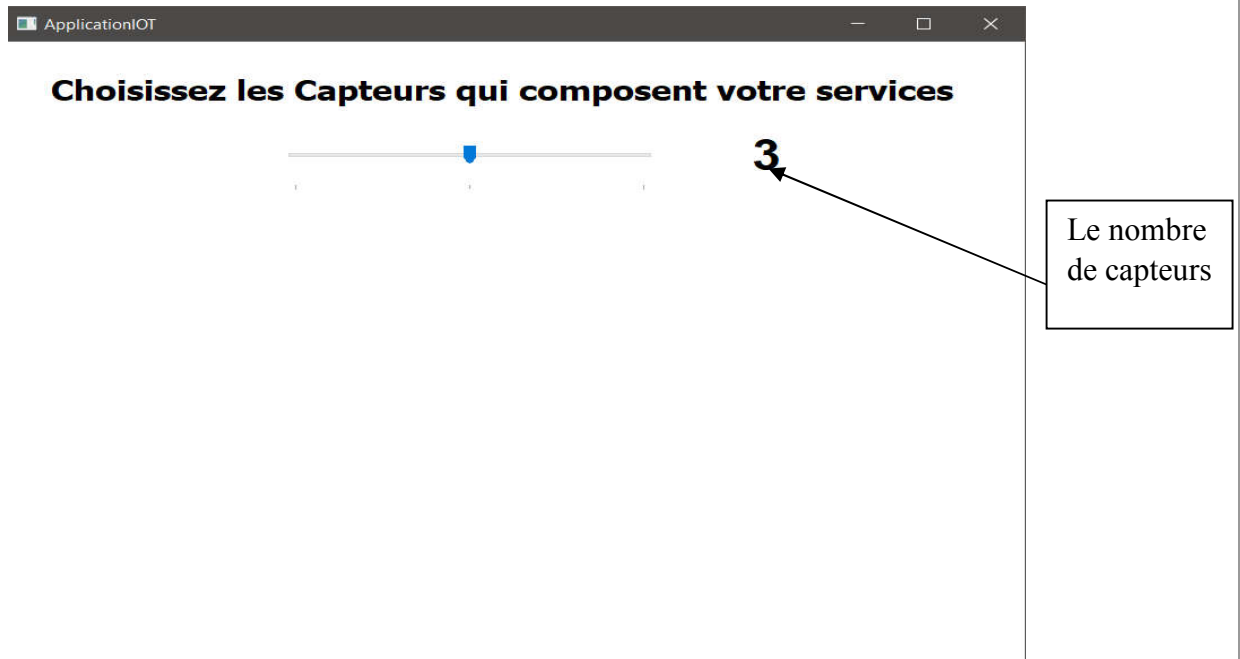
Voici quelques captures de code d'interface :

```
1  from PyQt5 import QtCore, QtGui, QtWidgets
2
3
4  class Ui_MainWindow(object):
5      def setupUi(self, MainWindow):
6          MainWindow.setObjectName("MainWindow")
7          MainWindow.resize(889, 661)
8          MainWindow.setStyleSheet("QMainWindow{\n"
9      "    background-color: rgb(255, 255, 255);\n"
10     "}")
11         self.centralwidget = QtWidgets.QWidget(MainWindow)
12         self.centralwidget.setObjectName("centralwidget")
13         self.pushButton = QtWidgets.QPushButton(self.centralwidget)
14         self.pushButton.setGeometry(QtCore.QRect(500, 560, 231, 41))
15         font = QtGui.QFont()
16         font.setFamily("MS Shell Dlg 2")
17         font.setPointSize(-1)
18         self.pushButton.setFont(font)
19         self.pushButton.setStyleSheet("QPushButton{\n"
20     "    background-color: rgb(85, 170, 255);\n"
21     "    font-size: 16px;\n"
22     "border-radius: 14px;\n"
23     "\n"
24     "\n"
25     "\n"
26     "}")
27         self.pushButton.setObjectName("pushButton")
28         self.pushButton_2 = QtWidgets.QPushButton(self.centralwidget)
29         self.pushButton_2.setGeometry(QtCore.QRect(130, 560, 231, 41))
30         font = QtGui.QFont()
31         font.setFamily("MS Shell Dlg 2")
32         font.setPointSize(-1)
33         self.pushButton_2.setFont(font)
34         self.pushButton_2.setStyleSheet("QPushButton{\n"
35     "    background-color: rgb(255, 83, 83);\n"
```

```
self.label_4.setObjectName("label_4")
self.label_5 = QtWidgets.QLabel(self.centralwidget)
self.label_5.setGeometry(QtCore.QRect(500, 120, 171, 41))
font = QtGui.QFont()
font.setPointSize(9)
font.setBold(True)
font.setWeight(75)
self.label_5.setFont(font)
self.label_5.setObjectName("label_5")
self.verticalLayoutWidget = QtWidgets.QWidget(self.centralwidget)
self.verticalLayoutWidget.setGeometry(QtCore.QRect(40, 210, 150, 131))
self.verticalLayoutWidget.setObjectName("verticalLayoutWidget")
self.verticalLayout = QtWidgets.QVBoxLayout(self.verticalLayoutWidget)
self.verticalLayout.setContentsMargins(0, 0, 0, 0)
self.verticalLayout.setObjectName("verticalLayout")
self.label_7 = QtWidgets.QLabel(self.verticalLayoutWidget)
font = QtGui.QFont()
font.setBold(True)
font.setWeight(75)
self.label_7.setFont(font)
self.label_7.setObjectName("label_7")
self.verticalLayout.addWidget(self.label_7)
self.label_6 = QtWidgets.QLabel(self.verticalLayoutWidget)
font = QtGui.QFont()
font.setBold(True)
font.setWeight(75)
self.label_6.setFont(font)
self.label_6.setObjectName("label_6")
self.verticalLayout.addWidget(self.label_6)
self.label_8 = QtWidgets.QLabel(self.verticalLayoutWidget)
font = QtGui.QFont()
font.setBold(True)
font.setWeight(75)
self.label_8.setFont(font)
self.label_8.setObjectName("label_8")
self.verticalLayout.addWidget(self.label_8)
```

