

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Université de Béjaïa
Faculté des Sciences Exactes
Département d'Informatique

MEMOIRE DE MASTER RECHERCHE

En
Informatique

Option
Systemes d'Information Avancés

Thème

Proposition d'une approche de composition de
services IoT avec prise en compte des QoS

Présenté par : M. *SEKKAL nassim*

Soutenu le 13 septembre 2023 devant le jury composé de

Présidente	<i>M^{me}</i> BOUADEM Nassima	MCB	Université de Béjaïa
Examinatrice	<i>M^{me}</i> CHABANE Sarah	MCB	Université de Béjaïa
Encadrante	<i>M^{me}</i> AIT HACENE souhila	MAA	Université de Béjaïa
Co-Encadrante	<i>M^{me}</i> CHIBANI SADOUKI Samia	MCA	Université de Béjaïa

Béjaïa, Septembre 2023.

Au Nom de Dieu Le Plus Puissant ...

Remerciements

Je tiens à exprimer ma profonde gratitude envers mes parents et ma famille, qui ont toujours été présents et fidèles, me soutenant tout au long de mes études. Leur amour, leur encouragement et leur soutien ont été un moteur essentiel dans ma réussite académique.

Je souhaite également adresser mes sincères remerciements à mon encadrante, *M^{me} AIT HACENE Souhila*, ainsi qu'à ma Co-Encadrante, *M^{me} CHIBANI SADOUKI Samia*, pour le temps qu'elles m'ont consacré, leurs précieux conseils et leur assistance inestimable tout au long de la réalisation de ce projet. Leurs orientations éclairées ont été cruciales pour la réussite de ce travail.

Enfin, je tiens aussi à remercier tous les membres du jury pour avoir accepté d'évaluer ce travail.

Dédicaces

A mes chers parents

A mes chères sœurs

A mes chères tantes

A ma chère grand-mère

A mes chers cousins

A mes chers oncles

A mes chères cousines

A tous les membres de ma famille

M. SEKKAL Nassim

Table des matières

Table des figures	iii
Liste des tableaux	iv
Liste des abréviations	v
Introduction générale	1
1 Généralités	2
1.1 Introduction	2
1.2 L'Internet des objets (IoT)	2
1.2.1 Définition	2
1.2.2 Les composants d'un systèmes IoT	3
1.3 Architecture d'un système IoT	5
1.3.1 Couche perception	5
1.3.2 Couche réseau	5
1.3.3 Couche application	5
1.4 Phases exécutives d'un système IoT	6
1.4.1 Collecte de données	6
1.4.2 Stockage des données	6
1.4.3 Traitement des données	7
1.4.4 Transmission de données	7
1.4.5 Livraison des données	7
1.5 Les principales caractéristiques de l'IoT	7
1.6 Les Défis de l'IoT	8
1.7 Conclusion	9
2 État de l'art sur la Composition de Services	10
2.1 Introduction	10
2.2 Service IoT	10
2.3 Composition de services	10
2.3.1 Définition	10
2.3.2 Étapes du processus de composition	11
2.3.3 Définitions des concepts liés à la problématique de composition de services	12
2.3.4 Composition concrète (comp)	13
2.3.5 Les critères de QoS	14
2.3.6 Vecteur QoS d'un service candidat	14

2.3.7	Vecteur d'attributs QoS d'une composition	14
2.3.8	Fonction d'utilité	15
2.4	Machine learning	15
2.4.1	Définition	15
2.4.2	Types de systèmes d'apprentissage automatique	15
2.5	Métaheuristique	18
2.5.1	Définition	19
2.6	Etat de l'art sur les approches de composition de services IoT basées sur les QoS	19
2.6.1	Approches basées sur des techniques déterministes	19
2.6.2	Approches basées sur des techniques non déterministes	20
2.6.3	Comparaison entre les approches métaheuristiques et les approches dé- terministes	22
2.7	Conclusion	25
3	Approche proposée	27
3.1	Introduction	27
3.2	Motivation	27
3.3	Aperçu sur les arbres de décision	28
3.3.1	Définition	28
3.3.2	Construction d'un arbre de décision	29
3.4	WOA original	29
3.4.1	Définition	29
3.5	Le fonctionnement de l'approche proposée DT-WOA	30
3.5.1	Classification et sélection des services candidats via l'arbre de décision	30
3.5.2	Application du WOA discrétisé	32
3.6	Scénario d'application de DT-WOA (Etude de cas)	35
3.7	Simulation	37
3.7.1	Environnement, outils et bibliothèques utilisés	37
3.7.2	Dataset	38
3.7.3	Pré-traitement des données	39
3.7.4	Normalisation des données du dataset	39
3.7.5	Classification via l'arbre de décision	40
3.7.6	Recherche d'une composition sous optimale via WOA discrétisé	44
3.8	Conclusion	44
	Conclusion générale et perspectives	45
	Bibliographie	46

Table des figures

1.1	Paradigme de l'Internet des Objets	3
1.2	Fonctionnement d'un système IoT	4
1.3	Architecture IoT à trois couches	6
2.1	Exemple de composition de services	11
2.2	Aperçu de la composition sensible à la QoS	12
2.3	Représentation d'un service abstrait	13
2.4	Apprentissage non supervisé	16
2.5	Apprentissage supervisé	17
2.6	Processus classique de reconnaissance par apprentissage	18
3.1	Schéma d'un arbre de décision	28
3.2	La chasse en filet de bulles des baleines à bosses	30
3.3	Plan de composition associé au scénario de DT-WOA	36
3.4	Description des critères de QoS du dataset QWS	39
3.5	QWS 2.0 avant la normalisation	40
3.6	QWS 2.0 après la normalisation	40
3.7	Les valeurs de l'attribut temps de réponse avant la classification	41
3.8	Les valeurs de l'attribut Succès avant la classification	41
3.9	QWS 2.0 après la classification	42
3.10	les valeurs de l'attribut Temps de réponse après la classification	42
3.11	les valeurs de l'attribut Succès après la classification	43
3.12	QWS 2.0 après la selection	43
3.13	Évolution des valeurs de fitness de DT-WOA et WOA originale	44

Liste des tableaux

2.1	Fonction d'agrégation des propriétés QoS	15
2.2	Tableau comparatifs des approches étudiées	25
3.1	Services candidats disponibles pour chaque service abstrait dans le scénario de composition	37
3.2	Valeurs de fitness de DT-WOA et de WOA	44

Liste des abréviations

IoT	Internet of Things
QoS	Quality of Service
WOA	Whale Optimization algorithm
SC	Service candidat
AS	Service abstrait
SCA	services composites abstraits
comp	composition
Rs	Temps de réponse
Di	Disponibilité
D	Débit
S	Succès
F	Fiabilité
C	Conformité
Mp	Les meilleures pratiques
L	Latence
Dc	Documentation
HMM	Hidden Markov Model
ACO	Ant Colony Optimization
HMAK	Hybrid Markov-Ant Colony
GWO	Gray Wolf Optimizer
SSO	Social Spider Optimization
RI – SSO	Reputation Improved- Social Spider Optimization Algorithm
BSPO	bi-objective shortest path optimization
SMSC	solutions to the manufacturing service composition
FPA	flower pollination
ABC	artificial bee colony
PSO	Practical Swarm Optimization
MFO	Month Flame Optimization
nSC	nombre de services candidats
nAS	nombre de services abstraits
nCOM	nombre de compositions
niteration	nombre des iterations

Introduction générale

L'Internet des objets (IoT) représente une avancée majeure dans la technologie, en reliant des objets physiques au monde numérique. Grâce à cette interconnexion, ces objets collectent et partagent des données, ouvrant ainsi de nouvelles perspectives dans des domaines variés, allant des maisons intelligentes aux villes intelligentes, en passant par la santé, la sécurité, la logistique et la fabrication.

Au cœur de cette évolution se trouve le concept de service IoT. Un service IoT peut être défini comme une fonctionnalité spécifique offerte par un objet connecté, capable de collecter, traiter et transmettre des données pour répondre à des besoins précis. Dans ce contexte, la composition de services IoT devient l'art de combiner intelligemment ces fonctionnalités pour créer des solutions plus larges et plus sophistiquées.

La composition de services IoT revêt une importance capitale en raison du caractère multifacette des besoins des utilisateurs. Un seul service IoT ne suffit souvent pas à répondre de manière complète à ces besoins diversifiés. Ainsi, l'approche de la composition de services IoT s'avère nécessaire pour créer des applications complexes et personnalisées qui tirent pleinement parti des capacités de l'IoT.

Notre contribution principale consiste à proposer une approche de composition de services IoT avec prise en compte des QoS. Cette approche, qui combine l'arbre de décision avec une version discrétisée de l'algorithme WOA, est nommée DT-WOA. Son objectif est de rechercher une solution quasi optimale en choisissant la meilleure composition de services IoT afin de répondre aux attentes des utilisateurs.

Le rapport est structuré en trois chapitres. Le premier chapitre présente la technologie de l'IoT et ses principales fonctionnalités. Dans le deuxième chapitre, les définitions essentielles sont clarifiées pour faciliter la compréhension de l'approche proposée, en détaillant les étapes du processus de composition. De plus, les concepts clés du Machine Learning et des méta-heuristiques sont explorés, accompagnés d'une présentation de l'état de l'art et d'un tableau comparatif des approches de composition de services. Le troisième et dernier chapitre sera consacré à la présentation de notre approche, nommée Decision Tree with Whale Optimization Algorithm (DT-WOA), ainsi qu'à son évaluation, en le comparant au WOA sans réduction de l'espace de recherche.

1.1 Introduction

Ce premier chapitre est consacré à la présentation de la technologie de l'Internet des objets (IoT) et de ses principales fonctionnalités. Nous débuterons par une définition de l'IoT ainsi que ces principaux composants. Ensuite, nous allons aborder l'architecture et les phases exécutives d'un système IoT. Enfin, nous terminons ce chapitre par la présentation de certains défis de l'IoT.

1.2 L'Internet des objets (IoT)

1.2.1 Définition

L'Internet des objets ou l'IoT est l'acronyme de « Internet of Things » en anglais. L'IoT est une technologie qui permet de connecter un ensemble d'objets intelligents via internet et de les faire collaborer, coopérer et/ou communiquer dans le but de fournir des services et satisfaire les besoins des utilisateurs[1].

Une autre définition proposée par l'UIT [2] (Union Internationale des Télécommunications) est la suivante : *L'Internet des objets est une infrastructure mondiale pour la société de l'information, qui permet de disposer de services évolués en interconnectant des objets (physique ou virtuels) grâce aux technologies de l'information et de la communication interopérables existantes ou en évolution.*

La technologie IoT est utilisée dans divers domaines à savoir, la santé, l'industrie automobile, l'agriculture, les télécommunications, la vente au détail et la gestion de la chaîne d'approvisionnement, ainsi que les maisons intelligentes et les villes intelligentes (Smart Cities)[3]. La figure 1.1 présente les domaines d'application de la technologie IoT

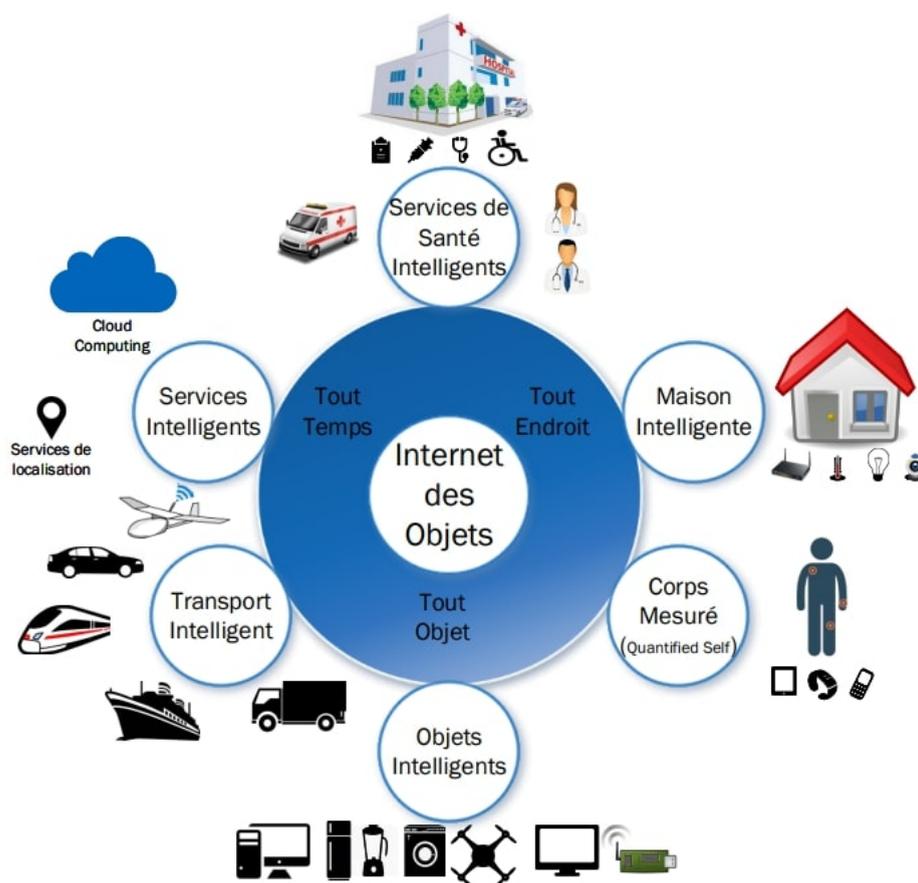


FIGURE 1.1 – Paradigme de l’Internet des Objets [4]

1.2.2 Les composants d’un systèmes IoT

L’IoT est une convergence de diverses technologies et disciplines qui travaillent en harmonie pour créer un réseau global interconnecté.

Dans cette section, nous présentons un aperçu des principales composantes qui constituent l’IoT [5] illustrées dans la figure 1.2 :

Les objets

Un objet est une entité physique, comme un livre, une montre, une voiture ou un téléphone, tandis qu’un objet connecté est un appareil doté de composants électroniques qui lui permettent de communiquer des informations avec un autre objet via une liaison sans fil. Bluetooth ou Wi-Fi, etc[5].

De manière générale, L’IoT met en œuvre deux types d’objets connectés pour interagir avec le monde physique : des capteurs et des actionneurs.

- Les capteurs permettent de recueillir des informations depuis le monde physique et de les transmettre vers le système informatique. Les types de capteurs les plus courants incluent les capteurs de température, de pression, de mouvement, de lumière et de gaz.

- Les actionneurs permettent d’agir dans le monde physique, c’est-à-dire, changer son état. Les actionneurs se manifestent par des actions concrètes, tels que l’allumage à distance d’un appareil, le déclenchement d’un avertisseur sonore ou encore la commande de robots.

Le réseau

Les réseaux fournissent le canal de communication entre les objets connectés. Ils varient en fonction de la portée, de la vitesse et de la consommation d’énergie, offrant des options adaptées aux besoins spécifiques.

Les données

Les données sont collectées par les objets connectés, puis transmises à une plateforme de traitement par le biais de divers protocoles de communication. Cette plateforme de traitement est chargée de convertir ces données brutes en informations utiles et exploitables.

Les informations

Une information est le résultat obtenu suite aux traitements et l’analyse des données brutes interceptées par les objets, il est préférable aussi de les enregistrer dans des bases de données.

Applications et Services

Les applications et services dans le contexte de l’IoT englobent les solutions spécifiques offertes dans divers domaines tels que les maisons intelligentes, les villes intelligentes, la santé connectée et la gestion de l’énergie. Ces applications représentent le cœur même du potentiel offert par l’IoT, fournissant des fonctionnalités et des avantages concrets à travers l’interconnexion d’objets et la mise en œuvre de systèmes intelligents. En effet, elles constituent la concrétisation des possibilités offertes par les objets connectés et forment la base du développement dans ce domaine en évolution constante.

