ROYAUME DU MAROC
Université Mohammed V - RABAT -
جامعة محمد الخامس – الرباط –
Faculté des sciences
كلية العلوم
CENTRE D'ÉTUDES DOCTORALES - SCIENCES ET TECHNOLOGIES

N° d'ordre : 3958

THÈSE

en vue de l'obtention du : **DOCTORAT**

Centre de Recherche : Sciences et Technologies
Structure de Recherche : Intelligent Processing & Security of Systems
Discipline : INFORMATIQUE
Spécialité : Intelligence Artificielle et cloud computing

Présentée et Soutenue le : 20 / 07 / 2024

par :

Aristide NDAYIKENGURUKIYE

Modélisation et Optimisation des ressources virtuelles dans un environnement Cloud

Devant le JURY :

Rachid OULAD HAJ THAMI	PES	École Nationale Supérieure d'Informatique et d'Analyse des Systèmes, Université Mohammed V	Président
El Habib BENLAHMAR	PES	Faculté des sciences de Casablanca , Université Hassan II	Examinateur/Rapporteur
Mostafa EL MALLAHI	PH	École Normale Supérieure de Fès, Université Sidi Mohammed Ben Abdellah	Examinateur/Rapporteur
Ali OUACHA	PH	Faculté des Sciences de Rabat , Université Mohammed V	Examinateur/Rapporteur
Hicham GEUDDAH	PH	École Normale Supérieure de Rabat , Université Mohammed V	Examinateur/Rapporteur
Hassan ECHOUKAIRI	PH	Faculté des Sciences de Rabat , Université Mohammed V	Examinateur/Rapporteur
Jérémie NDIKUMAGENGE	PES	Faculté des Sciences de l'Ingénieur de Bujumbura , Université du Burundi	Examinateur
Abderrahmane EZ-ZAHOUT	PH	Faculté des Sciences de Rabat , Université Mohammed V	Directeur de thèse

Année Universitaire : 2023 - 24

À ma tante Gertrude HATUNGIMANA, partie si tôt, dont les conseils indéfectibles m'ont toujours accompagné jusqu'à la réalisation de cette thèse.

À mon fils, Dony Brayan Yanis NGENDANZI, parti si tôt, dont la mémoire reste vivante et inspirante à chaque pas de ce chemin.

Dédicace

A mes parents,

A ma sœur Nina,

A mes frères Enée et Juste Divin,

A ma femme Raissa P.

A ma belle famille

Remerciements

Je tiens à exprimer ma profonde reconnaissance envers le Laboratoire IPSS pour son accueil chaleureux et son soutien tout au long de ma thèse. Je suis reconnaissant pour l'environnement de travail stimulant et les ressources mises à ma disposition, qui ont grandement contribué à la réussite de mon projet de recherche.

Je tiens tout d'abord à exprimer ma profonde gratitude envers mon encadrant principal, Professeur Monsieur **Abderrahmane EZ-ZAHOUT**, pour avoir généreusement consacré son temps et son expertise à superviser mon travail depuis ses prémices. Sa présence et ses précieux conseils ont été essentiels pour façonner mes idées de recherche. Je souhaite également adresser mes remerciements les plus sincères au Professeur Madame **Fouzia OMARY**, pour m'avoir accueilli au sein de son équipe de recherche (IPSS). Ses conseils éclairés et son soutien indéfectible ont été d'une importance capitale dans la réalisation de ce travail.

Je tiens à exprimer ma profonde gratitude envers le président du jury le Professeur Monsieur **Rachid OULAD HAJ THAMI** de l' École Nationale Supérieure d'Informatique et d'Analyse des Systèmes de l'Université Mohammed V pour son temps, son expertise et son dévouement. Merci infiniment pour vos précieuses recommandations et votre bienveillance tout au long de cette évaluation.

Je tiens à exprimer ma profonde gratitude aux rapporteurs du jury pour leur engagement et leurs précieuses contributions à l'évaluation de ma thèse. Mes sincères remerciements vont au Professeur Monsieur **El Habib BENLAHMAR** de la Faculté des sciences de Casablanca , Université Hassan II, au Professeur Monsieur **Mostafa EL MALLAHI** de l'École Normale Supérieure de l'Université Sidi Mohammed Ben Abdellah de Fès, au Professeur Monsieur **Ali OUACHA** de la Faculté des Sciences de Rabat de l'Université Mohammed V, au Professeur Monsieur **Hicham GEUDDAH** de la Faculté des Sciences de Rabat de l'Université Mohammed V et au Professeur Monsieur **Hassan ECHOUKAIRI** de la Faculté des Sciences de Rabat de l'Université Mohammed V. Vos remarques pertinentes, vos critiques constructives et vos conseils éclairés ont grandement enrichi ce travail et ont été essentiels à son aboutissement. Votre expertise, votre rigueur et votre bienveillance ont été des sources de motivation et d'inspiration. Merci infiniment pour votre temps, votre implication et votre soutien indéfectible.

Je remercie tout particulièrement le Professeur Monsieur **Jérémie NDIKUMAGENGE** de la Faculté de Sciences de l'Ingénieur de l'Univérsité du Burundi, qui m'a fait l'honneur d'être examinateur de cette thèse et m'a consacré un temps précieux pour l'examiner.

À ma famille, je suis infiniment reconnaissant pour leur soutien inconditionnel tout au long de ce parcours. A mes parents, à ma sœur Nina Laetitia NSHIMIRIMANA et mes frères Enée NGENDANZI et Juste Divin ISHIMWE pour leurs encouragements constants et leur compréhension face aux exigences de ce projet ont été une source inestimable de motivation. En particulier, je tiens à remercier ma chère épouse Raïssa Prospérine IRAKOZE, dont le soutien et la compréhension ont été mes piliers dans les moments de doute et de difficulté.