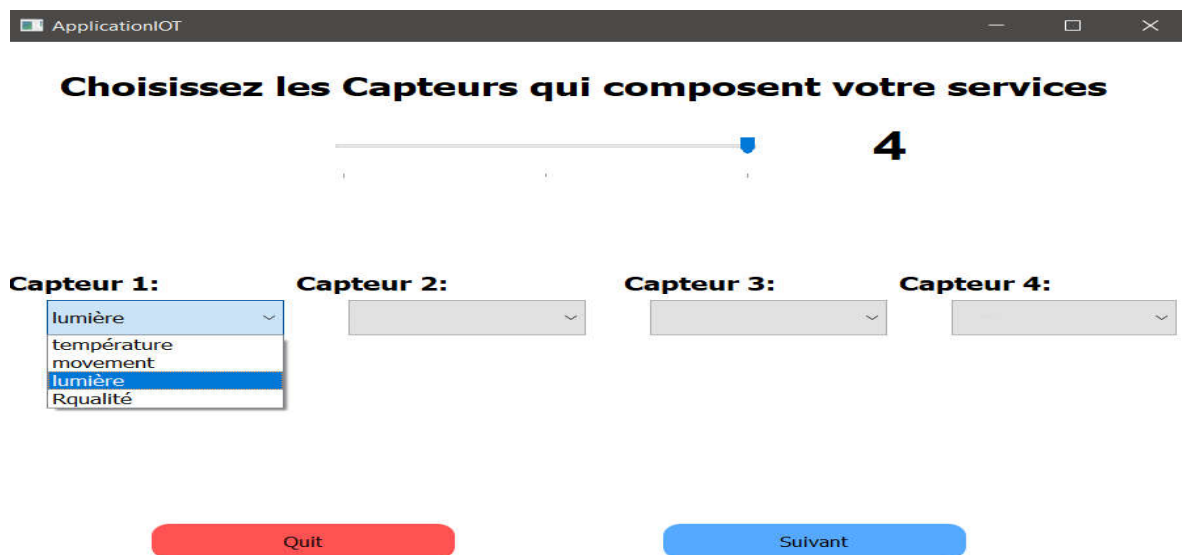
### 3.4 L'interaction de l'utilisateur avec notre Système