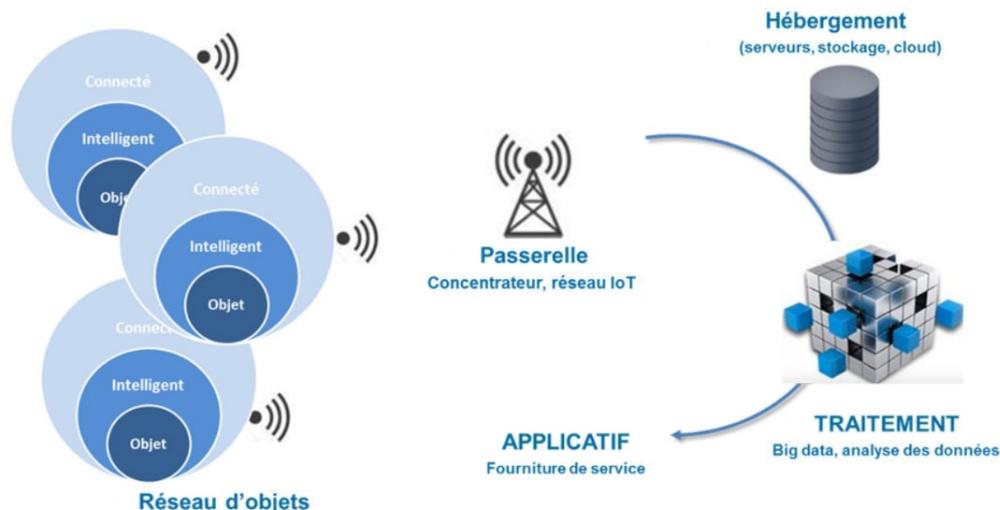


FIGURE 1.2 – Fonctionnement d’un système IoT
[6]

1.3 Architecture d'un système IoT

L'architecture de l'IoT n'est pas standardisée, néanmoins il existe une configuration courante qui comprend généralement trois couches distinctes : perception, réseau et application [3], comme le montre la figure 1.3. Dans la suite de cette section, nous explorerons en détail le rôle et les fonctionnalités de chaque couche au sein de cette architecture.

1.3.1 Couche perception

La couche perception est la couche physique dotée de capteurs et d'actionneurs destinés à détecter et collecter des informations de l'environnement. Cette couche peut être utilisée pour mesurer divers paramètres physiques, tels que la température, mais également pour identifier d'autres objets intelligents présents dans son environnement.

1.3.2 Couche réseau

La couche réseau est responsable de la connexion des objets intelligents, périphériques réseau et serveurs. Ses fonctionnalités sont également utilisées pour la transmission et le traitement des données collectées par les capteurs.

1.3.3 Couche application

La couche application est chargée de fournir à l'utilisateur des services spécifiques et des applications intelligentes, en utilisant les données collectées par les objets connectés. Cela ouvre la voie à la création de solutions novatrices et personnalisées pour répondre aux besoins variés des utilisateurs.

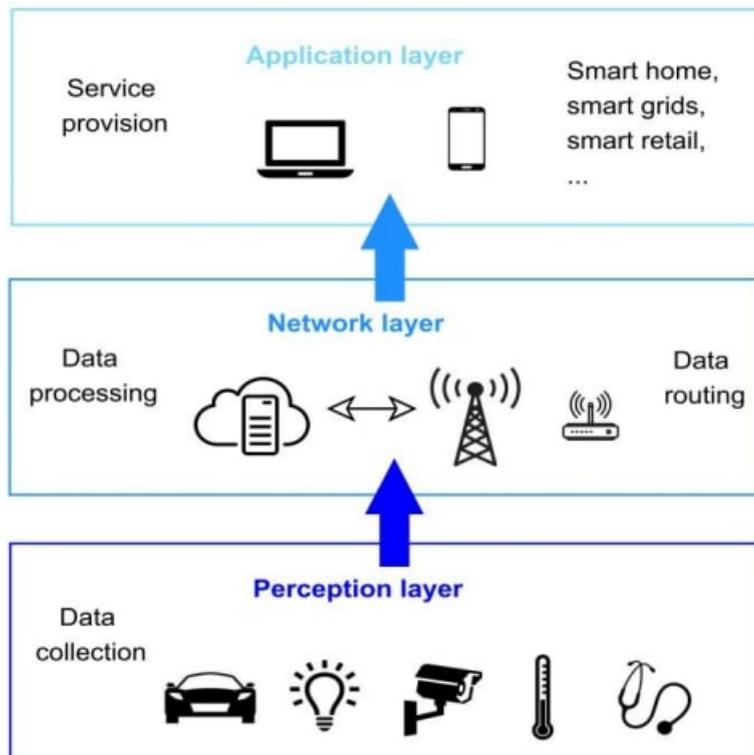


FIGURE 1.3 – Architecture IoT à trois couches
[7]

1.4 Phases exécutives d'un système IoT

Un système IoT comporte cinq phases basées sur les demandes en ligne et hors ligne. Ces phases comprennent la collecte de données jusqu'à la livraison de données [8].

1.4.1 Collecte de données

La première phase exécutive d'un système IoT consiste à collecter ou recevoir des données à partir d'objets intelligents. Différents types de collecteurs de données, choisis en fonction des caractéristiques des objets, peuvent être utilisés. Les objets peuvent être soit fixes, tels que des capteurs corporels ou des étiquettes d'identification par radiofréquence (RFID), soit dynamiques et mobiles, comme des bracelets de suivi d'activité, des balises de géolocalisation et des puces.

1.4.2 Stockage des données

Quand un objet dispose d'une mémoire interne suffisante, les données collectées pourront être stockées au niveau de cette objet. Cependant, les composants IoT sont généralement équipés de mémoires et de puissances de traitement limitées. Dans ces cas, les solutions de stockage dans le cloud prendront en charge la sauvegarde des données.

1.4.3 Traitement des données

Dans l'étape de traitement des données, les informations collectées sont analysées, filtrées et transformées en connaissances exploitables. Cela implique l'application d'algorithmes et de méthodes d'analyse de données pour identifier des tendances, des schémas et des informations pertinentes. Les données brutes sont nettoyées et préparées pour obtenir des résultats significatifs. Le traitement peut également inclure la corrélation entre différentes sources de données et la génération de rapports ou de visualisations pour une meilleure compréhension. En résumé, cette étape vise à donner du sens aux données et à les rendre utilisables pour la prise de décisions et l'optimisation des processus.

1.4.4 Transmission de données

Dans l'étape de transmission de données, les informations traitées et préparées sont transmises d'un objet connecté à un autre ou vers une plateforme centrale. Cela implique l'utilisation de divers protocoles de communication, tels que Wi-Fi, Bluetooth, Zigbee, LoRa, etc., en fonction des exigences spécifiques de l'application. Les données sont encapsulées, chiffrées si nécessaire, puis envoyées à travers le réseau. Cette étape joue un rôle crucial dans l'acheminement sécurisé et fiable des données vers leur destination, que ce soit un autre objet connecté ou un serveur distant.

1.4.5 Livraison des données

L'étape de livraison des données implique le transfert des informations traitées vers leur destination finale. Cela peut inclure l'acheminement des données vers une base de données, un serveur distant, une application ou un tableau de bord de visualisation. Les données sont organisées et stockées pour une utilisation ultérieure. Cette étape garantit que les informations collectées et traitées sont accessibles et prêtes à être exploitées par les utilisateurs finaux ou les systèmes d'analyse.

1.5 Les principales caractéristiques de l'IoT

L'IoT présente un ensemble distinct de caractéristiques qui définissent son potentiel et son impact sur notre monde connecté. Ces caractéristiques fondamentales jouent un rôle essentiel dans la façon dont l'IoT évolue.[9]. Nous présentons dans ce qui suit ces traits caractéristiques :

- inter-connectivité : Cette caractéristique permet la communication et l'échange de données entre divers objets et dispositifs, formant un réseau interconnecté de systèmes intelligents.
- Les services liés aux objets : sont des services qui permettent aux objets physiques de communiquer entre eux et d'interagir avec le monde numérique.
- Hétérogénéité : L'hétérogénéité dans l'IoT fait référence à la diversité des objets connectés, des réseaux et des protocoles utilisés dans l'IoT.
- Changement dynamique : Le changement dynamique dans IoT fait référence à la capacité des dispositifs ou des systèmes IoT à ajuster leur fonctionnement, leur configuration ou leur comportement en temps réel en réponse aux conditions changeantes de leur environnement ou aux besoins variables. Cela permet aux dispositifs IoT de s'adapter de manière autonome aux situations en évolution, améliorant ainsi leur efficacité et leur pertinence.

- passage à l'échelle : passage à l'échelle dans IoT fait référence à la capacité d'un système IoT à gérer et à fonctionner de manière efficace à mesure que le nombre d'objets connectés ou de dispositifs augmente, tout en maintenant des performances optimales et en évitant les problèmes de congestion ou de saturation du réseau. Cela implique généralement une conception robuste et une gestion efficace des ressources pour garantir la scalabilité du système IoT.
- Sécurité : La sécurité dans l'IoT implique la protection des données et des appareils contre les menaces, en utilisant des mesures telles que le chiffrement et l'authentification pour prévenir les accès non autorisés et les cyberattaques.

En résumé, les caractéristiques clés de l'IoT établissent un environnement où les objets interagissent dynamiquement, partagent des données et exigent une sécurité renforcée.

1.6 Les Défis de l'IoT

L'essor rapide de l'internet des objets a apporté des opportunités révolutionnaires, mais il est accompagné de défis complexes qui nécessitent des solutions novatrices. Dans cette section, nous présentons certaines préoccupations majeures qui entourent l'internet des objets, explorant ainsi certains de ces obstacles et défis inhérents à cette technologie en constante évolution.[10].

- **Interconnectivité Complexes (Gérer la Multitude d'Appareils Connectés)** L'un des défis majeurs de l'IoT réside dans la gestion de l'interconnectivité complexe entre une variété d'appareils et d'objets connectés. Ce défi émerge de la nécessité de faire fonctionner harmonieusement des appareils divers, utilisant différents protocoles de communication et technologies. La coordination de ces interactions peut entraîner des difficultés en termes de compatibilité, de maintenance et de sécurité. L'interconnectivité complexe exige des solutions novatrices pour garantir un fonctionnement fluide tout en prévenant les problèmes potentiels associés à la diversité des systèmes et des protocoles.
- **Préservation de la Confidentialité et Sécurité** L'un des défis majeur au sein d'un système IoT est la préservation de la confidentialité des utilisateurs et la garantie de leur sécurité. Ces deux aspects sont d'une importance cruciale pour assurer la confiance et l'acceptation généralisée de la technologie IoT. La confidentialité s'articule autour de la protection des données personnelles, tandis que la sécurité vise à contrer les menaces potentielles telles que les cyberattaques et les intrusions malveillantes. Il est primordial de mettre en œuvre des mécanismes robustes pour relever ces défis et garantir un environnement IoT sûr et respectueux de la vie privée.
- **Standardisation** La standardisation dans l'IoT est un défi important. Elle cherche à définir des règles communes pour les dispositifs et protocoles, mais la variété des technologies rend cela complexe. Trouver l'équilibre entre compatibilité et innovation est essentiel et nécessite une collaboration entre les parties prenantes pour le succès à long terme de l'IoT.
- **Développement technologique** Le développement technologique est un élément clé mais aussi un défi dans le contexte de l'IoT. Alors que de nouvelles innovations émergent

rapidement, l'interopérabilité entre les diverses solutions reste un obstacle majeur. Il faut gérer la complexité tout en garantissant une évolution fluide et des mises à jour compatibles, afin de soutenir la croissance et la durabilité de l'écosystème de l'IoT.

1.7 Conclusion

Ce chapitre a fait l'objet d'une étude sur la technologie IoT. Nous avons présenté les différents composants ainsi qu'une architecture d'un système IoT, ces caractéristiques, ces phases exécutives et certains défis liés à cette technologie. Dans le chapitre suivant, nous aborderons le concept de la composition des services IoT ainsi qu'un état de l'art sur les approches de composition des services sensibles à la QoS.

Chapitre II

État de l'art sur la Composition de Services

2.1 Introduction

Dans ce chapitre, nous présentons les étapes du processus de composition. Ensuite, nous étudions un des aspects du machine learning, à savoir les arbres de décision. Comme notre approche propose une solution pour le problème de composition de services IoT combinant un arbre de décision et une métaheuristique, nous dédions une partie de ce chapitre aux métaheuristicues. Nous clôturons ce chapitre par un état de l'art des travaux dédiés à la problématique de composition de services.

2.2 Service IoT

Un service IoT peut être défini comme une entité informatique ou logicielle conçue pour offrir des fonctionnalités spécifiques liées à l'IoT et à ses utilisateurs. Il est caractérisé par quatre éléments clés : les entrées (input), les sorties (output), les fonctionnalités (propriétés fonctionnelles) et les QoS (propriétés non-fonctionnelles). Les entrées représentent les données ou les informations nécessaires au fonctionnement du service IoT, souvent collectées à partir de capteurs ou de dispositifs connectés. Les sorties sont les résultats produits en réponse à ces entrées, généralement sous forme d'informations ou d'actions transmises à d'autres dispositifs ou systèmes. Les fonctionnalités décrivent les capacités spécifiques du service IoT, telles que la surveillance, le contrôle, l'analyse en temps réel, etc., qui exploitent les données générées par les objets connectés. Les QoS englobent des aspects essentiels de la qualité du service IoT, tels que la fiabilité, la disponibilité, les performances et d'autres caractéristiques qui définissent son aptitude à répondre aux besoins des utilisateurs dans le contexte de l'IoT[11].

2.3 Composition de services

2.3.1 Définition

Nous pouvons définir la composition de services comme le processus qui consiste à fournir de nouvelles fonctionnalités en combinant les fonctionnalités fournies par d'autres services existants (atomiques ou composites) pour satisfaire des besoins plus complexes qui ne peuvent être satisfaits par un seul service [12]. Cette combinaison doit tenir compte de la qualité de service (QoS) et de l'efficacité énergétique des objets composés [13].

La Figure 2.1 illustre un exemple de composition de services.

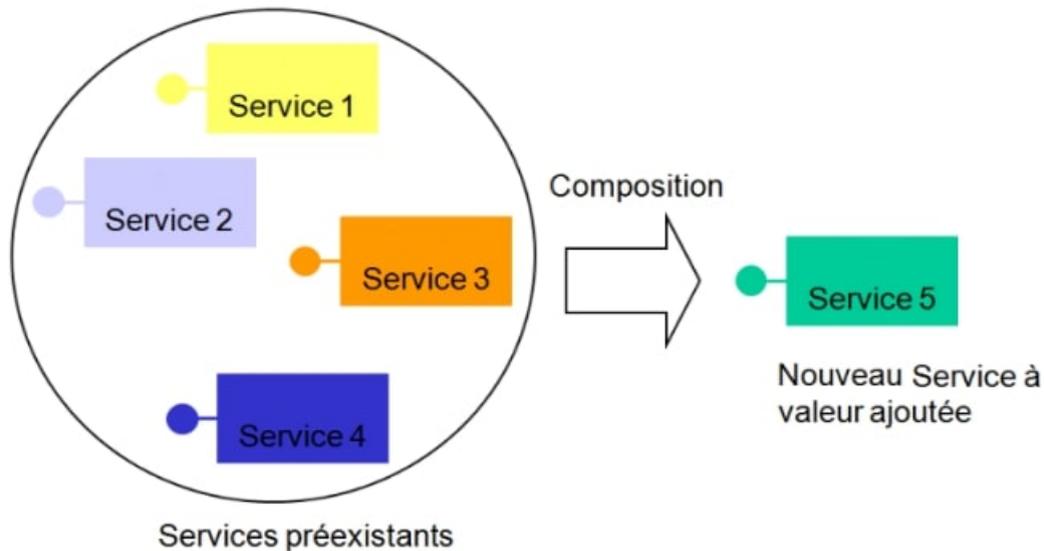


FIGURE 2.1 – Exemple de composition de services
[14]

2.3.2 Étapes du processus de composition

Le processus de composition avec prise en compte des critères de QoS se déroule en quatre étapes principales (figure 2.2) : l'élaboration du plan de composition, la découverte de services, la sélection de services basée sur les critères de QoS, ainsi que l'exécution de la composition [12][15].