Je souhaite exprimer ma reconnaissance à Monsieur Eric NIYUKURI, et Monsieur Eric MUHETO, ainsi qu'à tous ceux qui ont été présents pour moi lors de ce périple académique. A mon père Monsieur Fulgence NGENDANZI, Monsieur Jean Bosco IGIRUKWISHAKA et Madame Lina Inès NEZERWE pour le temps et l'attention qu'ils ont consacrés à la lecture et à la correction de ma thèse. A Madame Nelly DUSABE et Madame KWIZERA Amal Catherine pour le temps que vous avez consacrés à la lecture et à la correction des articles.

Je souhaite exprimer ma profonde gratitude envers la famille de l'Ingénieur MBONIHANKUYE Aloys pour leur soutien indéfectible tout au long de mes recherches. J'adresse également mes sincères remerciements à Monsieur Jean Bosco NDAYIZEYE, à la famille de Monsieur Olivier NAHAYO, ainsi qu'à Madame Marie Chantal NIYONKURU pour leur aide précieuse.

Enfin, un remerciement spécial au Professeur Madame Rachel AKIMANA pour son appui constant et ses précieux conseils tout au long de ce travail. Son expertise et sa bienveillance ont été des atouts inestimables dans la réussite de beaucoup de mes projets ainsi qu'à ma collegue Madame Rim DOUKHA pour la bonne collaboration .

À chacun d'entre vous, je suis profondément reconnaissant pour votre soutien inébranlable, votre patience et votre encouragement tout au long de ce voyage. Cette thèse est le fruit de nos efforts collectifs, et je vous en suis infiniment reconnaissant.

Résumé

La gestion des ressources virtuelles dans le cloud computing est cruciale pour garantir une utilisation efficace des ressources. La modélisation consiste à représenter les composants sous forme de modèles mathématiques pour analyser le comportement du système et prévoir l'utilisation des ressources. Ces modèles permettent également d'évaluer les compromis entre coût et performance. L'optimisation des ressources vise à allouer les ressources en minimisant les coûts tout en répondant aux besoins des applications. Les approches incluent des algorithmes heuristiques, rapides mais parfois imprécis, et des méthodes d'optimisation mathématique, plus précises mais complexes.

Des algorithmes comme Grey Wolf Optimization (GWO) et Seagull Optimization (MOSOAVMP) ont été développés pour améliorer le placement des machines virtuelles (VMP). GWO permet de réduire la consommation d'énergie de 20,99% et les gaspillages de 1,80%. MOSOAVMP, en optimisant l'utilisation des ressources et en minimisant les violations des accords de niveau de service, améliore l'efficacité des centres de données, réduisant les gaspillages et augmentant l'utilisation du CPU, de la mémoire et de la bande passante. Ces approches garantissent une meilleure gestion des ressources dans le cloud, tout en réduisant les coûts.

En parallèle, les systèmes de recommandation, utilisant des algorithmes comme K-NN, contribuent à divers secteurs, y compris la gestion électorale, en proposant des suggestions pertinentes en exploitant les données massives et l'apprentissage automatique.

Mots clé : Cloud Computing, Optimisation, Système de Recommandation, Métaheuristiques et Placement des Machines Virtuelles

Abstract

Virtual resource management in cloud computing is crucial to ensure efficient resource utilization. Modeling involves representing components as mathematical models to analyze system behavior and predict resource utilization. These models can also be used to assess trade-offs between cost and performance. Resource optimization aims to allocate resources while minimizing costs and meeting application needs. Approaches include heuristic algorithms, which are fast but sometimes imprecise, and mathematical optimization methods, which are more precise but complex.

Algorithms such as Grey Wolf Optimization (GWO) and Seagull Optimization (MOSOAVMP) have been developed to improve the placement of virtual machines (VMPs). GWO reduces energy consumption by 20.99% and wastage by 1.80%. MOSOAVMP, by optimizing resource utilization and minimizing SLA violations, improves data center efficiency, reduces waste, and increases CPU, memory, and bandwidth utilization. These approaches guarantee better resource management in the cloud while reducing costs.

At the same time, recommendation systems, using algorithms like K-NN, contribute to various sectors, including election management, by offering relevant suggestions by exploiting massive data and machine learning.

Keywords: Cloud Computing, Optimization, Recommendation System, Metaheuristics and Virtual Machine Placement

Liste des Figures

Liste des Tableaux

Sigles et Abbréviations

ARP	Address Resolution Protocol
ARP	Address Resolution Protocol
BGP	Border Gateway Protocol
CAPEX	CAPital EXpenditures
CC	Cloud Computing
DDoS	Distributed Denial-of-Service
DoS	Denial-of-Service
EIGRP	Enhanced Interior Gateway Routing Protocol
ERP	Enterprise Ressource Planning
GPRS	General Packet Radio Service
IBM	International Business Machines Corporation
ICACIN	International Conference of Advanced Computing and informatics
ICMDS	International Conference on Mathematics and Data Science
IDS	Intrusion Detection System
IoT	Internet Of Things
IPS	Intrusion Prevention System
ISO	International Organization for Standardization
LXC	Linux Containers
MIT	Massachusetts Institute of Technology
MITM	Man-in-the-middle
NFC	Near Field Communication
NFV	Network Function Virtualization
NIST	National Institute of Standards and Technology

OPEX	OPerational EXpenditures
OSPF	Open Shortest Path First
RFID	Radio Frequency Identification
RGW	Residential Gateway
RTT	Round-Trip Time
SBI	Southbound Interface
SSL	Secure Socket Layer
STP	Spanning Tree Protocol
TSP	Telecom Service Provider
vCPU	Virtual Central Processing Unit
VLAN	Virtual Local Area Network
VM	Virtual Machine
VMP	Virtual Machine Placement
VPN	Virtual Private Network
vRAM	Virtual Random Access Memory
XSS	Cross Site Scripting

TABLE DES MATIERES

INTRODUCTION GENERALE

Dans cette introduction fondamentale, nous plongeons au cœur de la problématique à l'origine de cette thèse de doctorat, en exposant ses multiples facettes et en établissant sa pertinence. À la suite de cette exploration, nous définissons clairement les objectifs majeurs de cette recherche, ainsi que les contributions significatives qu'elle apporte au domaine scientifique. Par la suite, nous détaillons méticuleusement l'organisation de ce document, offrant une vue d'ensemble structurée pour chaque chapitre. Enfin, chaque chapitre est accompagné d'une brève description, soigneusement élaborée pour orienter le lecteur dans sa lecture et stimuler son intérêt pour les développements à venir.