Dans cette étape l'utilisateur choisit le nombre de capteurs dont il a besoin



**Figure 4.6:** interface qui permet a l'utilisateur de choisir le nombre de capteurs

Puis, l'utilisateur choisit les capteurs dont il a besoin :



**Figure 4.7:** interface qui permet à l'utilisateur de choisir les capteurs

Ensuite, l'utilisateur entre les caractéristiques de service :



ApplicationIoT

## Enter caractéristiques de votre service

<b>Prix</b>	<b>Temps de réponse:</b>	<b>succès</b>	<b>Batterie</b>
<input type="text" value="300€"/>	<input type="text" value="800"/>	<input type="text" value="87%"/>	<input type="text" value="20000"/>

**Figure 4.8:** l'interface lorsque l'utilisateur met des caractéristiques de service

Meilleur résultat de service affiché à l'utilisateur :

ApplicationIoT
— □ ×

## le meilleur service

**CAPTEUR DE TEMPERATURE NUMERIQUE ET D'HUMIDITE**  
**AM2560**

Prix : 58€

Temp de réponse : 132

Succes : 72

Batterie : 5500

**DÉTECTEUR DE MOUVEMENT**  
**Goobay 95172**

Prix : 95€

Temp de réponse : 430.5

Succes : 95

Batterie : 7000

**CAPTEUR DE LUMIERE AMBIANTE**  
**TEM8000**

Prix : 65€

Temp de réponse : 170

Succes : 100

Batterie : 3500

**CAPTEUR QUALITÉ D'AIR**  
**Dioxyde de carbone**

Prix : 78€

Temp de réponse : 47.27

Succes : 85

Batterie : 4000

Les Caractéristiques	Prix	Temp de réponse	Succes	Batterie
Les Besoins d'utilisateur	300€	800	87%	20000
<b>total :</b>	<b>296€</b>	<b>779.77</b>	<b>88%</b>	<b>20000</b>

Quit

Termier

Figure 4.8: l'interface du résultat qui s'affiche à l'utilisateur

## **Conclusion**

Dans la première partie de ce dernier chapitre nous avons présenté les différents langages et outils de développement que nous avons utilisé afin d'implémenter notre solution proposé. Dans la deuxième partie nous avons donné le coté réalisation de notre projet, avec des exemples de code, et un ensemble des capteurs de l'interface.

### **Conclusion Générale:**

L'IoT est apparu en 1999 lorsque des chercheurs du Massachusetts Institute de Technologie (MIT) créaient un réseau basé sur Internet qui couvrirait toutes les choses dans le monde pour réaliser une identification automatique des choses grâce au partage d'informations. Cette technologie est donc un réseau qui connecte et rassemble des objets avec Internet, en suivant les protocoles qui assurent leur communication et l'échange d'informations à travers plusieurs appareils. Il est principalement basé sur les objets connectés. Un objet connecté a la capacité de capturer des données et de les envoyer, via Internet ou d'autres technologies, afin qu'elles puissent être analysées et affichées sur des tableaux de bord dédiés. Les objets connectés interagissent avec leur environnement grâce à des capteurs. Il est constitué d'un ensemble hétérogène de réseaux qui permettent la communication de ces objets.