Elaboration du plan de composition

À cette étape, l'accent est mis sur l'identification des exigences fonctionnelles attendues du service résultant. Ces exigences sont ensuite décomposées pour créer une description abstraite (workflow) de la composition. Cette description abstraite comprend un ensemble de tâches, chacune représentant une fonction différente. En d'autres termes, il s'agit de traduire la demande de l'utilisateur en une série de tâches fonctionnelles, qui seront ensuite intégrées dans la composition.

Découverte de services

La découverte de services consiste à rechercher, pour chaque tâche du workflow, un ensemble de services fonctionnellement équivalents. Cette ensemble de services est dit service abstrait tandis que les services qui le composent sont notés services candidats. La recherche effectuée à cette étape se base généralement sur les descriptions syntaxiques ou sémantiques, en mettant l'accent sur la correspondance fonctionnelle sans prendre en compte les attributs de QoS.

Sélection de services basée sur la QoS

La phase de sélection de services dans le processus de composition vise à choisir le service idéal pour chaque tâche abstraite en fonction de leurs critères de Qualité de Service (QoS). Une fois que les services appropriés sont sélectionnés pour chaque tâche abstraite, cela définit une composition candidate. Nous nous focalisons sur la problématique de sélection de services

lors du processus de composition avec prise en compte des critères de QoS.

Il existe deux types de sélection : locale et globale. La sélection locale se concentre sur le choix du meilleur service candidat pour chaque tâche abstraite dans le plan de composition. Cependant, elle ne garantit pas toujours que la composition globale respectera les contraintes globales de QoS définies par l'utilisateur.

En revanche, la sélection globale prend en compte les contraintes de QoS au niveau de la composition entière. Une approche directe consiste à évaluer toutes les combinaisons possibles de services pour déterminer la meilleure en termes de QoS, tout en garantissant le respect des contraintes globales définies.

Exécution et contrôle de la composition

Pendant cette phase, la composition candidate précédemment définie est exécutée par un moteur d'exécution chargé d'appeler les services composants. Cette exécution de la composition peut être vue comme une série d'échanges de messages entre les services composants. Cette composition est mise à disposition d'autres utilisateurs qui pourront l'appeler en tant que nouveau service.

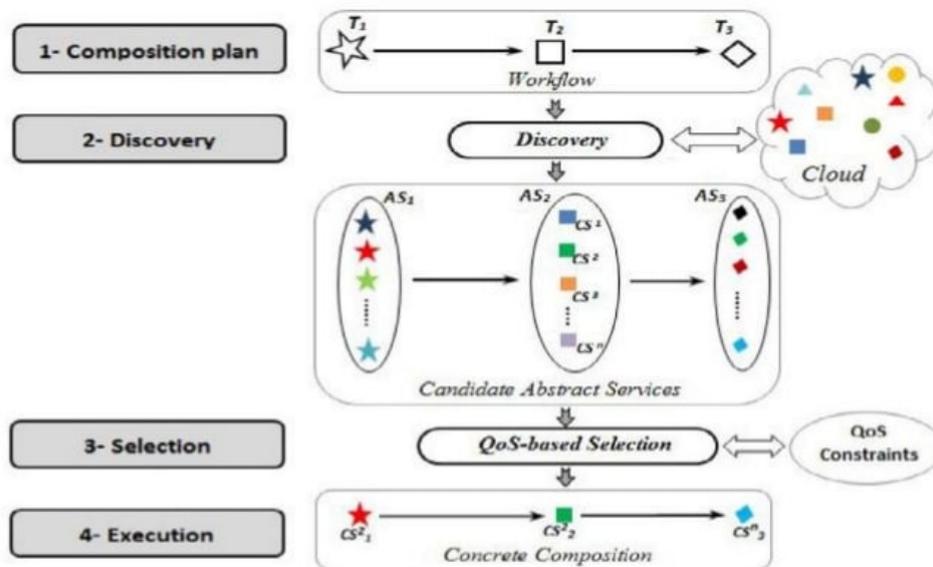


FIGURE 2.2 – Aperçu de la composition sensible à la QoS [15]

2.3.3 Définitions des concepts liés à la problématique de composition de services

Service candidat (SC)

Un service candidat est un service potentiellement approprié pour être utilisé dans une composition. Il est encore à un stade de proposition ou de considération. Les services candidats peuvent être des abstractions ou des descriptions de services qui pourraient répondre aux besoins fonctionnels de l'utilisateur.

Un service concret, en revanche, est un service candidat qui a été sélectionné pour accom-

plir une tâche spécifique au sein d'une composition concrète. Il fait partie intégrante de la composition en cours d'utilisation[16].

Service abstrait (AS)

Un service abstrait peut être défini comme une catégorie de services candidats partageant des fonctionnalités équivalentes. Plus précisément, nous pouvons décrire un service abstrait, noté AS_i , en considérant quatre éléments essentiels[16], comme indiqué dans la Figure 2.3 :

Les messages d'entrée (AS_i^{in}).

Les messages de sortie (AS_i^{out}).

La catégorie de service (AS^{cat})

Une liste de services candidat (AS_i^{CS}) appartenant au service abstrait AS_i

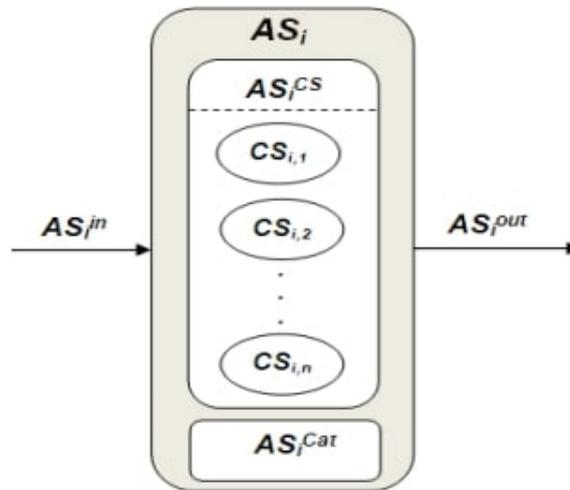


FIGURE 2.3 – Représentation d'un service abstrait [16]

Les services composites abstraits (SCA)

Un service composite abstrait (SCA) aussi dit composition abstraite, est un ensemble de m services abstraits. Un SCA est représenté sous la forme d'un vecteur de m AS : $\langle AS_1, AS_2, \dots, AS_m \rangle$. Ils sont interconnectés en utilisant l'une des structures de composition de service disponibles, telles que la structure séquentielle, parallèle, conditionnelle et/ou en boucle. L'élaboration du SCA est une réponse aux exigences fonctionnelles spécifiés par l'utilisateur c'est le résultat de la première étape du processus de composition. Elle consiste à relier un ensemble de services abstraits de manière à répondre aux fonctionnalités requises. Cette composition d'un SCA permet de combiner les fonctionnalités offertes par divers services existants, qu'ils soient atomiques ou composites, dans le but de satisfaire des besoins plus complexes.[15].

2.3.4 Composition concrète (comp)

La composition concrète (ou tout simplement composition), est définie comme un vecteur qui contient les indices des services candidats inclus dans cette composition. Cette représentation

vectorielle nous permet de modéliser et de décrire de manière concise les compositions de services[15].

2.3.5 Les critères de QoS

La qualité de service, abrégée QoS, fait référence à la capacité à fournir un service conforme aux exigences et aux préférences de l'utilisateur, par exemple en ce qui concerne la disponibilité [11]. Les critères de QoS se divisent en deux catégories : les attributs positifs, tels que la disponibilité, le débit et le succès, doivent être maximisés, tandis que les attributs négatifs, comme la latence, doivent être minimisés [15]. Nous allons fournir une définition de certains critères de qualité de service dans ce suit.

- Temps de réponse : Temps nécessaire pour envoyer une demande et recevoir une réponse.
- Disponibilité : Nombre d'appels réussis/nombre total d'appels.
- Débit : Nombre total d'appels pour une période de temps donnée.
- Succès : Nombre de réponses / nombre de messages de requête .
- La fiabilité : Rapport entre le nombre de messages d'erreur et le nombre total de messages.

2.3.6 Vecteur QoS d'un service candidat

Les valeurs de QoS d'un service candidat sont représentées sous la forme d'un vecteur. Ce vecteur englobe un ensemble de valeurs ou de mesures qui permet d'évaluer le niveau de qualité offert par ce service. Ce vecteur comprend divers propriétés non-fonctionnelles tel que la performance, la fiabilité et la disponibilité [15].

2.3.7 Vecteur d'attributs QoS d'une composition

Le vecteur QoS d'une composition est défini comme suit : $QoS_{comp} = [Q_1, \dots, Q_q]$, où q est le nombre d'attributs QoS et Q_p est le valeur du p^{th} attribut après agrégation. L'agrégation de attributs dépend de la fonction utilisée (Somme, Produit, Maximum ou Minimum)[15] comme illustre dans le tableaux, comme illustré dans le tableau 2.1.

Type	Fonction d'agrégation
Temps de réponse (Rs)	$Rs_{cs} = \sum_{i=1}^n Rs_j$
Disponibilité (Di)	$Di_{cs} = \prod_{i=1}^n Di_j$
Débit (D)	$D_{cs} = \min(D_1, D_2, ..D_n)$
Succès (S)	$S_{cs} = \prod_{i=1}^n S_j$
Fiabilité (F)	$F_{cs} = \prod_{i=1}^n F_j$
Conformité (C)	$C_{cs} = \max(C_1, C_2, ..C_n)$
Les meilleures pratiques (Mp)	$Mp_{cs} = \max(Mp_1, Mp_2, ..Mp_n)$
Latence (L)	$L_{cs} = \sum_{i=1}^n L_j$

Documentation (Dc)	$Dc_{cs} = \max(Dc_1, Dc_2, ..Dc_n)$
--------------------	--------------------------------------

TABLE 2.1 – Fonction d'agrégation des propriétés QoS [17][14]

2.3.8 Fonction d'utilité

Une fonction d'utilité est un outil essentiel dans le processus de sélection de compositions, car elle permet d'attribuer une mesure quantitative à chaque composition candidate. Cette mesure est basée sur les critères de qualité de service (QoS) et les préférences de l'utilisateur. En fin de compte, la fonction d'utilité facilite le choix de la meilleure composition pour répondre aux besoins spécifiques de l'utilisateur.[12]. La fonction d'utilité F d'un service composite $S_{composite}$ est calculée en utilisant la formule suivante [11] :

$$F_i = \sum_{j=1}^{nb_qos} w_j * Q'_j \quad (2.1)$$

nb_qos est le nombre de paramètres QoS ;

w_j est le poids du $j^{ème}$ attribut de QoS ;

Q'_j est la valeur agrégée du $j^{ème}$ attribut de QoS.

Notre objectif est de rechercher une composition quasi-optimale qui minimise la fonction d'utilité F (également appelée fitness).[12].

Pour exprimer les mesures de service en termes de QoS, il faut d'abord attribuer un poids à chaque critère de QoS. Ensuite, la fonction fitness calcule le score de chaque service en faisant la somme pondérée des valeurs des critères de QoS normalisés[11].

2.4 Machine learning

L'une des techniques employées lors de la résolution du problème composition de services sensible aux QoS est l'apprentissage supervisé, qui fait partie des techniques du Machine Learning. Ces techniques sont souvent utilisées pour les espaces de recherches à large échelle dans le but de réduire l'espace de recherche en gardant les meilleurs services. Dans la suite, nous allons présenter le Machine Learning et ses différents types.

2.4.1 Définition

Le Machine Learning, également appelé apprentissage automatique, est une technologie qui repose sur des algorithmes capables d'apprendre automatiquement à partir de données. Pour ce faire, des données sont nécessaires, puis elles sont organisées et utilisées pour former le système, lui permettant ainsi de classer de nouvelles données de manière autonome. En fonction de cette classification, le système peut ensuite effectuer des actions programmées [18].

2.4.2 Types de systèmes d'apprentissage automatique

Les systèmes d'apprentissage automatique peuvent être regroupés en quatre catégories principales en fonction du degré de supervision nécessaire pendant la phase d'entraînement. Ces

catégories comprennent l'apprentissage supervisé, l'apprentissage non supervisé, l'apprentissage semi-supervisé et l'apprentissage avec renforcement[18].

2.4.2.1 Apprentissage non supervisé

L'apprentissage non supervisé se caractérise par l'utilisation exclusive de données d'entrée (X) sans avoir de variables de sortie correspondantes, comme indiqué dans la Figure 2.4. Son objectif principal est de modéliser la structure ou la distribution sous-jacente des données pour en extraire des informations. Il porte ce nom car, contrairement à l'apprentissage supervisé, il ne dispose pas de réponses correctes ou de données d'enseignement. Ces algorithmes fonctionnent de manière autonome pour découvrir et mettre en évidence des structures significatives dans les données [18].



FIGURE 2.4 – Apprentissage non supervisé
[18]

2.4.2.2 Apprentissage supervisé

L'apprentissage supervisé est basé sur des variables d'entrée (x) et des variables de sortie (Y), comme illustré dans la Figure 2.5. Grâce à un algorithme, ce type d'apprentissage vise à établir une correspondance entre les données d'entrée et de sortie.

L'objectif principal de l'apprentissage supervisé est de comprendre cette correspondance de manière à pouvoir mesurer des valeurs de sortie (Y) lorsque de nouvelles données d'entrée (x) sont fournies.

On le qualifie d'apprentissage supervisé parce que tout au long du processus, l'algorithme apprend en se basant sur un ensemble d'apprentissage où les bonnes réponses sont déjà connues. On peut considérer cet ensemble d'apprentissage comme étant l'enseignant qui supervise le processus. Nous disposons des réponses correctes, et l'algorithme effectue des calculs itératifs sur les données d'apprentissage, avec des corrections fournies par l'ensemble d'apprentissage. L'apprentissage prend fin lorsque l'algorithme atteint un niveau de performance satisfaisant.



FIGURE 2.5 – Apprentissage supervisé
[18]

Les algorithmes d'apprentissage automatique supervisé sont largement utilisés, et il existe deux catégories principales dans l'apprentissage supervisé : la classification et la régression.[18].

La classification

La classification implique l'apprentissage d'une fonction objective, notée f , qui attribue un ensemble d'attributs x à l'une des classes préétablies étiquetées y . On peut également désigner cette fonction f comme un modèle de classification. Ce modèle de classification trouve des applications dans divers contextes[19].

L'objectif principal de la classification est d'identifier la classe à laquelle un objet appartient en se basant sur des caractéristiques descriptives.

Toutes les méthodes de classification partagent une caractéristique commune : elles nécessitent un processus d'apprentissage et d'évaluation de leurs performances. Pour cela, Nous avons besoin de deux ensembles d'échantillons[19].

- La base d'apprentissage doit inclure un nombre adéquat d'échantillons x_i pour chaque classe. Cette collection d'échantillons doit être à la fois qualitative, c'est-à-dire représentative de l'application envisagée, et quantitative, avec un nombre significatif d'échantillons tout en tenant compte des probabilités a priori de chaque classe.
- La base de test doit être sélectionnée avec soin, car elle sert à évaluer l'algorithme.