1. Contexte Général

Dans le contexte économique actuel, les entreprises, en particulier les départements informatiques, ont recours aux solutions basées sur des systèmes CC (Cloud Computing) pour améliorer le rendement de leurs investissements et optimiser leurs systèmes d'information et leurs flux de travail opérationnels. L'intensification de la concurrence entre les entreprises est à l'origine de cette tendance, encore amplifiée par l'avènement de centres de données respectueux de l'environnement, qui joueront un rôle crucial dans la définition de l'avenir du cloud computing. En tirant parti de la technologie de virtualisation, les centres de données évoluent vers des modèles de déploiement basés sur le cloud computing à l'échelle mondiale, ce qui permet d'améliorer la souplesse et la réactivité de l'entreprise et de réaliser des économies.

Le concept du Cloud Computing repose sur l'utilisation de ressources informatiques à partir de sites distants. Les utilisateurs peuvent louer une partie du cloud pour leur usage personnel et le fournisseur du cloud gère les contraintes d'utilisation, tout en mettant en commun les ressources pour répondre aux besoins de tous les utilisateurs. Cette location de ressources partagées correspond à la virtualisation, car les ressources sont virtuellement infinies et toujours disponibles pour les besoins d'un utilisateur.

Pour servir efficacement les utilisateurs, le fournisseur de services cloud doit garantir deux caractéristiques essentielles : l'intégration transparente des services informatiques et l'efficacité. Cela révèle deux défis fondamentaux d’un système informatique basé sur le Cloud : gérer correctement les ressources virtuelles et garantir la performance des applications lorsqu'elles sont transférées dans le Cloud Computing. La virtualisation est une technologie fondamentale du Cloud Computing, qui prend la forme de machines virtuelles et de conteneurs, chacun ayant ses propres

avantages et inconvénients. Ces formes peuvent être imbriquées pour tirer parti de caractéristiques complémentaires ou en raison de limitations techniques, les machines virtuelles agissant comme le premier niveau et les conteneurs comme le second. Cet empilement introduit de nouveaux problèmes, mais aussi de nouvelles possibilités de gestion des ressources et d'amélioration des performances.

Dans un cloud multi-virtualisé, les défis traditionnels de la gestion des ressources dans les plateformes cloud computing persistent, même avec l'ajout de couches de virtualisation. Ces défis comprennent la nécessité de consolider les machines virtuelles, de partager efficacement les ressources et de garantir des performances constantes. Cependant, l'empilement des couches de virtualisation introduit également de nouveaux problèmes, tel que la redondance des services entre les différents niveaux de virtualisation et une coordination insuffisante entre eux en ce qui concerne la gestion des ressources.

2. Problématique Générale de la Thèse

Avec l'augmentation de la popularité des technologies de l'information, l'usage des différents services du cloud computing, a connu une progression rapide au cours des dernières années. En effet, plusieurs technologies font recours aux avantages du cloud pour leurs bons fonctionnements, ce qui implique un volume de donnée très important nécessitant ainsi un nombre élevé de machines virtuelles pour le traitement de ces applications et différents services. C'est dans cette optique que la gestion d'un centre de données d'un environnement cloud géo-distribuée exige à des fournisseurs un certain nombre de défis. Les fournisseurs de services cloud font face à des défis majeurs en matière de sécurité et d'optimisation des ressources. Un aspect crucial de cette optimisation est la mise en place de stratégies efficaces pour le placement des machines virtuelles dans le cloud.

Cette section aborde les problématiques de recherche cardinales de cette thèse ainsi que les questionnements fondamentaux résultant de l'exploration des domaines axés sur la modélisation et l'optimisation des ressources virtuelles dans le cloud. Les enjeux relatifs à la sécurité, au placement des machines virtuelles et à la réduction de la consommation énergétique suscitent une pléthore d'approches et de perspectives, soulignant la complexité inhérente à ces problématiques cruciales. La présentation détaillée qui suit offre un panorama des diverses stratégies pour relever ces défis, mettant en exergue la richesse et la diversité des méthodes envisageables :

Sécurité des équipements de l'IoT et analyse de donnée massive dans le cloud

L'utilisation croissante du cloud computing et de l'IoT dans les systèmes modernes génère des volumes massifs de données. La sécurité des systèmes doit être garantie pour protéger ces données sensibles, tandis que l'optimisation des ressources est essentielle pour maximiser l'efficacité opérationnelle. Au niveau de la sécurité on profite des avantages de la blockchain pour assurer la sécurité des équipements de l'IoT générant une masse de données dans le cloud. Mais également, on profite des bienfaits des systèmes de recommandations pour optimiser les ressources dans le cadre de l'analyse de ces données. Dans ce contexte, on se pose quelques questions cruciales :

- *Comment la Blockchain peut-elle être intégrée dans le cloud computing et l'IoT pour assurer une sécurité?*
- *Comment les recommandations issues de l'analyse de données massives peuvent-elles être intégrées dans ce modèle global, contribuant à une meilleure compréhension des schémas de comportement et facilitant la prise de décision automatisée ?*

Le problème d'optimisation du placement des machines virtuelles dans le Cloud

Au cours de ces dernières années, on observe plusieurs travaux sur le problème de placement, de migration des VM et d'amélioration des qualités de service(QoS) dans le cloud computing. Une grande partie des travaux existant, se basaient sur des algorithmes heuristiques qui génèrent des solutions moins optimales surtout pour des problèmes complexes comme il est le cas. Malgré l'usage répandu des métaheuristiques dans les travaux récents, il demeure impératif d'introduire de nouvelles solutions en adoptant des approches métaheuristiques inédites.