Notre projet a été réalisé dans le cadre de la technologie Internet des Objets, Pour répondre au besoin de l'utilisateur qui cherche à obtenir le bon service au bon moment et au bon endroit, et au meilleur coût.

L'apprentissage automatique est largement utilisé dans différents domaines, dans notre projet, nous avons mis en œuvre une méthode pour classer les services IoT, basée sur une combinaison de deux algorithmes, le premier est un algorithme d'apprentissage automatique : K-means est un algorithme de minimisation alternatif qui, étant donné un entier K, cherchera à séparer un ensemble de points dans K clusters en fonction de leur similarité.

Et le second est l'algorithme génétique, l'un des algorithmes d'optimisation.

Pour atteindre cet objectif, nous avons un ensemble de données que nous créons, cet ensemble de données contient la qualité des services, leur description, leurs entrées et leurs sorties, nous appliquons l'algorithme k-means sur notre ensemble de données pour trouver les services qui ont la même fonctionnalité que nous appliquons l'algorithme génétique sur le résultat de l'algorithme k-means pour trouver les services optimaux.

Dans la seconde partie nous réalisons une interface qui permet à l'utilisateur de trouver le meilleur service en fonction de ses besoins.

# **Bibliographie**

[1] <http://www.futura-sciences.com/tech/definitions/internet-internet-objets-15158/>,

(consulté le 19 Avril 2021)

[2] Rahim Tafazolli, "TECHNOLOGIES FOR THE WIRELESS FUTURE", John Wiley & Sons, 2006

[3] Instituts carnot, Le livre blanc : Objets communicants et Internet des objets, Réseau des instituts carnot 21-06-2011

[4] O. Taleb et A. Mankouri. Programmation de la sécurité Internet des Objets, Etude de cas module WIFI Electric imp. Mémoire de L'université de Telemcen, 2016, 82p

[5] <http://blog.octo.com/modeles-architectures-internet-des-objets/>, (consulté le 15 Avril 2021).

[6] CHALLAL, Yacine. Sécurité de l'Internet des Objets : vers une approche cognitive et systémique. Thèse de doctorat : Technologies de l'Information et des Systèmes. France : Juin 2012, 78 p.

[7] Intel. Applications pour l'internet des objets dans l'industrie. [en ligne]. Disponible sur : <https://www.intel.fr/content/www/fr/fr/internet-of-things/industry-solutions.html> (Consulté le 27 Avril 2021)

[8] Hadjadj, Walid. étude de cas sur un système médical domotique contrôlé par un SMA, Mémoire de l'université Larbi Ben Mhidi Oum El Bouaghi. 13/06/2018, 64p

[9] [https://www.itu.int/en/ITU-D/Regional-Presence/Africa/Documents/PRIDA%202020%20-%20ONLINE%20Capacity%20building%20%26%20digital%20services/FR\\_Workshop\\_Slides.pdf](https://www.itu.int/en/ITU-D/Regional-Presence/Africa/Documents/PRIDA%202020%20-%20ONLINE%20Capacity%20building%20%26%20digital%20services/FR_Workshop_Slides.pdf) (consulté le 19/04/2021)

[10] <https://fr.statista.com/statistiques/584481/internet-des-objets-nombre-d-appareils-connectes-dans-le-monde--2020/> (consulté le 27/05/2021)

[11] Martín-Palma, R. J., & Lakhtakia, A. (2013). Engineered Biomimicry For Harvesting Solar Energy: A Bird's Eye View. International Journal of Smart and Nano Materials, 4(2), 83-90