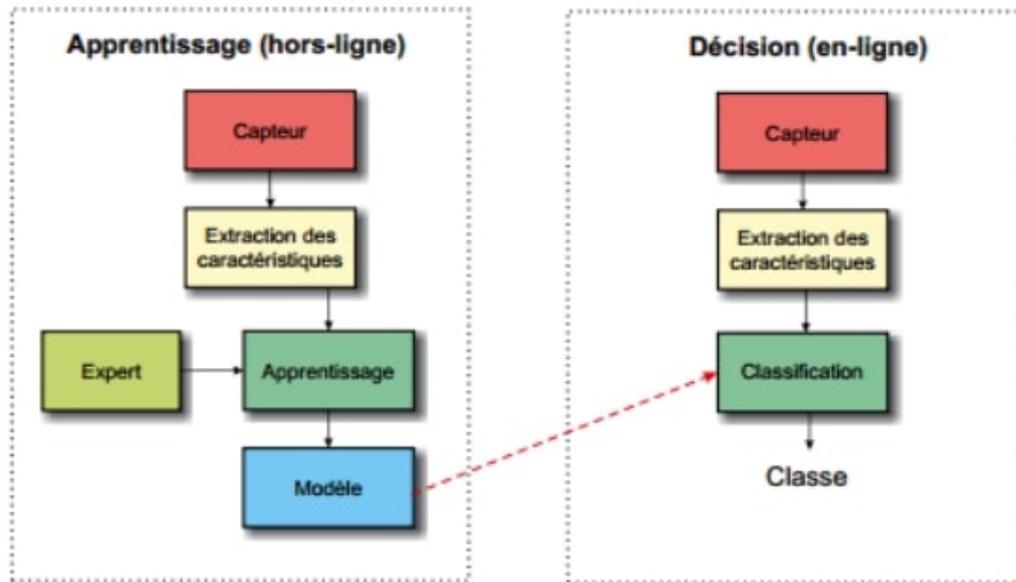


FIGURE 2.6 – Processus classique de reconnaissance par apprentissage [19]

Classiquement, le processus se déroule en deux étapes principales : l'apprentissage(hors ligne) et la décision (en ligne), comme illustré dans la figure 2.6. Le module d'apprentissage permet de construire des modèles, qui sont des ensembles de règles ou de critères de décision. Ces modèles sont élaborés en utilisant des données supervisées, c'est-à-dire des données pour lesquelles nous avons une connaissance préalable des classes d'échantillons fournies par des experts. Ensuite, un classificateur utilise ce modèle pour prendre une décision concernant la classe à laquelle appartient un échantillon inconnu, représenté par ses attributs [19].

la régression

Un problème de régression se pose lorsque la variable de sortie est une valeur réelle[19].

2.4.2.3 Apprentissage semi-supervisé

L'apprentissage semi-supervisé a pour objectif de résoudre des problèmes impliquant des données non étiquetées en utilisant des ensembles d'informations partiellement étiquetées[19].

2.4.2.4 Apprentissage par renforcement

L'apprentissage par renforcement est une approche de l'intelligence artificielle axée sur l'apprentissage d'un système par le biais de son interaction avec l'environnement. Dans ce processus, le système adapte ses paramètres en fonction des retours qu'il reçoit de l'environnement et émet à son tour des réponses en fonction des décisions qu'il a prises[19].

2.5 Métaheuristique

Lors de nos recherches nous avons constater que les métaheuristiques ont largement été utilisées pour la résolution du problème de composition de services IoT sensibles aux QoS. En

effet, les approches basées sur des métaheuristiques retournent une solution de bonne qualité en un temps d'exécution raisonnable.

2.5.1 Définition

Les métaheuristiques sont des stratégies qui permettent de guider la recherche d'une solution optimale. Le but visé par les métaheuristiques est d'explorer l'espace de recherche efficacement afin de déterminer des solutions (presque) optimales. Les techniques qui constituent des algorithmes de type métaheuristiques vont de la simple procédure de recherche locale à des processus d'apprentissage complexes. Les métaheuristiques sont en général non déterministes et ne donnent aucune garantie d'optimalité [20].

2.6 Etat de l'art sur les approches de composition de services IoT basées sur les QoS

Dans cette section, nous présentons des travaux de recherches traitant la composition de services IoT. Pour cela, un éventail d'approches, qu'elles soient basées sur des techniques déterministes ou non déterministes a été étudié.

2.6.1 Approches basées sur des techniques déterministes

Lors de la réalisation de notre travail, nous avons étudié certains travaux proposant des solutions basées sur des méthodes déterministes. Ces méthodes suivent une procédure stricte et prédéfinie pour résoudre les problèmes d'optimisation, ce qui les rend particulièrement efficaces pour les problèmes de petite taille, car elles examinent l'intégralité de l'espace de recherche. En utilisant un algorithme déterministe, il est possible de mesurer avec précision les performances et la qualité de service en suivant des étapes bien définies dans le processus de composition des services[21].

Dans cette étude [22], les chercheurs ont présenté une approche visant à résoudre plusieurs défis liés aux systèmes cyber-physiques-sociaux, notamment le coût élevé en temps et la faible fiabilité. L'approche proposée, appelée FRSkyline, repose sur plusieurs étapes clés. Tout d'abord, elle utilise le calcul des composants Skyline pour éliminer les composants redondants. Ensuite, elle emploie le coefficient de variation pour évaluer la fluctuation de la qualité de service et garantir la fiabilité. Enfin, les chercheurs ont recours à la programmation mixte en nombres entiers pour déterminer la meilleure solution, tout en réduisant le temps de calcul. Les résultats expérimentaux ont démontré que cette approche offre une fiabilité améliorée tout en demandant moins de temps que d'autres méthodes. Cependant, il est important de noter qu'elle peut ne pas convenir aux nouveaux composants ou à ceux qui sont rarement utilisés, et qu'elle peut ne pas parvenir à trouver une solution lorsque le nombre de composants est limité.

Dans cette recherche [23], les auteurs ont proposé un algorithme de composition de services qui utilise des techniques de sélection partielle pour réduire l'espace de recherche. Cette approche tire parti de l'élagage et de la parallélisation des composants à plusieurs niveaux. L'algorithme proposé, appelé DPSA, est capable d'améliorer considérablement l'efficacité tout

en garantissant l'optimalité de la composition des services. Cependant, il présente des limitations lorsque le nombre de candidats est élevé, car il nécessite un temps d'exécution important.

Dans cette recherche [24], les auteurs ont développé un cadre axé sur la composition de services sensibles au contexte en utilisant un modèle de service abstrait appelé wEASEL. Ce modèle de service abstrait représente les services et les tâches de l'utilisateur en termes de signature, de spécification et de conversation. L'évaluation de cette approche a montré que le système basé sur wEASEL permet une composition plus précise des services et offre aux utilisateurs finaux la possibilité de découvrir et d'explorer davantage d'opportunités de composition par rapport à d'autres méthodes.

2.6.2 Approches basées sur des techniques non déterministes

Les techniques non déterministes peuvent produire des résultats différents pour une même entrée lors d'exécutions différentes. Le comportement probabiliste de ces techniques dépend généralement d'un générateur de nombres aléatoires. Ces approches se révèlent efficaces en termes de temps de calcul, même lorsque l'espace de recherche est vaste. Cette caractéristique est illustrée par les algorithmes métaheuristiques, qui font partie intégrante de cette catégorie de techniques non déterministes[21].

Dans ce qui suit nous présentons certains travaux proposant des approches basées sur des métaheuristiques pour la résolution du problème de sélection de services web/IoT avec prise en compte des contraintes QoS.

Dans cette étude [21], les auteurs ont développé une méthode pour composer les meilleurs services en utilisant Hidden Markov Model (HMM) et l'algorithme artificiel Ant Colony Optimization (ACO). Cette approche sélectionne et combine les services en fonction des priorités de la QoS afin d'améliorer la qualité globale et d'atteindre l'optimalité. L'approche, appelée HMAc (Hybrid Markov-Ant Colony), bénéficie de la capacité robuste de ACO à explorer de manière étendue, combinée à la capacité de sélection de la QoS offerte par le modèle HMM. Les résultats des expériences ont montré que HMAc est excellent pour réduire les coûts tout en offrant une meilleure disponibilité, fiabilité et des performances plus rapides en termes de temps de réponse par rapport à d'autres méthodes. Cependant, il convient de noter que l'efficacité énergétique de HMAc est moins satisfaisante que celle d'autres approches.

Dans cette étude [25], les chercheurs ont examiné l'efficacité de l'algorithme Gray Wolf Optimizer (GWO) pour composer des services web tout en tenant compte de la qualité de service. Quatre paramètres de QoS ont été pris en compte pour évaluer GWO : le temps de réponse, la fiabilité, la disponibilité et le coût. Les résultats des expérimentations ont montré que l'algorithme GWO était approprié pour produire des résultats efficaces, quel que soit le nombre d'itérations de l'algorithme. Cependant, un inconvénient de cette méthode est qu'elle tend à privilégier le service web ayant la meilleure valeur de "fitness" comme solution suggérée. Cela signifie qu'un service web optimal peut être choisi même s'il existe d'autres solutions similaires avec des valeurs de "fitness" légèrement inférieures mais qui pourraient être plus adaptées aux besoins des utilisateurs. En conséquence, ces solutions alternatives pourraient ne pas être prises en compte.

Dans ce papier [26], les chercheurs ont développé un modèle de sélection de services prenant en compte les contraintes de QoS dans un environnement IoT. Ce modèle utilise une méta-

heuristique d'optimisation associée à une méthode d'évaluation sur une échelle de mesure de Likert. L'objectif principal était d'améliorer les performances d'un algorithme d'optimisation bio-inspiré appelé Social Spider Optimization (SSO) en introduisant une notion de réputation pondérée pour les membres. Le modèle ainsi créé a été nommé RI-SSO, signifiant "Reputation Improved-Social Spider Algorithm".

Pour évaluer la réputation des services, les chercheurs ont utilisé une échelle de mesure de Likert. Les résultats de comparaison ont montré que RI-SSO était performant pour les problèmes d'optimisation, qu'il s'agisse de maximisation ou de minimisation. Il se démarquait notamment par ses valeurs de fitness élevées, ce qui en faisait un choix approprié pour optimiser la sélection de services en garantissant une disponibilité et une fiabilité élevées tout en maintenant des coûts et des temps d'exécution faibles.

Dans cette étude [13], les chercheurs ont développé un modèle bi-objective shortest path optimization (BSPO) pour la composition de services IoT, en prenant en compte à la fois la consommation d'énergie et la qualité de service. L'objectif principal du BSPO est de minimiser la consommation d'énergie du service composite tout en minimisant les aspects de la QoS, tels que le temps d'exécution, la latence réseau et le coût du service. Pour résoudre ce problème complexe, les chercheurs ont proposé un algorithme d'impulsion, associé à quatre techniques d'élagage intégrées (The pulse algorithm with four embedded pruning techniques) visant à réduire la complexité de l'algorithme. Cette approche se distingue d'autres schémas qui ne considéraient que la QoS ou la consommation d'énergie de manière individuelle. En effet, cette nouvelle approche parvient à obtenir de bonnes performances en termes de consommation d'énergie, ce qui prolonge la durée de vie du réseau, tout en maintenant un niveau de QoS raisonnable.

Dans ce travail [27], les chercheurs ont introduit un modèle mathématique prenant en compte la qualité de service floue, qui prend en considération les préférences des entreprises manufacturières en attribuant différents poids à diverses sous-tâches pour évaluer la QoS globale floue des Systèmes de Fabrication de Services (solutions to the manufacturing service composition SMSC). Ensuite, ils ont amélioré l'algorithme flower pollination (FPA) pour obtenir de manière plus efficace un SMSC optimal. Cela a été réalisé en rendant la probabilité de commutation auto-adaptative, en améliorant la capacité de recherche locale et en ajoutant une stratégie de remplacement d'élite. Les résultats de simulations effectuées ont indiqué que cet algorithme amélioré, appelé EFPA, s'avère efficace pour résoudre le problème de composition de services de fabrication.

Les auteurs ont présenté dans [28] une méthode efficace pour résoudre le problème de la composition optimale des services sensibles aux ressources et à la qualité de service (RQ-SOC) dans le domaine du Big Service et de l'Internet des Objets. Pour ce faire, ils ont commencé par améliorer l'algorithme artificial bee colony (ABC) en se basant sur les caractéristiques de service dans divers domaines de service (SFSD). Ensuite, ils ont introduit un opérateur de vérification des ressources, qu'ils ont intégré à l'algorithme ABC amélioré, créant ainsi un nouvel algorithme appelé S-ABCRRSC spécialement conçu pour résoudre les problèmes de RQ-SOC. Les résultats des expérimentations ont démontré que l'algorithme S-ABCRRSC se distingue par ses performances, son efficacité, et sa capacité à identifier des schémas de composition de services de haute qualité en un temps d'exécution réduit.

Dans cette étude [17], les chercheurs ont présenté une approche de sélection de services qui combine l'utilisation d'une échelle de Likert pour prendre en compte les commentaires des utilisateurs finaux avec l'algorithme amélioré de Practical Swarm Optimization (PSO amélioré). L'objectif était de résoudre le problème de la sélection de services tout en améliorant les performances en termes de temps de sélection. Cette approche a permis d'obtenir des solutions excellentes avec des temps d'exécution réduits. Les résultats ont montré que cette méthode atteignait les meilleures valeurs de fitness et qu'elle était plus efficace en général.

Dans ce papier [29], les auteurs ont développé un modèle d'évaluation de la QoS et un modèle d'évaluation énergétique pour la composition de services basée sur la QoS. Ils ont ensuite appliqué avec succès l'algorithme Month Flame Optimization (MFO) à ce modèle. Les résultats de la simulation démontrent que cette approche qui est plus efficace et performante, permet d'atteindre une efficacité énergétique proche de l'optimale, une meilleure capacité d'optimisation globale, et a une plus grande probabilité de trouver la solution quasi optimale au problème de composition de services.

2.6.3 Comparaison entre les approches métaheuristiques et les approches déterministes

Après avoir analysé des travaux basés sur de techniques déterministes et d'autres basés sur de métaheuristiques, nous déduisant que les métaheuristiques se distinguent des méthodes déterministes par leur approche flexible et adaptative pour résoudre des problèmes complexes. Contrairement aux méthodes déterministes, telles que la programmation linéaire ou la recherche exhaustive, qui suivent des algorithmes basés sur des règles strictes, les métaheuristiques exploitent des stratégies plus évolutives. Les méthodes traditionnelles sont souvent efficaces pour des problèmes bien structurés avec des espaces de recherche restreints, mais elles peuvent être inadaptées aux défis complexes présentant des espaces de recherche vastes et mal structurés. Les métaheuristiques, en revanche, font preuve d'une grande adaptabilité et d'une capacité à explorer l'espace de recherche de manière aléatoire et itérative, ce qui leur permet de trouver des solutions de qualité dans des contextes difficiles. Cette comparaison met en évidence le fait que les méta-heuristiques excellent dans la résolution de problèmes complexes où les méthodes traditionnelles atteignent leurs limites, faisant ainsi des méta-heuristiques un choix précieux dans de nombreux domaines d'application, y compris pour la résolution du problème de sélection de services IoT lors du processus de composition avec prise en compte des QoS [20].

Nous clôturons cette section par une étude comparative des travaux traitant le problème de composition de services IoT étudiés. Les critères de comparaisons des approches sont présentés ci dessous.

- **Année et Auteur :** Dans le contexte de la composition de services IoT, ce critère fait référence à l'année de publication de l'article ainsi qu'à l'auteur ou aux auteurs de cet article. Il permet d'identifier la source de l'information et de situer chronologiquement la recherche, ce qui peut être utile pour évaluer la pertinence et la fiabilité des informations dans le domaine de la composition de services IoT.
- **l'approche proposée :** L'approche proposée se réfère à la méta-heuristique sur laquelle repose la méthode de composition des services IoT étudiée.