Ce problème de placement des machines virtuelles (VMP) prend une ampleur considérable dans la gestion des ressources au sein d'un environnement de cloud computing pour plusieurs raisons Ce qui nous conduit à nous poser la question suivante:

Pourquoi est-il impératif de résoudre la problématique du placement des machines virtuelles (VMs) dans l'environnement Cloud ?

En effet, pour répondre à cette question, nous devons examiner en détail la pertinence et le rôle de la technique de placement des machines virtuelles (VMP). Ce problème occupe une position centrale dans la gestion et la performance des infrastructures cloud. Sa pertinence repose sur son influence directe sur l'efficacité opérationnelle, la fiabilité des services, la durabilité

environnementale, et les coûts liés au cloud. L'optimisation du placement des VMs contribue significativement à maximiser l'utilisation des ressources, à garantir la disponibilité des applications, à réduire la consommation énergétique, et en fin de compte, à optimiser les dépenses associées à l'exploitation des services cloud. Plusieurs questions peuvent survenir :

- *Où et comment placer les machines virtuelles dans les centres de donnée cloud ?*
- *Quelles approches appropriées pour mieux modéliser et optimiser le placement des VM dans le cloud ?*
- *Quels critères doivent être pris en compte pour garantir une allocation efficace des ressources dans des environnements cloud hétérogènes ?*
- *Comment pouvons-nous concevoir des algorithmes de placement dynamiques capables de s'adapter aux fluctuations de la charge de travail ?*
- *Quelles métriques doit-on prendre en considération pour évaluer la performance des approches proposées ?*

Au cours de cette thèse, notre objectif est d'apporter des réponses éclairées à ces questions cruciales afin d'optimiser les ressources virtuelles dans le cloud computing. Nous nous attacherons à comprendre les mécanismes essentiels qui influent sur le placement optimal des VMs et à développer des stratégies novatrices pour répondre à ces défis.

3. Objectifs et Contributions

Cette thèse contribue à l'avancement de ce domaine en proposant des méthodologies et des algorithmes innovants conçus pour traiter les complexités associées au problème d'optimisation des ressources virtuelles dans les environnements hétérogènes du cloud. Il apporte également un aperçu sur les défis des équipements IoT générant une grande masse de données stockée et traitée dans le cloud tout en proposant une architecture sécurisée. Pour pouvoir atteindre ces objectifs, cette thèse apporte ces contributions :

Modèle Basé sur les Données Massives pour les Systèmes Sécurisés : Une Architecture Fondée sur la Blockchain pour l'Informatique en cloud et la Sécurisation de l'IoT

L'émergence spectaculaire de l'Internet des Objets (IoT) a ouvert la voie à l'utilisation de petits dispositifs pour la gestion de tâches complexes. Ce paradigme a entraîné une collecte incessante de données non structurées, souvent nécessitant une analyse approfondie et un stockage dans les centres de données cloud. Cependant, l'un des principaux défis réside dans la sécurisation des dispositifs de l'Internet des objets (IoT) ainsi que dans la disponibilité continue des données. Dans

cette première contribution, nous avons tiré parti des avantages de la technologie blockchain afin de proposer une solution destinée à renforcer la sécurité de ces technologies interconnectées. A la base, une revue de littérature est présentée, pour mettre en exergue les diverses applications de la technologie Blockchain ; pour aviver la sécurité de l'IoT et identifier les opportunités, les défis ainsi que les tendances de ces domaines en plein essor. Pour parier à ce défi de sécurité des technologies cloud, une architecture basée sur la technologie Blockchain est propoosée. Cette contribution ouvre la voie des systèmes plus fiables et résilients pour l'avenir de ces technologies.

Optimisation du placement des machines virtuelles dans un système hétérogène de centre de données cloud

La deuxième contribution repose sur la conception d'un algorithme métaheuristique novateur, inspiré par le comportement des loups, et visant à optimiser le placement dynamique des machines virtuelles au sein d'infrastructures hétérogènes. Conscient des capacités et des caractéristiques de performance distinctes des diverses ressources des centres de données, l'algorithme proposé, dénommé GWOVMP, s'adapte intelligemment à l'hétérogénéité de l'environnement. Cette adaptabilité assure une allocation optimale des machines virtuelles à l'hôte le plus approprié, tout en minimisant la consommation énergétique et le gaspillage des différentes ressources, optimisant ainsi l'efficience globale d'un environnement de cloud computing. L'étude expérimentale a été effectuée dans le simulateur CloudSim et cette approche proposée a été comparée à quatre algorithmes(BFD, RFF, MBFD et FFD) de l'état de l'art.

SOAVMP : Placement multi-objectif de machines virtuelles dans le cloud computing à base de l'algorithme d'optimisation des mouettes

Le placement optimal des machines virtuelles (VM) au sein d'un centre de données hétérogène constitue un enjeu crucial, aux répercussions significatives sur la performance, l'utilisation des ressources et l'efficience économique. Plusieurs études ont essayé d'apporter leurs contributions anfin d'améliorer la gestion optimale des ressources dans le cloud. En tenant compte des résultats de la précédentee contribution, nous avons entrepris d'apporter des améliorations significatives dans cette contribution en proposant une nouvelle approche et en utilisant d'autres métriques d'évaluation.

Nous avons conçu une méthode métaheuristique basée sur le Seagull Optimization Algorithm pour assurer une gestion optimale et multi-objectifs du placement des machines virtuelles dans le cloud.

L'algorithme SOAVMP proposé prend en considération la réduction de la consommation énergétique, la minimisation du gaspillage des ressources, la gestion des migrations ainsi que la réduction des violations des accords de niveau de service (SLA).

Dans le but d'évaluer les performances de la méthode proposée, une modélisation mathématique du comportement du cloud et de ces métriques d'évaluation ont été effectuées.