- [12] Baluja, S. (1994). Population-Based Incremental Learning. A Method For Integrating Genetic Search Based Function Optimization And Competitive Learning (No. CMU-CS-94-163). Carnegie-Mellon Univ Pittsburgh Pa Dept Of Computer Science.
- [13] Fogel, D. B. (1995), Evolutionary Computation: Toward a New Philosophy of Machine Intelligence (Piscataway, NJ: IEEE).
- [14] Bäck, T. (1996), Evolutionary Algorithms In Theory And Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms. Oxford university press.
- [15] Bäck, T., Fogel, D. B., & Michalewicz, Z. (1997), Handbook Of Evolutionary Computation. Release, 97(1), B1
- [16] Mühlenbein, H., & Mahnig, T. (2001), Evolutionary Algorithms: From Recombination to Search Distributions. In Theoretical Aspects of Evolutionary Computing (pp. 135-173). Springer Berlin Heidelberg.
- [17] Le Riche, R., Schoenauer, M., & Sebag, M. (2007). Un état des lieux de l'optimisation évolutionnaire et de ses implications en sciences pour l'ingénieur.
- [18] Banzhaf, W. (2013), Engineered Biomimicry: Chapter 17. Evolutionary Computation and Genetic Programming. Elsevier Inc. Chapters.
- [19] Aziza, BECHIR, RÉOLUTION DE PROBLÈMES D'OPTIMISATION PAR LES SYSTÈMES MULTI-AGENTS ET LES APPROCHES ÉVOLUTIONNAIRES. Mémoire
- [20] [en ligne] Disponible sur: <<https://elearn.univ-ouargla.dz/2013->
- [21] [en ligne] Disponible sur: <[https://elearn.univ-ouargla.dz/2013-2014/courses/INTRODUCTIONAUXMETHO/document/CH\\_3.pdf?cidReq=INTRODUCTION\\_AUXMETHO](https://elearn.univ-ouargla.dz/2013-2014/courses/INTRODUCTIONAUXMETHO/document/CH_3.pdf?cidReq=INTRODUCTION_AUXMETHO)> (consulté le 21/05/2021)
- [22] Dominique REVUZ, Algorithme génétique, [en ligne] ,30 Janvier 2014 Disponible sur: <<http://www-igm.univ-mlv.fr/~dr/>>consulté le (28/05/2021)
- [23] génétique algorithme, [en ligne] Disponible sur :<https://schwarztiger.wordpress.com> 2008/08/30 consulté le(28/05/2021)
- [24] J. Landré, " Analyse multi résolution pour la recherche et l'indexation d'images par le contenu dans les bases de données images -Application à la base d'images paléontologique trans'nyfipal ", Décembre 2005

- [25] Jérôme Azé, These, K-plus proches voisins- - Université Paris 11 / LRI-CNRS- 1er mars 2007
- [26] Laouamer, L. (2006). Approche exploratoire sur la classification appliquée aux images.
- [27] Govaert, G. (1990). Classification binaire et modèles. Revue de statistique appliquée, 38(1), 67-81.
- [28] J. B. MacQueen, "Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability", Berkeley, University of California Press, 1971.
- [29] E. Forgy, Cluster analysis of multivariate data: Efficiency vs. interpretability of classifications, Biometrics, pp.768, 1965.
- [30] L. Khodja, Contribution à la Classification Floue non Supervisée, thèse de Doctorat,
- [31] [Python - Overview. s.d, sur Tutorials Point SimplyEasyLearning: [https://www.tutorialspoint.com/python/python\\_overview.htm](https://www.tutorialspoint.com/python/python_overview.htm) Consulté le 07/06/2021
- [32] Ilemona S.Atawodi.(2019).A Machine Learning Approach to Network Intrusion Detection System Using K Nearest Neighbor and Random Forest.Thèse
- [33] [Open Source Community. Anaconda. Avril 2019]
- [34] [[https://kite.com/python/docs/nltk.word\\_tokenize](https://kite.com/python/docs/nltk.word_tokenize) Consulté le (29 /06/ 2021)
- [35] [<http://pandas.pydata.org/pandas-docs/stable/overview.html#license> Consulté le (29 /06/ 2021)