- **Sensible à la qualité de service** : La considération de la qualité de service, également appelée sensibilité à la QoS (Quality of Service awareness), implique que l'algorithme de composition prend en considération les critères de QoS lors de la phase de sélection de services.
- **Méthodes d'optimisation** : Les méthodes d'optimisation de la composition des services IoT sont des techniques qui utilisent des algorithmes d'optimisation pour trouver la meilleure combinaison de services pour répondre aux objectifs spécifiques des utilisateurs. Il existe deux principaux types de méthodes d'optimisation : les méthodes déterministes et les méthodes non déterministes.
- **Les avantages et les limites** : Les avantages désignent les aspects positifs ou les points forts de l'approche proposée tandis que les limites font référence aux inconvénients ou aux aspects négatifs de cette approche.
- **Réduction de l'espace de recherche** : La réduction de l'espace de recherche dans le problème de composition des services IoT se réfère à la pratique de restreindre délibérément le nombre de combinaisons possibles de services à considérer lors la recherche de solutions. Cela vise à optimiser l'efficacité du processus de composition en éliminant les options peu prometteuses et en se concentrant sur celles qui ont le potentiel d'aboutir à des résultats de meilleure qualité.

Le tableau 2.2 illustre une étude comparative des travaux passés en revues.

Auteurs + Année	Approche Proposée	QoS	Optimisation	Reduction de l'espace de recherche	Avantages/Limites
2021 [SeyedSalar Sefati, Nima Jafari Navimipour] [21]	L'algorithme HMAC	Oui	Non déterministe	Non	Bonnes performances en termes de temps de réponse, mais son efficacité en matière de consommation d'énergie est insuffisante.
2017 [Meysam Karimi, Seyed Morteza Babamir] Article [25]	L'algorithme Grey Wolf Optimizer (GWO)	Oui	Non déterministe	Non	Il excelle en termes d'efficacité et de performances, mais sa sélection se base uniquement sur la fitness et néglige la convivialité.
2017 [S. Wang, A. Zhou, M. Yang, L. Sun, C. Hsu, F. Yang] Article [22]	FRSkyline	oui	Déterministe	Oui	Offre une fiabilité accrue, mais elle n'est pas adaptée pour les nouveaux composants ou ceux qui sont rarement utilisés

2020 [Dr. Salah Elfaki Elrofai] Article [26]	Algorithme Reputation Improved-Social Spider Optimization (RI-SSO)	Oui	Non déterministe	Non	Elle atteint de meilleures performances en ce qui concerne les valeurs de fitness.
2018 [osama alsaryrah, ibrahim mashal, tein-yaw chung] Article [13]	L'algorithme d'impulsion avec quatre techniques d'élagage intégrées	Oui	Non déterministe	Non	Il peut offrir de bonnes performances en matière de consommation d'énergie tout en maintenant un niveau de QoS raisonnable.
2019 [Shuai Zhang, Yangbing Xu, Wenyu Zhang, Dejian Yu] Article [27]	EFPA	Oui	Non déterministe	Non	Il s'avère efficace et efficient pour résoudre le problème de la composition des services de fabrication, mais il néglige les corrélations entre les services manufacturiers.
2014 [Ying Chen, Chuang Lin, Jiwei Huang, Jie Hu] Article [23]	DPSA (Distributed Partial Selection Algorithme)	Oui	Déterministe	Oui	Peut considérablement améliorer l'efficacité tout en garantissant l'optimalité de la composition des services, mais elle demande beaucoup de temps d'exécution lorsque le nombre de candidats est élevé.
2018 [xunyou min, xiaofei xu, zhizhong liu, dianhui chu, zhongjie wang] Article [28]	RQ-SOC appelée (S-ABCRRSC)	Oui	Non déterministe	Non	Il offre de bonnes performances et se distingue par son efficacité.
2020 [Aghabi N. Abu-Safe, Salah Elfaki Elrofai] Article [17]	Mesure d'échelles de Likert avec un algorithme d'optimisation d'essaim pratique amélioré (PSO amélioré)	Oui	Non déterministe	Oui	Il atteint de meilleures valeurs de fitness et excelle en termes d'efficacité.

2017 [A. Urbietta, A. Gonzalez-Beltrán, S. Ben Mokhtar, M. Anwar Hossain, L. Capra] Article [24]	un cadre qui traite de la composition de services sensibles au contexte basés sur un modèle de service abstrait wEASEL	Non	Déterministe	Non	Elle permet aux utilisateurs finaux d'explorer plus largement les opportunités de composition que d'autres approches, mais elle souffre de temps d'exécution réduit.
2020 [Jianhao Gao] Article [29]	MFO (month flame optimization)	Oui	Non déterministe	Non	Il se distingue par son efficacité et ses performances.
2023 DT-WOA	arbre de décision+ WOA discrétisé	Oui	Non déterministe	Oui	Offre des résultats satisfaisants en termes de QoS comparé à WOA.

TABLE 2.2 – Tableau comparatifs des approches étudiées

Discussion

En comparant les travaux, il est clair que le problème de composition de services sensibles à la qualité de service a fait l'objet d'une attention considérable. La QoS est un critère largement adopté dans ces approches [[21], [25], [22], [26], [13], [27], [23], [28], [17]], [29] permettant de répondre efficacement aux exigences de l'utilisateur tout en prenant en compte ses préférences. Les méthodes d'optimisation les plus couramment utilisées dans ces travaux sont les méthodes non déterministes. Des approches telles que [[21], [25], [26], [13], [27], [28], [17], [29]] reposent sur l'utilisation d'algorithmes métaheuristiques. Les méta-heuristiques sont conçus pour explorer efficacement des espaces de recherche vastes de manière aléatoire, cela leur permet de garantir un bon compromis entre l'exploration et l'exploitation de l'espace de recherche. Leur objectif est de trouver une composition quasi-optimale de bonne qualité en un temps raisonnable.

D'un autre côté, les approches déterministes comme [[22], [23], [24]] suivent une procédure stricte et définie pour résoudre les problèmes de composition de services. Ces méthodes examinent de manière toutes les combinaisons de services possibles. Bien que cela puisse garantir une solution optimale dans certains cas, cela peut devenir rapidement inefficace pour des problèmes complexes en raison de l'explosion combinatoire.

La réduction de l'espace de recherche, quant à elle, est adoptée dans ces approches [[22], [23], [17]] pour limiter le nombre de combinaisons de services examinées lors de la composition. Cela permet de réduire la complexité du processus et d'accélérer la recherche de solutions.

2.7 Conclusion

La composition des services est un processus utilisé pour satisfaire les demandes complexes des utilisateurs. Ce processus se déroule en quatre étapes, la définition du plan de compo-

sition, la découverte des services candidats, la sélection de composition et l'exécution de la composition. Dans ce mémoire, nous traitant le problème de sélection de composition avec prise en compte des QoS et des préférences des utilisateurs. Nous avons présenté les concepts fondamentaux des algorithmes d'apprentissage en mettant l'accent sur les arbres de décisions puis nous sommes passé à la définition des métaheuristiques et de certains travaux traitant le problème de composition de services IoT. Nous clôturons ce chapitre par une comparaison et une synthèse sur les travaux étudiés.

Chapitre III

Approche proposée

3.1 Introduction

Comme nous l'avons déjà mentionné dans le chapitre 2, la composition de services offre la possibilité de répondre à des requêtes complexes en combinant les fonctionnalités de plusieurs services, créant ainsi de nouveaux services composites réutilisables. Le défi de la composition de services réside dans le fait de satisfaire les besoins de l'utilisateur en considérant plusieurs critères de QoS de différents types (à maximiser et à minimiser) et tout cela dans un vaste espace de recherche. Cela fait du problème de composition de services sensible aux critères de QoS un problème Np-difficile. Dans ce chapitre, nous présentons notre approche nommée Decision Tree with Whale Optimization Algorithm (DT-WOA) qui combine un arbre de décision avec l'algorithme WOA pour la résolution du problème de composition de services IoT sensibles aux QoS. Notre objectif est de trouver une solution quasi optimale. L'arbre de décision est employé pour classer les services IoT en utilisant leurs attributs QoS et WOA est utilisé pour sélectionner une solution sous-optimale de bonne qualité en un temps de réponse raisonnable.

3.2 Motivation

Dans le but de réduire l'espace de recherche et d'optimiser le temps d'exécution, nous optons pour une approche consistant à classer les services IoT, puis à sélectionner ceux bénéficiant des meilleurs niveaux d'évaluation de qualité globale. Bien qu'il existe diverses approches en matière d'apprentissage automatique, telles que les réseaux de neurones, les arbres de décision, le clustering, les réseaux bayésiens, l'apprentissage par renforcement, les machines à vecteurs de support, et bien d'autres, notre choix s'est porté sur l'utilisation de la méthode des arbres de décision. Cette approche nous permet de résoudre efficacement le défi de la réduction de l'espace de recherche dans le contexte de l'IoT. Les arbres de décision se révèlent être une technique d'apprentissage automatique facile à mettre en œuvre, relativement simple à comprendre et à déboguer. De plus, ils offrent une grande flexibilité en matière de personnalisation et figurent généralement parmi les méthodes les plus rapides en termes d'apprentissage et de classification.

la combinaison de l'arbre de décision avec l'algorithme métaheuristique d'optimisation des baleines WOA (Whale Optimization Algorithm) lors de la recherche d'une composition sous-optimale nous permet de garantir une bonne balance entre l'exploration et l'exploitation de l'espace de recherche et ainsi de tester un grand nombre de composition pour trouver une

bonne solution satisfaisant l'utilisateur en terme de qualité et de temps de réponse.

3.3 Aperçu sur les arbres de décision

3.3.1 Définition

Les arbres de décision sont des algorithmes d'apprentissages utilisée dans l'exploration de données et en informatique décisionnelle. Ils sont utilisés à la fois pour les tâches de classification et de régression. Ils apprennent à partir d'observations, appelées exemples, où chaque exemple est caractérisé par une série d'attributs et une classe associée. Il est important de noter que les arbres de décision fonctionnent en mode supervisé, ce qui signifie que la classe associée à chaque exemple est connue. Un arbre de décision est un moyen efficace d'illustrer les inférences qui permettent de distinguer les similitudes et les différences entre les attributs des exemples au sein d'un ensemble de données. Les statisticiens les utilisent fréquemment pour visualiser les résultats de leurs analyses.

La structure d'un arbre de décision comprend des nœuds d'arbre, où le nœud en haut de l'arbre est appelé la racine, et chaque nœud situé sous la racine est soit une feuille, soit le sommet d'un sous-arbre[30].

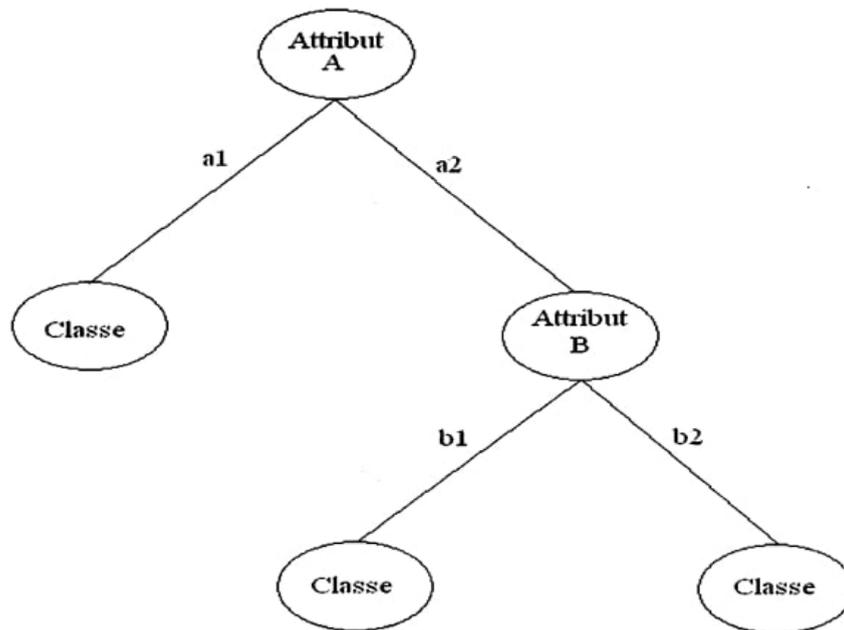


FIGURE 3.1 – Schéma d'un arbre de décision [30]

Dans l'arbre de décision illustré dans la figure 3.1, les attributs A et B possèdent chacun deux valeurs distinctes. Lorsque les exemples ont l'attribut A égal à a_1 , ils appartiennent à une seule classe. Cependant, lorsque l'attribut A est égal à a_2 , les exemples sont répartis en deux classes différentes. Dans ce cas, il est nécessaire d'utiliser l'attribut B pour diviser les exemples en fonction de leurs classes respectives[30].

3.3.2 Construction d'un arbre de décision

L'utilisation d'arbres de décision dans les problèmes de classification comporte deux étapes principales :

- La construction d'un arbre de décision à partir d'une base d'apprentissage.
- La classification ou l'inférence consistant à classer une nouvelle instance à partir de l'arbre de décision construit dans la première étape.

La construction d'un arbre de décision est basée sur un ensemble d'apprentissage donné. Cette procédure implique la sélection d'un attribut de test approprié pour chaque nœud de décision, suivi de la définition de la classe de chaque feuille de l'arbre d'induction.

Pour accomplir cette tâche, plusieurs algorithmes ont été développés pour la phase de construction. Parmi les algorithmes non-incrémentaux, on peut citer ID3 et C4.5, qui ont été développés par Quinlan, ainsi que l'algorithme CART de Breïman et ses collaborateurs, ces derniers étant parmi les plus populaires[31].

Nous avons choisi d'utiliser l'algorithme WOA (Whale Optimization Algorithm) pour notre problème de composition de services en raison de sa capacité à gérer efficacement les problèmes d'optimisation complexes, tout en offrant la flexibilité nécessaire pour intégrer des techniques d'apprentissage automatique et des arbres de décision. Cette combinaison nous permet de tirer parti des avantages de WOA, tels que sa capacité à explorer efficacement l'espace de recherche, tout en adaptant l'algorithme à notre contexte spécifique pour rechercher des solutions quasi-optimales qui répondent aux besoins de nos utilisateurs.

3.4 WOA original

3.4.1 Définition

Proposé en 2016 par Mirjalili et Lewis, l'algorithme d'optimisation des baleines (WOA) est une métaheuristique inspirée par le comportement de chasse des baleines à bosse. Ces baleines préfèrent chasser le krill et les petits essaims de poissons en formant un groupe qui recherche sa proie dans un cercle de plus en plus restreint. Lorsqu'elles trouvent une proie, les baleines à bosse utilisent une technique appelée "attaque au filet à bulles" : elles émettent des bulles en spirale autour de la proie, puis remontent vers la surface. Dans le contexte de l'algorithme WOA, le terme "population" est utilisé pour décrire le groupe de baleines, et "composition" pour désigner chaque baleine individuelle. La solution du problème d'optimisation correspond à l'emplacement de la proie. La stratégie est de permettre au meilleur composition de s'approcher le plus près possible de la proie représentant la solution optimale dans un problème d'optimisation [32]. L'encerclement par rétrécissement et la mise à jour en spirale, ont été développées pour modéliser mathématiquement le comportement d'attaque au filet à bulles des baleines à bosse, comme illustré dans la figure 3.2.