4. Publication et communication

Dans cette thèse, les différents résultats et travaux scientifiques ont été publiés sous les références suivantes :

Journal

- Ndayikengurukiye, A., Ez-zahout, A., Aboubakr , A. ., Charkaoui , Y. ., & Fouzia, O. (2022). Resource Optimisation in Cloud Computing: Comparative Study of Algorithms Applied to Recommendations in a Big Data Analysis Architecture. *Journal of Automation, Mobile Robotics and Intelligent Systems*, *15*(4), 65–75. https://doi.org/10.14313/JAMRIS/4-2021/28.
- Aristide Ndayikengurukiye, Abderrahmane Ez-Zahout, Fouzia Omary, "Optimizing Virtual Machines Placement in a Heterogeneous Cloud Data Center System", International Journal of Computer Networks and Applications (**IJCNA**), 11(1), PP: 1-12, 2024, DOI: 10.22247/ijcna/2024/224431.
- Aristide Ndayikengurukiye, Rim Doukha, Eric Niyukuri, Eric Muheto, Abderrahmane Ez-zahout, Fouzia Omary, "SOAVMP: Multi-Objective Virtual Machine Placement in Cloud Computing Based on the Seagull Optimization Algorithm", International Journal of Computer Networks and Applications (**IJCNA**), 11(3), PP: 375-389, 2024, DOI: 10.22247/ijcna/2024/24.

Conférence

- Ndayikengurukiye, Aristide, Abderrahmane Ezzahout and Fouzia Omary. "An overview of the different methods for optimizing the virtual resources placement in the Cloud Computing." *Journal of Physics: Conference Series* 1743 (2021): n. pag.
- Ndayikengurukiye, A., Ez-Zahout, A., Omary, F. (2022). Un modèle basé sur les mégadonnées pour les systèmes sécurisés : **une architecture basée sur la chaîne de blocs**

pour le cloud computing et la sécurité de l'IoT . Dans : Saeed, F., Al-Hadhrami, T., Mohammed, E., Al-Sarem, M. (eds) Advances on Smart and Soft Computing. Advances in Intelligent Systems and Computing, vol 1399. **Springer**, Singapour. https://doi.org/10.1007/978-981-16-5559-3_33

Communication

- International Conference on Mathematics and Data Science (ICMDS2020);
- The 2nd International Conference of Advanced Computing and informatics 2021 (ICACIN 2021);
- Journée Doctoriale: « Innovation Pédagogiques et Numériques ». Organisée, les 21 et 22 Décembre 2021 à l'Université Mohammed V ;
- Journée Doctoriale de la Faculté des Sciences de Rabat, du 19 au 22 Juillet 2022 ;
- The communication at the Doctoral Days of the Center for Doctoral Studies in Science and Technology, from May 15 to 20, 2023.

5. Organisation du manuscrit

La démarche de rédaction adoptée consiste à structurer chaque contribution comme un chapitre distinct, offrant une approche holistique qui explore à la fois la partie théorique et la mise en pratique de chaque aspect de la recherche. Chaque chapitre sera conçu pour fournir une compréhension approfondie des fondements conceptuels tout en démontrant la mise en œuvre concrète des idées discutées.

Cette thèse est composée par 6 chapitres. Cette partie de la thèse introduit le présent travail, le contexte et les problématiques du sujet, les objectifs à atteindre, différentes contributions apportées ainsi que les publications scientifiques effectuées.

Le premier chapitre introduit les concepts de base du cloud computing tels que les définitions selon différents chercheurs, les différentes technologies qu'utilise le cloud, les modèles de déploiement et différents services. Ce chapitre a pour objectif principal de lister et détailler les concepts clés du cloud computing et souligne l'importance de ces fondamentaux pour la compréhension et l'utilisation efficace de cette technologie.

Le deuxième chapitre se focalise sur différentes technologies associées au cloud tel que le big data et L'IoT. Dans un premier temps, ce chapitre donne des éléments clés pour comprendre ces deux

technologies et souligne l'importance de la synergie entre le Big Data et l'IoT dans la transformation numérique, en mettant en évidence les opportunités et les défis liés à l'intégration de ces deux technologies. Il encourage également la réflexion sur les implications éthiques et les meilleures pratiques pour tirer le meilleur parti de la convergence du Big Data et de l'IoT. Dans ce chapitre, une architecture basée sur la blockchain est également proposée pour sécuriser les équipements de l'Internet des Objets (IoT).

Le troisième chapitre récapitule les différentes approches étudiées et en soulignant l'importance de l'optimisation des ressources virtuelles dans le contexte du cloud computing. Dans ce chapitre, on donne l'état de l'art des algorithmes appliqués dans le placement de machines virtuelles dans le cloud ainsi que les systèmes de recommandation pour optimiser le cloud. Il encourage également la recherche continue et l'exploration de nouvelles approches pour répondre aux défis émergents dans le domaine de l'optimisation des ressources cloud.

Dans le quatrième chapitre, une étude comparative sur les algorithmes appliqués dans les systèmes de recommandations est effectuée. Il constitue une étape essentielle dans la validation des approches de recommandation pour l'optimisation du cloud computing au sein de l'architecture Big Data. Il vise à fournir une compréhension approfondie des performances des algorithmes choisis et à orienter la discussion vers des solutions pratiques et innovantes pour améliorer l'efficacité opérationnelle dans les environnements cloud.

Le cinquième chapitre vise à présenter une approche novatrice basée sur le Grey Wolf Optimization pour résoudre le problème complexe de placement des machines virtuelles dans un environnement cloud. Il met en lumière la modélisation du problème, la méthodologie de l'approche proposée, les résultats de l'évaluation, et offre des discussions approfondies pour guider la compréhension et l'appréciation de cette contribution spécifique.

Ce dernier chapitre explore une nouvelle approche qu'on a appelée MOSOAVMP, pour le placement multi-objectif des machines virtuelles dans le cloud, en mettant en avant le potentiel du Seagull Optimization Algorithm. Il offre une compréhension approfondie du modèle, de la méthodologie, des performances et des implications pratiques de cette contribution particulière.

Ce travail se termine par une conclusion générale et des perspectives d'avenir.