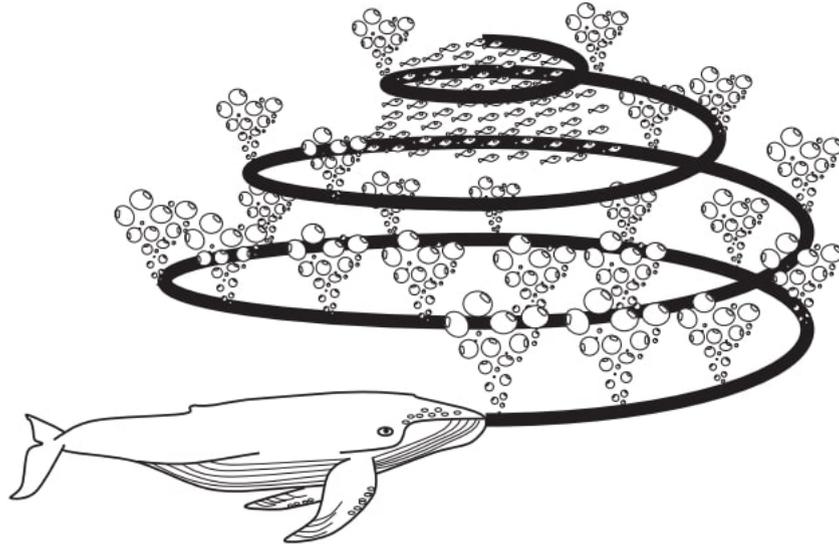


FIGURE 3.2 – La chasse en filet de bulles des baleines à bosses
[33]

3.5 Le fonctionnement de l’approche proposée DT-WOA

DT-WOA est une combinaison d’un modèle d’arbre de décision et d’une version discrétisé de la méta-heuristique WOA pour le traitement du problème de composition de service IoT sensible aux QoS. L’arbre de décision est utilisé dans Dt-WOA pour réduire l’espace de recherche et WOA est utilisé pour rechercher une bonne solution proche de l’optimale qui satisfait les besoins de l’utilisateur en terme de QoS. Dans cette section, nous expliquons la mise en oeuvre de notre approche. Nous présentons les deux étapes de DT-WOA à savoir la classification et la sélection des services IoT en utilisant l’arbre de décision et l’application de l’algorithme WOA discrétisé.

3.5.1 Classification et sélection des services candidats via l’arbre de décision

Nous avons développé un modèle d’arbre de décision pour évaluer chaque service IoT selon son vecteur d’attributs QoS, pour l’affecter à une catégorie de niveau de QoS. A notre connaissance, il n’existe pas de dataset définissant des services IoT. Aussi pour la mise en oeuvre du modèle d’arbre de décision de DT-WOA les deux dataset QWS 1.0 et QWS 2.0 ont été utilisés. QWS 1.0 est utilisé pour l’entraînement supervisé du modèle de l’arbre de décision. Tandis que QWS 2.0 sera utilisé pour l’évaluation de notre approche. Pour la phase d’apprentissage de l’arbre de décision, nous utilisons le QWS 1.0 qui contient les vecteurs QoS des services candidats (vecteurs d’entrées) avec leur catégorie (étiquettes ayant comme valeurs : bronze, argent, platine ou or). Le dataset QWS 1.0 contient 365 services. chaque service est caractérisé par neuf attributs de QoS en plus de deux attributs supplémentaires à savoir : un classement des services basé sur une fonction de pertinence de services (WsRF) et une classification des services en fonction de leurs performances globales.

La normalisation des valeurs des QoS des dataset

Avant de procéder à la classification des services, il est nécessaire de normaliser leurs valeurs de QoS de manière à les ramener dans l'intervalle $[0,1]$ [34]. Pour normaliser les valeurs des attributs de QoS, nous avons utilisé les formules suivantes [11] :

— **Attributs négatifs :**

$$q'_{(i,j)} = \begin{cases} \frac{q_j^{max} - q_{(i,j)}}{q_j^{max} - q_j^{min}} & \text{si } q_j^{max} - q_j^{min} \neq 0 \\ 1 & \text{sinon} \end{cases} \quad (3.1)$$

— **Attributs positifs :**

$$q'_{(i,j)} = \begin{cases} \frac{q_{(i,j)} - q_j^{min}}{q_j^{max} - q_j^{min}} & \text{si } q_j^{max} - q_j^{min} \neq 0 \\ 1 & \text{sinon} \end{cases} \quad (3.2)$$

où $q'_{(i,j)}$ est la valeur normalisée de l'attribut j du service candidat SC_i . Elle est calculée en utilisant la valeur non normalisée $q_{i,j}$ ainsi que les valeurs maximales q_j^{max} et minimales q_j^{min} possibles de l'attribut j dans le dataset.

La classification et sélection

Après avoir préparé et entraîné notre modèle d'arbre de décision, nous l'utilisons pour classer les services du dataset QWS Data 2.0. Les valeurs de QoS de ce dataset ont été normalisées au préalable. Ensuite, une fois que ces services IoT sont classés, nous sélectionnons ceux qui font partie des deux meilleures catégories de services IoT, c'est-à-dire ceux ayant une meilleure note de qualité globale, à savoir les catégories "platine" et "or".

Divisions du dataset

Une fois que les services du dataset QWS 2.0 sont classés, nous sélectionnons les services candidats des catégories "platine" et "or" pour les étapes suivantes, notamment l'application du WOA discrétisé. Après avoir choisi ces services, nous procédons à leur extraction et les répartissons ensuite parmi les services abstraits du plan de composition. Il est important de souligner que chaque service abstrait AS recevra le même nombre de services candidats, noté nSC .

l'algorithme 1 représente les étapes de l'arbre de décision utilisé pour la classification des services.

Algorithm 1: Algorithme de l'arbre de décision

1 Entrées : QwS 1.0, QWS 2.0

2 Sorties : QWS 2.0 normalisé, classé et réparti sur les AS

3 DEBUT

1: Lire le dataset QWS 1.0 ;

2: Normalisation des valeurs du dataset QWS 1.0 ;

3: Faire appel à la fonction `decisionTreeClassifier()` pour l'entraînement supervisé ;

4: Lire le dataset QWS 2.0 ;

5: Normalisation des valeurs du dataset QWS 2.0 ;

6: Utiliser le modèle de l'arbre de décision pour classer les services du dataset QWS 2.0 ;

7: Sélectionner les services qui font partie des deux meilleures catégories, à savoir les catégories "platine" et "or" du QWS 2.0 classifié ;

8: Diviser les services sélectionnés à l'aide de la fonction `diviser_dataset`.

FIN

3.5.2 Application du WOA discrétisé

Dans ce qui suit, nous présentons notre modélisation du problème de composition de services IoT sensibles aux critères de QoS. Une composition abstraite, ou un plan de composition (Workflow), notée SCA, est un enchaînement nAS services abstraits (AS) et est représentée comme suit $SCA = \{AS_1, AS_2, \dots, AS_{nAS}\}$. Chaque AS contient un ensemble de nSC services candidats fonctionnellement équivalent $AS = \{SC_1, SC_2, \dots, SC_{nSC}\}$. Nous représentons une composition par un vecteur de nAS nombre entier comme suit : $comp = \{x_1, x_2, \dots, x_{nAS}\}$ où x_1, x_2, \dots, x_{nAS} représentent les indices des services candidats appartenant à la composition. Dans la suite de cette section nous allons détailler les étapes du WOA discrétisé.

a. Généré aléatoirement les compositons de la population initiale

Dans cette étape, on génère aléatoirement des compositions à travers l'espace de recherche. Les valeurs générées ne doivent pas dépasser le nombre de service candidat dans les AS (c'est à dire les x_i de chaque composition ont des valeurs entre 1 et nSC).

b. Évaluation de la population initiale

Après avoir généré la population initiale, nous procédons au calcul de la fitness de chaque composition en utilisant la fonction d'utilité présentée dans l'équation 2.1. Ensuite, la composition ayant la meilleure valeur de fitness est enregistrée et sauvegardée jusqu'à la fin de l'exécution de DT-WOA. Dans notre cas, nous cherchons à minimiser la valeur de la fitness, donc c'est la composition ayant la plus petite valeur de fitness qui sera retenue.

c. Appliquer les opérations de WOA à la population initiale

Pour chaque composition de la population, on applique les opérations de WOA pour mettre à jour les compositions. Deux opérations sont définies dans WOA, l'encerclement et l'attaque. Noté qu'à la fin de chaque opération une nouvelle composition est générée et ira remplacer l'ancienne dans la population.

L'encerclement

Le WOA commence par identifier et sauvegarder la meilleure composition de la population en calculant et comparant les fitnesses des compositions. Étant donné que la composition optimale n'est pas connue au départ, la valeur fitness de la meilleure composition sauvegardée pour mettre à jours les autres compositions et cela selon l'équation suivante [32] :

$$comp(t+1) = comp^*(t) - \vec{A} \cdot |\vec{C} \cdot comp^*(t) - comp(t)| \quad (3.3)$$

Où $comp^*$ représente la meilleure composition obtenue jusqu'à présent, $comp$ représente la composition actuelle, $comp(t+1)$ représente la nouvelle composition, t représente l'itération courante, \vec{A} et \vec{C} sont des vecteurs de coefficients qui peuvent être calculés comme suit :

$$\vec{A} = 2\vec{a} \cdot \vec{r} - \vec{a} \quad (3.4)$$

$$\vec{C} = \vec{a} \cdot \vec{r} \quad (3.5)$$

Où \vec{a} décroît linéairement de 2 à 0 au cours de la recherche, et \vec{r} est un vecteur aléatoire qui suit une distribution uniforme dans l'intervalle $[0, 1]$. L'équation 3.3 permet à chaque composition de mettre à jour sa position dans le voisinage de la meilleure solution actuelle, mimant ainsi l'encerclement de solutions. L'encerclement par rétrécissement est mis en œuvre en réduisant progressivement la valeur de \vec{a} dans les équations 3.4 et 3.5 au fil des itérations. En limitant la valeur de \vec{A} à l'intervalle $[-1, 1]$, la nouvelle composition peut être définie n'importe où entre sa composition actuelle et la composition ayant la meilleure fitness.

L'attaque au filet à bulles (phases d'exploitation)

Le processus de mise à jour en spirale, repose sur l'utilisation d'une équation en forme de spirale pour imiter le mouvement en hélice des baleines à bosse, comme suit [32] :

$$comp(t+1) = \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l) + comp^*(t) \quad (3.6)$$

Où $\vec{D}' = |comp^*(t) - comp(t)|$ définit la distance entre la i^{me} composition et la solution, b est une constante définissant une spirale logarithmique, et l est un vecteur aléatoire défini selon une distribution uniforme dans l'intervalle $[-1, 1]$.

Puisque les baleines à bosse nagent autour de la proie dans un cercle rétréci tout en suivant un chemin en forme de spirale, l'algorithme WOA suppose qu'il existe une probabilité de 0,5 de choisir l'un ou l'autre de ces processus à chaque itération comme le montre la formule suivante :

$$comp(t+1) = \begin{cases} comp^*(t) - \vec{A} \cdot |\vec{C} \cdot comp^*(t) - comp(t)|, & p < 0.5 \\ \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l) + comp^*(t), & p \geq 0.5 \end{cases} \quad (3.7)$$

Où p est un nombre aléatoire compris entre 0 et 1.

Recherche de solutions (phase d'exploration)

Contrairement à la phase d'exploitation, WOA met à jour la population lors de l'exploration comme suit [32] :

$$comp(t+1) = \overrightarrow{comp_{rand}} - \vec{A} \cdot |\vec{C} \cdot \overrightarrow{comp_{rand}} - comp(t)| \quad (3.8)$$

où $\overrightarrow{comp_{rand}}$ représente une composition choisi au hasard parmi l'ensemble actuel des compositions. La valeur du coefficient A est utilisée pour décider si une composition doit effectuer une recherche aléatoire. Lorsque $|A| \geq 1$, cela signifie que la composition doit faire de l'exploration.

d. Choisir la meilleure solution

À la fin de DT-WOA, l'algorithme retourne la meilleure composition parmi toutes les compositions générées pendant les itérations (cela est possible car à chaque fin d'itération la meilleure composition retrouvée est sauvegardée dans un ensemble). Cela peut être déterminé en comparant les valeurs de fitness des meilleures compositions sauvegardées. En suivant ces étapes, l'algorithme WOA explore l'espace de recherche en générant et en évaluant de nouvelles compositions à chaque itération. Au fil des générations, il tend à converger vers une bonne composition proche de l'optimale. L'algorithme 2 présente les étapes suivi par WOA.

Algorithm 2: Algorithmme de WOA

1 **Entrées :** nAS , nQoS, nSC, niteration, nCOM, poids,

2 **Sortie :** meilleur composition de fitness,

3 **DEBUT**

1: Génération aléatoire de la population initiale ;

2: Évaluation de la population initiale en utilisant l'équation 2.1 ;

3: Sauvgarder la meilleur fitness et sa composition ;

4: **Tantque** t=1 et t < niteration

5: **Pour chaque composition faire :**

6: Mises à jour de a, A, C, I, et p ;

7: **Si** (p < 0.5)

8: **Si** (|A| < 1)

9: Mises à jour de la composition en utilisant l'équation 3.3 ;

10: **Sinon**

11: Mises à jour de la composition en utilisant l'équation 3.8 ;

12: **FinSi**

13: **Sinon**

14: Mises à jour de la composition en utilisant l'équation 3.6 ;

15: **FinSi**

16: **FinPour**

17: Après chaque mises à jour de la population évaluer les nouvelles compositions avec l'équation 2.1 ;

18: Sauvegarder la meilleur fitness et sa composition ;

19: t=t+1 ;

20: **FinTanque**

21: Retourner la meilleur composition et sa fitness.

END

3.6 Scénario d'application de DT-WOA (Etude de cas)

Dans un monde de plus en plus connecté, les maisons intelligentes gagnent en popularité où la technologie est au service de la sécurité et du confort de l'être humain. Nous avons choisit comme scénario pour valider notre approche la surveillance d'une maison intelligente. Cette maison est équipée de détecteurs de mouvements et de caméras de surveillance rota-

tives. L'objectif est de détecter toute tentative d'intrusion et d'alerter le propriétaire en cas de danger réel. Pour y parvenir, il faudra combiner quatre services IoT comme illustré dans la figure 3.3. Ces services sont essentiels pour garantir le bon fonctionnement de notre système de surveillance. Chacun d'eux a une fonctionnalité spécifique, allant de la détection des mouvements à la notification du propriétaire en cas d'intrusion. Cette coordination efficace assure la tranquillité d'esprit des résidents. Nous présentons dans ce qui suit la description des services abstrait de notre plan de composition ainsi que le rôle de chacun d'entre eux.

1. **Service de Détection de Mouvements (SDM)** Ce service gère les capteurs de mouvements installés autour de la maison. Lorsqu'un mouvement est détecté, il déclenche une action pour activer les caméras de sécurité.
2. **Service d'Activation des Caméras (SAC)** Ce service est chargé d'activer la caméra de sécurité la plus proche dès qu'un mouvement est détecté par le SDM. Une fois activée, la caméra rotative commence à enregistrer la vidéo de la zone concernée.
3. **Service d'Analyse Vidéo en Temps Réel (SAVTR)** Ce service prend la vidéo enregistrée par les caméras et analyse en temps réel son contenu pour déterminer s'il s'agit d'une intrusion réelle. Il peut identifier si la présence détectée est due à une personne, un animal ou une autre cause.
4. **Service de Notification au Propriétaire (SNP)** Lorsque le SAVTR détecte une possible intrusion humaine, il envoie une notification au propriétaire de la maison. Cette notification inclut un lien vers la vidéo en cours d'enregistrement pour permettre au propriétaire de visualiser la situation en direct. Le SNP permet également au propriétaire de décider s'il doit appeler le service de sécurité en cas de confirmation de tentative d'intrusion.

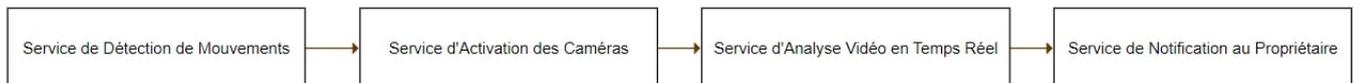


FIGURE 3.3 – Plan de composition associé au scénario de DT-WOA

Ces services fonctionnent de manière coordonnée pour garantir la sécurité de la maison intelligente, en minimisant les fausses alertes et en permettant au propriétaire de prendre des mesures rapides en cas d'intrusion réelle.

Rappelons que chaque service abstrait du plan de composition définit dans la figure 3.3 peut être exécuté par plusieurs services IoT candidats ayant la même fonctionnalité mais des valeurs d'attributs QoS différentes. Dans le tableau 3.1 nous présentons pour chaque services abstraits du scénario un exemple de trois services IoT candidats réels. Cela fait 3^3 compositions de services IoT possible. L'objectif de l'approche DT-WOA proposée est de sélectionner la meilleure combinaison de services répondant aux exigences de qualité de service de l'utilisateur.

Service abstrait	service candidat	Fournisseur
Service de Détection de Mouvements	CS_1^1	ADT
	CS_2^1	SimpliSafe
	CS_3^1	Vivint

Service d'Activation des Caméras	CS_1^2 CS_2^2 CS_3^2	Arlo Nest Ring
Service d'Analyse Vidéo en Temps Réel	CS_1^3 CS_2^3 CS_3^3	IBM Watson Visual Recognition Deep Sentinel Avigilon
Service de Notification au Propriétaire	CS_1^4 CS_2^4 CS_3^4	Verisure Frontpoint Xfinity Home

TABLE 3.1 – Services candidats disponibles pour chaque service abstrait dans le scénario de composition

3.7 Simulation

3.7.1 Environnement, outils et bibliothèques utilisés

L'implémentation et l'évaluation de l'approche proposée ont été réalisées sur un ordinateur portable sous Windows 10 Professionnel caractérisé par un Processeur Intel(R) Core(TM) i3-6006U CPU @ 2.00GHz 1.99 GHz équipé d'une RAM de 4 Go.

JupyterLab

JupyterLab est le dernier environnement de développement interactif basé sur le Web pour les blocs-notes, le code et les données. Son interface flexible permet aux utilisateurs de configurer et d'organiser des flux de travail en science des données, en informatique scientifique, en journalisme informatique et en apprentissage automatique. Une conception modulaire invite les extensions à étendre et enrichir les fonctionnalités[35].

Jupyter Notebook

Jupyter Notebook est l'application Web originale pour créer et partager des documents informatiques. Il offre une expérience simple, rationalisée et centrée sur les documents[35].

Python

Python est le langage de programmation informatique le plus populaire et le plus utilisé, notamment dans le domaine de la Data Science et du Machine Learning. De plus, Python est un langage multiplateforme qui fonctionne sur divers systèmes d'exploitation, tels que Windows, macOS et Linux, ce qui en fait un choix idéal pour les développeurs travaillant sur différents environnements[36].

Pandas

La bibliothèque logicielle open-source Pandas est spécifiquement conçue pour la manipulation et l'analyse de données en langage Python. Elle est à la fois performante, flexible et

simple d'utilisation.

Grâce à Pandas, le langage Python permet enfin de charger, d'aligner, de manipuler ou encore de fusionner des données. Les performances sont particulièrement impressionnantes quand le code source back-end est écrit en C ou en Python[37].

Scikit-Learn

C'est une librairie Python qui donne accès à des versions efficaces d'un grand nombre d'algorithmes courants. Elle offre également une API propre et uniformisée. Par conséquent, un des gros avantages de Scikit-Learn est qu'une fois que vous avez compris l'utilisation et la syntaxe de base de Scikit-Learn pour un type de modèle, le passage à un nouveau modèle ou algorithme est très simple. La librairie ne permet pas seulement de faire de la modélisation, elle peut assurer également des étapes de preprocessing[38].

3.7.2 Dataset

Pour tester notre approche nous allons utiliser le dataset QWS 2.0 qui comprend un ensemble de 2507 services et leurs mesures de qualité de service qui ont été effectuées au cours de l'année 2008. Chaque ligne de cet ensemble de données représente un service candidats et ses neuf mesures de QoS correspondantes (séparées par des virgules). Les valeurs QoS représentent les moyennes des mesures collectées au cours de cette période. Les deux derniers paramètres représentent le nom du service et la référence au document WSDL[39]. La liste des critères de QoS pris en considération dans le QWS 2.0 sont présentés dans la figure 3.4.

ID	Parameter Name	Description	Units
1	Response Time	Time taken to send a request and receive a response	ms
2	Availability	Number of successful invocations/total invocations	%
3	Throughput	Total Number of invocations for a given period of time	invokes/second
4	Successability	Number of response / number of request messages	%
5	Reliability	Ratio of the number of error messages to total messages	%
6	Compliance	The extent to which a WSDL document follows WSDL specification	%
7	Best Practices	The extent to which a Web service follows WS-I Basic Profile	%
8	Latency	Time taken for the server to process a given request	ms
9	Documentation	Measure of documentation (i.e. description tags) in WSDL	%
10	WsRF	Web Service Relevancy Function: a rank for Web Service Quality	%
11	Service Classification	Levels representing service offering qualities (1 through 4)	Classifier
12	Service Name	Name of the Web service	-
13	WSDL Address	Location of the Web Service Definition Language (WSDL) file on the Web	-

FIGURE 3.4 – Description des critères de QoS du dataset QWS [39]

Pour la phase d'évaluation de notre approche nous avons considéré les 9 premiers attributs de QoS du dataset 2.0.

3.7.3 Pré-traitement des données

Le Pré-traitement des données est une technique d'exploration de données qui consiste à convertir des données brutes dans un format compréhensible. Les données du monde réel sont souvent incomplètes, incohérentes et/ou manquent certains comportements ou tendances, et sont sujettes à de nombreuses erreurs. Le prétraitement des données est une méthode efficace pour résoudre ces problèmes[40].

Pendant la phase de simulation de notre approche, nous avons utilisé Jupyter pour créer le projet et Python comme langage de programmation.

3.7.4 Normalisation des données du dataset

Avant de classer les services, Nous devons normaliser les valeurs de QoS des services du dataset QWS afin qu'elles appartiennent à l'intervalle [0,1] en utilisant la fonction MinMaxScaler.

La figure 3.5 montre le dataset QWS 2.0 après qu'il a été lu grâce à la bibliothèque Pandas, et avant la normalisation.

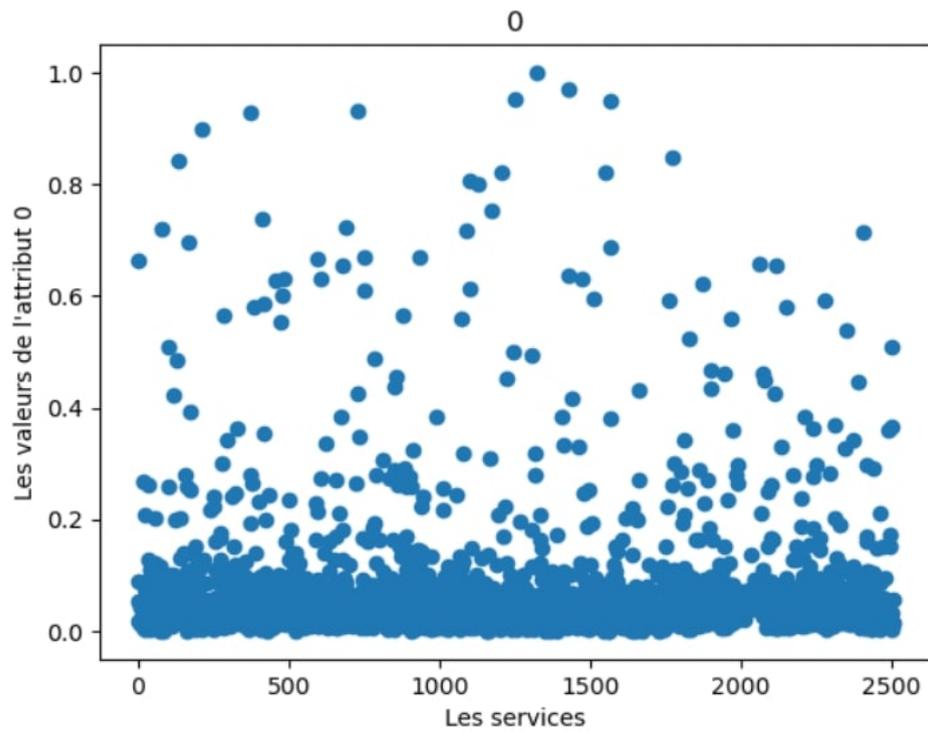


FIGURE 3.7 – Les valeurs de l'attribut temps de réponse avant la classification

La figure 3.8 présente les valeurs de l'attribut QoS succès des services du dataset QWS 2.0 avant la classification.

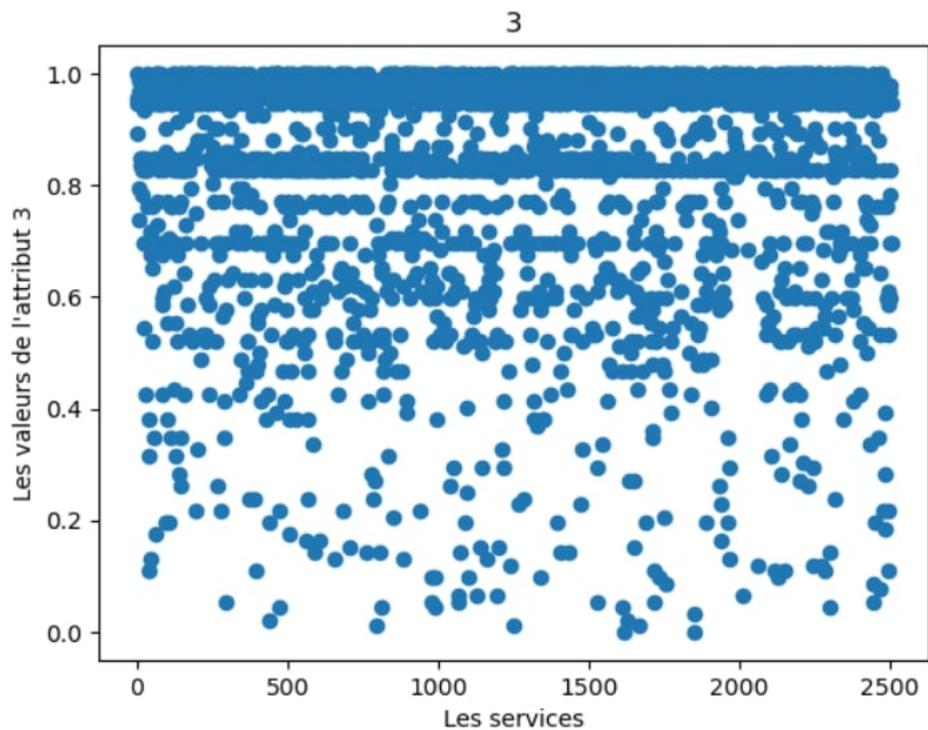


FIGURE 3.8 – Les valeurs de l'attribut Succès avant la classification

La figure 3.9 montre le QWS 2.0 après la classification

	0	1	2	3	4	5	6	\
0	0.053658	0.881720	0.162791	0.891304	0.714286	0.671642	0.666667	
1	0.089851	0.838710	0.369767	0.945652	0.714286	1.000000	0.755556	
2	0.663157	0.881720	0.030233	0.956522	0.714286	0.671642	0.666667	
3	0.018004	0.978495	0.276744	1.000000	0.607143	0.671642	0.711111	
4	0.014134	0.860215	0.041860	0.945652	0.714286	0.835821	0.266667	
...	
2500	0.033073	0.924731	0.053488	0.978261	0.714286	1.000000	0.755556	
2501	0.003871	0.967742	0.260465	0.967391	0.892857	0.671642	0.911111	
2502	0.011495	0.784946	0.046512	0.782609	0.607143	0.671642	0.711111	
2503	0.014083	0.849462	0.027907	0.945652	0.839286	0.671642	0.822222	
2504	0.056434	0.698925	0.367442	0.695652	0.839286	0.835821	0.822222	
	7	8	target					
0	0.045289	0.322917	3					
1	0.000181	0.010417	3					
2	0.000568	0.989583	2					
3	0.005439	0.916667	2					
4	0.014029	0.958333	3					
...					
2500	0.001727	0.416667	2					
2501	0.001671	0.020833	2					
2502	0.000838	0.614583	2					
2503	0.000242	0.989583	4					
2504	0.001268	0.072917	2					

[2505 rows x 10 columns]

FIGURE 3.9 – QWS 2.0 après la classification

Rappelons que le dataset QWS 1.0 est utilisé pour l’entraînement supervisé du modèle d’arbre de décision grâce à la bibliothèque Scikit-Learn.

La figure 3.7 présente les valeurs de l’attribut QoS temps de réponse des services du dataset QWS 2.0 après la classification.

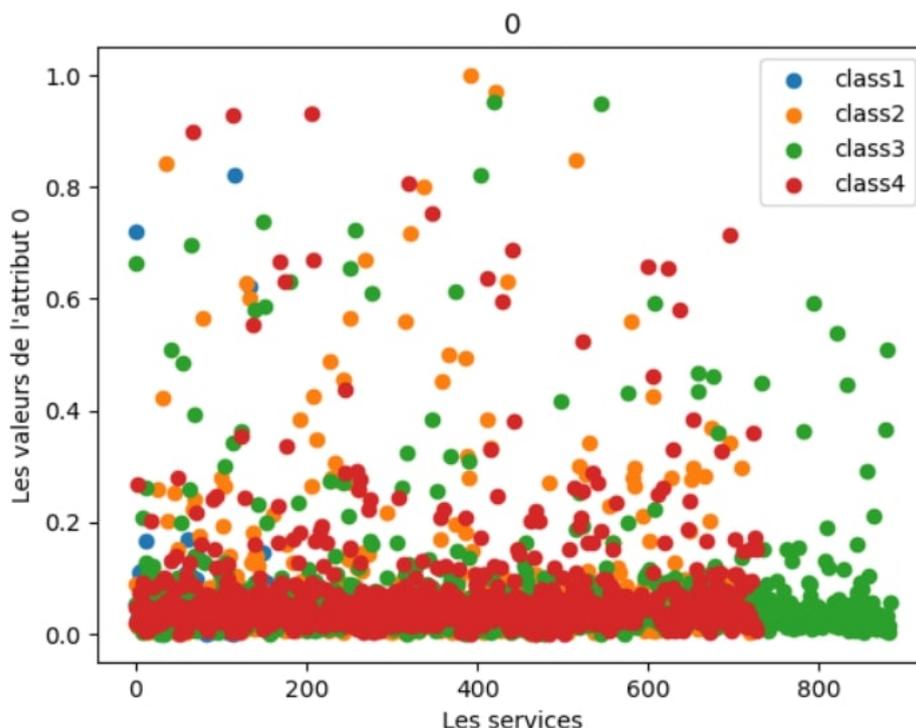


FIGURE 3.10 – les valeurs de l’attribut Temps de réponse après la classification

La figure 3.7 présente les valeurs de l'attribut QoS succès des services du dataset QWS 2.0 après la classification.