CHAPITRE I : APERÇU GENERAL SUR LES CONCEPTS FONDEMENTAUX DU CLOUD COMPUTING

I.1. Introduction

Indéniablement, les nouvelles technologies de l'information et de la communication (NTIC) se développent d'une manière rapide depuis ces trois dernières décennies, spécialement le cloud computing. Cette technologie est devenue un élément essentiel de la technologie moderne et son utilisation connaît une croissance exponentielle. Le cloud Computing permet aux entreprises et aux organisations d'accéder à des ressources, des données et des services stockés et gérés à distance. Il transforme la façon dont nous stockons et traitons les données, en facilitant et en rendant plus efficaces l'accès, la gestion et le déploiement des ressources.

Dans ce chapitre, nous allons nous plonger dans les détails des concepts du cloud computing, en commençant par un aperçu des définitions du cloud et ses caractéristiques. Nous explorerons également l'architecture du système, les modèles de déploiement et les différents services de cloud computing actuellement disponibles. En outre, nous discuterons des technologies qui sont utilisées dans le cloud computing et de la manière dont la virtualisation a contribué à son succès. Enfin, nous conclurons ce chapitre en abordant les fonctions de réseau virtuelles et en tirant des conclusions sur l'importance du cloud.

I.2. Définitions du cloud computing

Le cloud computing est un modèle de service qui offre diverses ressources informatiques, y compris le calcul, le stockage, les réseaux, les plateformes et les applications, sur l'internet. Ces services sont disponibles sur la base d'un paiement à l'utilisation, et les utilisateurs peuvent y accéder sans avoir besoin de connaître les spécificités du lieu ou de la manière dont ils sont hébergés[1], [2], [3].

Le terme "Cloud Computing" a été inventé en 1996, mais le concept s'aligne sur la prédiction faite par McCarthy lors de la célébration du centenaire du MIT en 1961. McCarthy envisageait un avenir où la puissance et les ressources informatiques pourraient être partagées et accessibles à distance, une vision qui s'est concrétisée grâce à cloud computing.

De nombreux chercheurs, tant dans le monde universitaire que dans l'industrie, se sont efforcés de donner une définition précise de l'informatique dématérialisée et de mettre en évidence ses caractéristiques distinctes. À l'heure actuelle, il existe toute une série de définitions et de caractéristiques uniques associées à l'informatique dématérialisée :

Le National Institute of Standards and Technology (NIST)[3] décrit le cloud computing comme un modèle offrant un accès rapide à un ensemble partagé de ressources informatiques (serveurs, stockage, applications, bande passante, etc.) nécessitant une interaction minimale avec le fournisseur de services.

Selon IBM [10], Le cloud computing, souvent appelé "cloud", offre des ressources informatiques à la demande, telles que des applications et des centres de données, accessibles via Internet. Ce modèle permet aux utilisateurs de payer uniquement pour ce qu'ils utilisent.

Selon la définition de Gartner [11], le cloud désigne une approche informatique dans laquelle des ressources informatiques flexibles, évolutives et à la demande sont mises à la disposition d'utilisateurs externes par le biais des technologies Internet. Le modèle de livraison se caractérise par cinq attributs clés, à savoir la multi-location (ressources partagées), l'évolutivité significative, l'élasticité, la tarification à l'utilisation et l'auto-approvisionnement des ressources.

En examinant ces définitions, il apparaît que la notion de cloud se divise en deux aspects principaux : technique et financier. Du côté des fournisseurs de services en cloud, les caractéristiques essentielles incluent l'élasticité, les mises à jour automatiques, l'extensibilité massive, l'auto-approvisionnement et le partage des ressources. Pour les utilisateurs finaux, les aspects clés du cloud sont l'instantanéité, la disponibilité, l'accès à la demande et la tarification à l'usage.

A partir de la Figure 1 [4] nous montrons les différents modèles, caractéristiques et services que l'on trouve dans un système basé cloud computing.

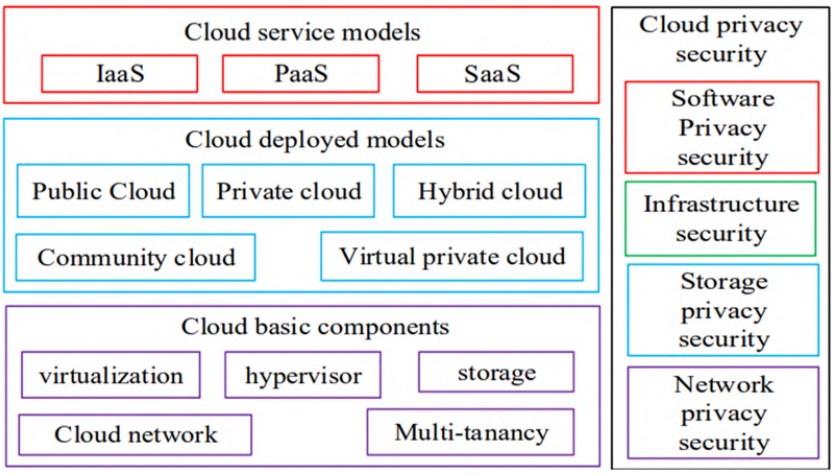

Figure 1: Différents composants du cloud computing[4]

Dans la section suivante, nous nous focalisons sur ces différentes caractéristiques du cloud d'une façon générale sans tenir compte de l'aspect technique ou financier que peut offrir.

I.3. Caractéristiques du cloud computing

En se basant sur les définitions antérieures du Cloud Computing et sur les distinctions faites par le NIST par rapport à l'informatique traditionnelle, cinq caractéristiques fondamentales émergent, reconnues par tous les chercheurs dans le domaine du Cloud.

I.3.1. Accès libre à la demande

Le cloud computing (CC) permet aux clients d'accéder à des ressources informatiques telles que serveurs, stockage, puissance de calcul et plateformes de développement en tant que services via des réseaux, selon leurs besoins spécifiques. Ce modèle offre une utilisation simple et flexible, sans nécessiter de contact direct avec le fournisseur de services. Conçu comme un système autogéré, le cloud peut s'adapter aux changements internes et externes. Le matériel, les logiciels et les données peuvent être reconfigurés, orchestrés et consolidés automatiquement pour optimiser l'expérience utilisateur[5].

I.3.2. Ressources partagées

Le CC est un paradigme informatique moderne qui fonctionne sur la base d'un modèle commercial qui consolide et offre des services et des ressources à plusieurs utilisateurs sur une base partagée. Les utilisateurs n'ont généralement pas connaissance de l'emplacement spécifique où les ressources sont hébergées ou n'en contrôlent pas la gestion[6]. Cela implique que, contrairement à l'informatique traditionnelle, les ressources physiques et virtuelles sont allouées et réallouées de manière dynamique en réponse aux variations de la demande.

I.3.3. Élasticité

Le cloud computing offre la possibilité d'augmenter ou de réduire rapidement les ressources informatiques selon les besoins de l'utilisateur. A la différence des infrastructures traditionnelles aux capacités restreintes, le cloud computing offre la possibilité de redimensionner considérablement les ressources comme la bande passante et l'espace de stockage. Une fois que ces ressources ne sont plus nécessaires, elles peuvent être facilement libérées pour d'autres usages. Cette élasticité est le résultat de services cloud qui peuvent répondre aux demandes croissantes des utilisateurs. Pour obtenir cette évolutivité de manière automatique et en cours d'exécution, l'évolutivité linéaire est couramment utilisée. Cette technique décompose les tâches principales en éléments plus petits et les distribue dans l'infrastructure virtuelle du Cloud[7].

I.3.4. Accèssibilité des Services par un réseau

Les services de Cloud computing doivent être accessibles sur le réseau via des mécanismes standardisés, permettant leur utilisation sur diverses plateformes hétérogènes telles que les ordinateurs portables, les postes de travail, les tablettes et les téléphones mobiles[8] .

I.3.5. Service mesuré

Les services cloud sont mesurés et facturés selon un modèle à la demande. Cette approche permet au fournisseur de services et au client de contrôler l'utilisation et de déterminer les dépenses en fonction de la quantité de ressources consommées.

I.4. Différents services du Cloud Computing

Le cloud computing offre des services informatiques qui peuvent être consommés à la demande par des individus ou des organisations. La Figure 2 illustre les différentes formes de services disponibles, en fonction du niveau de responsabilité de la gestion des couches de l'environnement informatique standard. Dans un environnement informatique traditionnel, cet environnement est constitué de couches qui vont du matériel physique, au niveau le plus bas, aux applications utilisées, au niveau le plus élevé. Ces couches comprennent l'environnement d'exécution, les données, les applications, les logiciels intermédiaires, l'environnement de développement, le système d'exploitation, la virtualisation, le calcul, le réseau et le stockage. Toutefois, dans l'environnement cloud, les utilisateurs ne sont pas responsables de la gestion de toutes ces couches, et le type de service fourni dépend du niveau de responsabilité pour les sous-ensembles de couches.

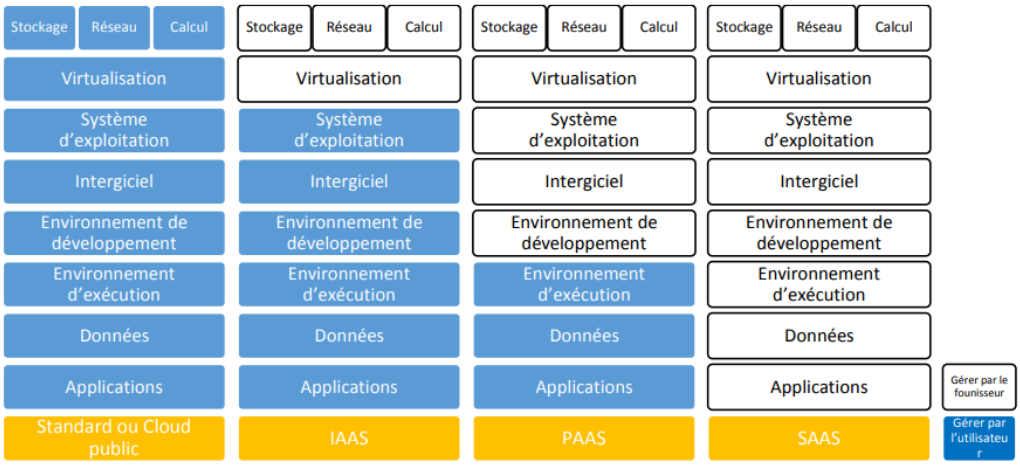

Figure 2 : Les services du cloud computing[6]

I.4.1. Software-as-a-Service (SaaS)

Le type SaaS des services du cloud computing implique la dématérialisation du matériel, de l'hébergement, du cadre d'application et des logiciels, qui sont ensuite fournis en tant que services à la demande[7], [9]. Ce modèle libère les utilisateurs de la gestion et du contrôle des ressources, ne leur laissant que des options de configuration spécifiques, tandis que le fournisseur gère les ressources nécessaires[10]. Les services, tels que les outils ERP(Enterprise Ressource Planning), les serveurs web, les outils de collaboration, les CRM(customer relationship management), les serveurs de messagerie, etc. sont hébergés dans le centre de l'utilisateur. Exemples de services SaaS incluent Yahoo Email, Gmail, Google Apps, Dropbox, Office 365, Salesforce CRM, Google Documents, Facebook, Twitter, et bien d'autres.

Cependant, héberger des données sur des VMs de fournisseurs de services cloud pose des préoccupations générales de confidentialité et de sécurité d'une manière générale. Les utilisateurs sont également limités à l'offre du fournisseur, ce qui peut constituer un inconvénient. En outre, ce modèle soulève des questions quant à la durabilité du fournisseur.

I.4.2. Platform-as-a-Service(PaaS)

Le PaaS est une plateforme pratique qui offre aux développeurs de logiciels un environnement à la demande, comprenant le matériel, l'infrastructure et les outils de développement, pour gérer les applications sans les tracas et les dépenses liées à la maintenance des systèmes sur site[11].