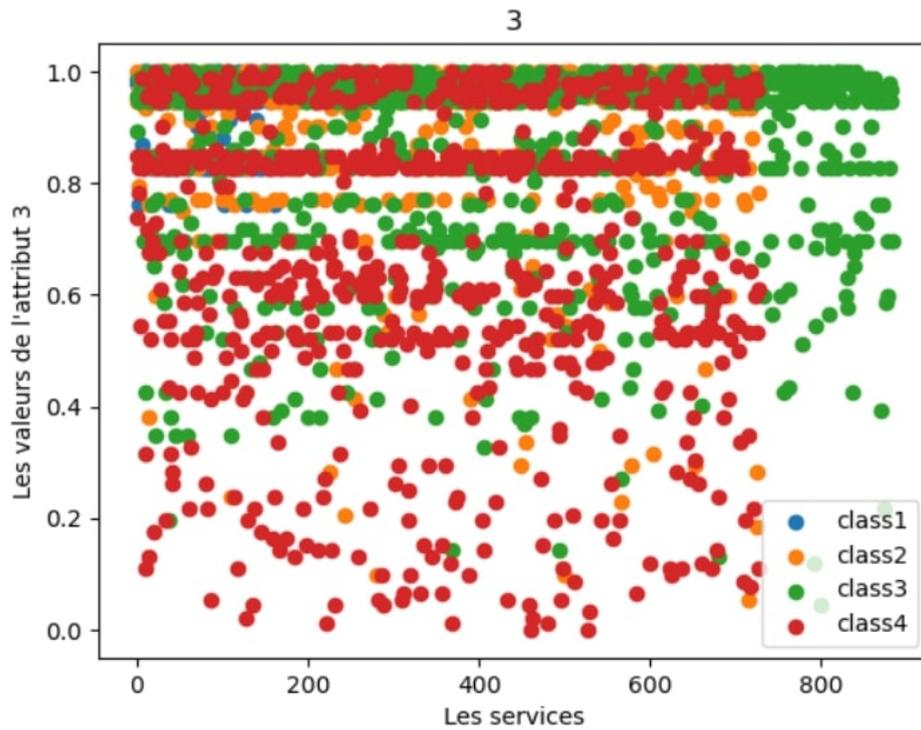


FIGURE 3.11 – les valeurs de l'attribut Succès après la classification

La figure 3.9 montre le QWS 2.0 après la selection

	0	1	2	3	4	5	6	\	
2	0.663157	0.881720	0.030233	0.956522	0.714286	0.671642	0.666667		
3	0.018004	0.978495	0.276744	1.000000	0.607143	0.671642	0.711111		
5	0.014249	0.784946	0.037209	0.793478	0.607143	0.671642	0.711111		
8	0.013249	0.903226	0.353488	0.967391	0.607143	0.671642	0.711111		
9	0.011382	0.956989	0.311628	0.989130	0.607143	0.835821	0.177778		
...	
2495	0.032171	0.860215	0.218605	0.945652	0.357143	1.000000	0.466667		
2500	0.033073	0.924731	0.053488	0.978261	0.714286	1.000000	0.755556		
2501	0.003871	0.967742	0.260465	0.967391	0.892857	0.671642	0.911111		
2502	0.011495	0.784946	0.046512	0.782609	0.607143	0.671642	0.711111		
2504	0.056434	0.698925	0.367442	0.695652	0.839286	0.835821	0.822222		
	7	8	target						
2	0.000568	0.989583	2						
3	0.005439	0.916667	2						
5	0.004338	0.625000	2						
8	0.000164	0.937500	1						
9	0.010002	0.958333	2						
...						
2495	0.009147	0.885417	2						
2500	0.001727	0.416667	2						
2501	0.001671	0.020833	2						
2502	0.000838	0.614583	2						
2504	0.001268	0.072917	2						

[1023 rows x 10 columns]

FIGURE 3.12 – QWS 2.0 après la selection

3.7.6 Recherche d'une composition sous optimale via WOA discrétisé

Après avoir appliqué le modèle d'arbre de décision, nous passons à la seconde étape de notre approche qui est l'application du WOA discrétisé. Dans ce qui suit, nous présentons le tableau 3.2 récapitulant les valeurs de fitness obtenues par DT-WOA en comparaison avec WOA.

CS	20	40	60	80	100
DT-WOA	0.173	0.162	0.132	0.132	0.132
WOA	0.182	0.182	0.182	0.174	0.174

TABLE 3.2 – Valeurs de fitness de DT-WOA et de WOA

Le graphique représenté dans la figure 3.13 montre que DT-WOA offre une meilleure composition que WOA en terme de fitness. Cela nous confirme que la combinaison de WOA avec notre modèle d'arbre de décision permet d'améliorer les performances de WOA.

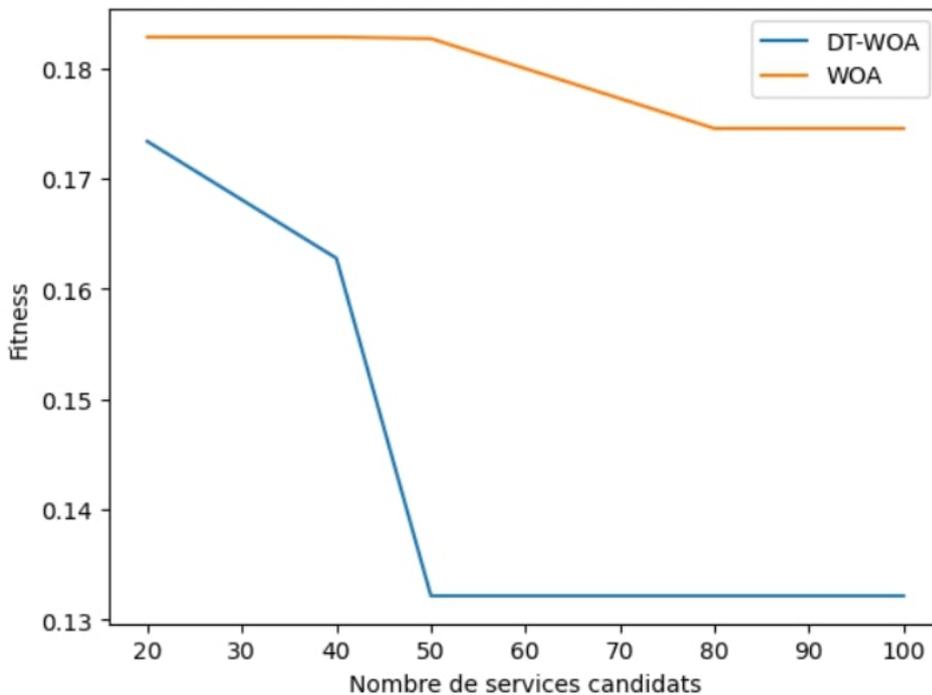


FIGURE 3.13 – Évolution des valeurs de fitness de DT-WOA et WOA originale

3.8 Conclusion

Ce chapitre a été consacré à la présentation de notre approche DT-WOA, de l'environnement de développement utilisé, notamment les différents outils et bibliothèques. Nous avons également évalué les performances de DT-WOA en le comparant au WOA sans réduction de l'espace de recherche. Comme nous l'avons montré dans ce chapitre, notre approche obtient des résultats satisfaisants en termes de valeurs de fitness QoS.

Conclusion générale et perspectives

Dans ce travail, nous avons proposé une nouvelle approche qui combine un arbre de décision avec une version discrétisée de l'algorithme WOA que nous avons nommée DT-WOA. DT-WOA est proposé pour la résolution du problème de sélection de composition avec prise en compte des QoS et des préférences de l'utilisateur. l'arbre de décision permet de classer les services candidats présents dans le dataset en quatre catégories : Or, platine, bronze et argent. Les meilleurs services en terme de valeurs de QoS appartiennent à la catégorie Or et juste après on a ceux de la catégorie platine. Pour cela, nous réduisons l'espace de recherche en ne gardant que les services or et platine. Cette réduction de l'espace de recherche permet d'optimiser le temps de réponse de DT-WOA et garantir que la qualité de la composition retournée sera proche de l'optimale.

Pour valider l'efficacité de notre approche, nous l'avons appliqué à un cas d'étude réel qui est la surveillance d'une maison intelligente équipée de capteur de mouvement et de caméra de surveillance rotative. Les résultats d'évaluation de DT-WOA en comparaison avec WOA original sans arbre de décision prouve que notre approche offre de meilleurs résultats en termes de fitness.

Comme perspective de notre approche, nous citons :

- Envisager de développer un ensemble de données (dataset) spécifique pour les services IoT.
- Explorer de nouvelles métaheuristiques.
- Employer des techniques du Deep Learning.

Bibliographie

- [1] Osama Alsaryrah, Ibrahim Mashal, and Tein-Yaw Chung. Bi-objective optimization for energy aware internet of things service composition. *IEEE Access*, 6 :26809–26819, 2018.
- [2] Oussama Afouf and Sofia Kouah. Développement d’un système d’iot (internet of things) dans le cadre de smart university. 2020.
- [3] NECIB HANA. A combinatorial auction-based approach for iot service composition. 2021.
- [4] Fatma Merabet. Solutions de sécurité pour l’internet des objets dans le cadre de l’assistance à l’autonomie à domicile. PhD thesis, Université de Limoges; Université Mouloud Mammeri (Tizi-Ouzou, Algérie), 2021.
- [5] Khaldoun Al Agha and Daniel Etiemble. Introduction à l’Internet des Objets. *Techniques de l’Ingenieur*, October 2018.
- [6] Philippe de Matteis. Comment se compose un système iot ? *Connectwave*.
- [7] Yasmine Harbi, Zibouda Aliouat, Allaoua Refoufi, and Saad Harous. Recent security trends in internet of things : A comprehensive survey. *IEEE Access*, 9 :113292–113314, 2021.
- [8] Somayyeh Zakerabasali and Seyed Mohammad Ayyoubzadeh. Internet of things and healthcare system : A systematic review of ethical issues. *Health science reports*, 2022.
- [9] Mohamed Abdel-Basset, Gunasekaran Manogaran, Mai Mohamed, and Ehab Rushdy. Internet of things in smart education environment : Supportive framework in the decision-making process. *Concurrency and Computation : Practice and Experience*, 31 :e4515, 05 2018.
- [10] Y KAROUANI and H ZIYATI. Big data dans l’architecture de l’iot pour les villes intelligentes.
- [11] Yazid Gherbal. 2021-2022.
- [12] Mme SADOUKI Née CHIBANI Samia. Métaheuristiques bio-inspirées pour la sélection basée sur la QoS dans une composition de service web. PhD thesis, Université A.MIRA-BEJAIA, 2017/2018.
- [13] Ibrahim et Chung Tein-Yaw Alsaryrah, Oussama et Mashal. Optimisation bi-objectif pour la composition d’un service internet des objets soucieux de l’énergie. *Accès IEEE*, 6.
- [14] Idir Aoudia. Composition adaptative de services pour l’Internet des objets. PhD thesis, Université de mohamed kheider biskra, 2022.
- [15] Hioual Ouassila Boucetti Rabah, Hemam Sofiane Mounine. f King Saud University - Computer and Information Sciences, 2022.

- [16] Ali Yachir. Composition dynamique de services sensibles au contexte dans les systèmes intelligents ambiants. PhD thesis, Paris Est, 2014.
- [17] Algorithme de sélection de services méta-heuristiques prenant en compte qos et mesure d'échelle de likert pour l'environnement iot. Journal international des tendances et technologies informatiques, 8 :1–8.
- [18] Aurélien Géron. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow. " O'Reilly Media, Inc.", 2022.
- [19] Imen Charfi. Détection automatique de chutes de personnes basée sur des descripteurs spatio-tempo. Theses, Université de Bourgogne, October 2013.
- [20] Hafidi Imad. Méta heuristique. 2019.
- [21] Nima Jafari Sefati, Seyedsalar et Navimipour. Un mécanisme de composition de service sensible au qos dans l'internet des objets utilisant un algorithme d'optimisation basé sur un modèle de markov caché. IEEE Internet of Things Journal, 8 :15620–15627.
- [22] Ao et Yang Mingzhe et Sun Lei et Hsu Ching-Hsien et Yang Fangchun Wang, Shang-guang et Zhou. Composition des services dans les systèmes cyber-physiques-sociaux. Transactions IEEE sur les sujets émergents en informatique, 8 :82–91.
- [23] Une méthodologie de sélection partielle pour une composition efficace de services sensibles à la qualité de service. Transactions IEEE sur l'informatique de services, 8 :384–397.
- [24] Aitor Urbieto, Alejandra González-Beltrán, S Ben Mokhtar, M Anwar Hossain, and Licia Capra. Adaptive and context-aware service composition for iot-based smart cities. Future Generation Computer Systems, 76 :262–274, 2017.
- [25] Meysam Karimi and Seyed Morteza Babamir. Qos-aware web service composition using gray wolf optimizer. International Journal of Information and Communication Technology Research, 9(1) :9–16, 2017.
- [26] Aghabi Abosaif and Salah Elrofai. An efficient qos-aware services selection in iot using a reputation improved- social spider optimization algorithm, 06 2020.
- [27] Une nouvelle méthode de composition de service de fabrication sensible à la qos floue utilisant un algorithme étendu de pollinisation des fleurs. Journal de la fabrication intelligente, 30 :2069–2083.
- [28] Une approche de la composition optimale des services sensibles aux ressources et à la qos dans le grand service et l'internet des objets. Accès IEEE, 6 :39895–39906.
- [29] Gestion stratégique de l'énergie verte pour un service de composition de qualité dans l'environnement de l'internet des objets. Complexité, 2020 :1–10.
- [30] Alain Girard. Exploration d'un algorithme génétique et d'un arbre de décision à des fins de catégorisation. PhD thesis, Université du Québec à Trois-Rivières, 2007.
- [31] Faïza BOUDJEMIL Wissam AZNI. Classification des informations contextuelles dans un environnement intelligent ubiquitaire. 2014.
- [32] Hui Li, Peng Zou, Zhiguo Huang, Chenbo Zeng, and Xiao Liu. Multimodal optimization using whale optimization algorithm enhanced with local search and niching technique. Mathematical Biosciences and Engineering, 17(1) :1–27, 2020.
- [33] Seyedali Mirjalili and Andrew Lewis. The whale optimization algorithm. Advances in engineering software, 95 :51–67, 2016.
- [34] Rizzi Meroua, Maazaoui Soumia, and Zertal Soumia. Une nouvelle approche basée deep learning pour la classification des services cloud. 2021.

- [35] Project jupyter | home. <https://jupyter.org/>. Consulté le 12/09/2023.
- [36] python. <https://datascientest.com/python-tout-savoir>. Consulté le 12/09/2023.
- [37] pandas. <https://datascientest.com/pandas-python-data-science>. Consulté le 12/09/2023.
- [38] scikit-learn. <https://datascientest.com/tout-savoir-sur-scikit-learn>. Consulté le 12/09/2023.
- [39] l'ensemble de données qws. <https://qwsdata.github.io/#/>. Consulté le 12/09/2023.
- [40] Prétraitement des données dans l'apprentissage automatique. <https://blog.arcoptimizer.com/pretraitement-des-donnees-dans-lapprentissage-automatique/>. Consulté le 12/09/2023.

Résumé

L'Internet des objets (IoT) est une technologie révolutionnaire qui promet de connecter des milliards d'objets allant des appareils domestiques aux infrastructures urbaines, créant ainsi un réseau interconnecté. La composition de services IoT est cruciale car les utilisateurs ont des besoins variés et complexe qui ne peuvent être satisfait par un seul service IoT. C'est pourquoi nous proposons une approche de composition de services IoT prenant en compte les exigences de qualité de service noté QoS. Cette approche, nommée DT-WOA, combine un arbre de décision avec une version discrétisée de l'algorithme WOA que nous avons développée. Cette approche permet de rechercher une solution quasi optimale en choisissant la meilleure composition de services IoT afin de répondre aux exigences des utilisateurs.

Mot clés : IoT, composition de services, arbre de décision, algorithme WOA, métaheuristique, QoS.

Abstract

The Internet of Things (IoT) is a revolutionary technology that promises to connect billions of objects, from household devices to urban infrastructures, creating an interconnected network. IoT service composition is crucial because users have diverse and complex needs which a single IoT service often does not suffice to fulfill them. That's why we propose an approach to IoT service composition that takes into account Quality of Service noted QoS requirements. This approach called DT-WOA, combines a decision tree with a discrete version of the WOA algorithm that we have developed. This approach allows us to search for a near-optimal solution by selecting the best composition of IoT services to meet user requirements.

Key words : IoT, service composition, decision tree, WOA algorithm, metaheuristic, QoS.