Lorsque l'on utilise le PaaS, le fournisseur de service en cloud s'occupe de tout, y compris des serveurs, des réseaux, du stockage, des systèmes d'exploitation, des logiciels intermédiaires et des bases de données, dans son centre de données. Les développeurs peuvent facilement sélectionner les serveurs et les environnements spécifiques nécessaires pour créer, tester, déployer, maintenir, mettre à jour et faire évoluer les applications à partir d'un menu.

Aujourd'hui, le PaaS est souvent basé sur des conteneurs, un modèle informatique virtualisé qui va plus loin que les serveurs virtuels. Les conteneurs permettent aux développeurs d'emballer des applications avec seulement les services de système d'exploitation nécessaires pour fonctionner sur n'importe quelle plateforme, sans intergiciel ni modification[7].

I.4.3. Infrastructure-as-a-Service(IaaS)

L'infrastructure en tant que service (IaaS) offre aux utilisateurs un accès aux ressources informatiques essentielles telles que les serveurs physiques et virtuels, le réseau et le stockage sur la base d'un paiement à l'utilisation, avec la commodité d'une disponibilité à la demande sur l'internet. Les utilisateurs de l'IaaS peuvent faire évoluer les ressources en fonction de leurs

besoins, ce qui élimine la nécessité d'investissements initiaux importants, d'une infrastructure sur site ou d'un surprovisionnement pour faire face à des pics d'utilisation occasionnels. Comparé à d'autres modèles de cloud computing tels que le SaaS, le PaaS et les conteneurs, l'IaaS est celui qui offre le moins de contrôle sur les ressources du cloud computing.

L'utilisation de l'acronyme aaS a connu une croissance considérable, et il sert maintenant de terme global pour XaaS (Everything as a Service). La lettre "X" dans XaaS signifie "tout" ou "n'importe quoi". Les trois services de base (IaaS, PaaS et SaaS) sont fondamentaux pour identifier le type de service. La notation XaaS est utilisée pour classifier les services et ressources comme des sous-ensembles de ces fondamentaux types principaux. Par exemple, SecaaS (Security as a Service), BaaS (Backup as a Service), CaaS (Communication as a Service), DaaS (Data as a Service), DBaaS (Database as a Service) et MaaS (Monitoring as a Service) sont quelques-unes de ces abréviations[12], [13].

I.5. Modèles de déploiement

La classification des modèles du cloud computing est basée sur la façon dont les parties prenantes utilisent les ressources physiques, qui peuvent être situées soit chez l'utilisateur, soit chez le fournisseur, et sur le fait qu'elles sont partagées ou dédiées. En outre, ces modèles peuvent appartenir aux entreprises, aux universités, ou à d'autres types d'utilisateurs. Par conséquent, l'informatique dématérialisée offre quatre typologies ou modèles de déploiement comme le montre la Figure 3.

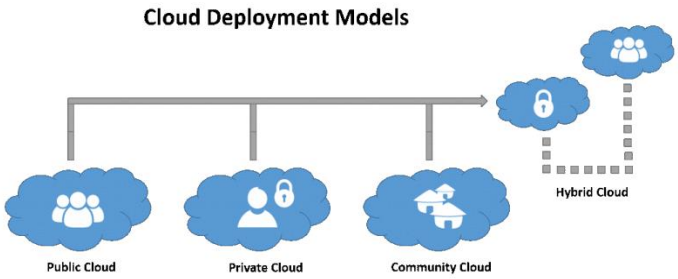

Figure 3 : Les modèles de déploiements du CC[14]

I.5.1. Cloud Privé

Le modèle de déploiement privé est conçu pour répondre exclusivement aux besoins des entreprises privées qui utilisent toutes les ressources disponibles. Ce modèle comprend des réseaux propriétaires, généralement des centres de données situés dans les locaux de l'entreprise, chargés de contrôler et de gérer les ressources du cloud. Les principales raisons de choisir ce modèle sont

les questions d'intégration et de sécurité liées aux données et applications critiques. Cependant, un cloud privé n'est pas toujours plus sûr. Il est essentiel de sécuriser l'environnement de virtualisation, y compris la sécurité de l'hyperviseur, du matériel physique, et des logiciels. Cette sécurisation reste cruciale, même dans un cloud privé, la sécurité n'est pas la seule responsabilité.) est toujours nécessaire, alors que dans un cloud privé, la sécurité n'est pas la seule responsabilité. En revanche, le fournisseur de cloud public est responsable de la sécurité dans un cloud public[15].

Il existe quatre types distincts de cloud privés, chacun ayant ses propres caractéristiques. Le premier est le cloud privé standard, dans lequel une organisation héberge le cloud dans son propre centre de données et le sécurise à l'aide d'un pare-feu. Le deuxième type est le cloud privé géré, dans lequel un fournisseur tiers gère et contrôle l'infrastructure appartenant à l'organisation. Le troisième type est le cloud privé hébergé, où les fournisseurs de services fournissent l'infrastructure et assurent la gestion pour une organisation, sans partage de ces ressources avec d'autres entités. Enfin, le quatrième type est le cloud privé virtuel, où les services cloud sont offerts dans un environnement multi-tenant par des fournisseurs. Le lieu ou il est hébergé est connu du client, est généralement situé dans le même pays que celui-ci[16].

I.5.2. Cloud Public

La majorité des clients de l'internet s'appuient sur le cloud public, qui est un modèle de cloud bien établi. Dans ce modèle, les consommateurs et les fournisseurs de services sont des entités distinctes, et les ressources sont dynamiquement auto-approvisionnées par le biais d'applications ou de services web dans un environnement multi-locataires. Ce modèle de cloud offre aux utilisateurs facilité et flexibilité sans nécessiter d'investissement initial, ce qui en fait une solution idéale. Toutefois, le fournisseur assume l'entière responsabilité du contrôle d'accès et de la sécurité, ce qui limite la liberté des clients de contrôler et de configurer leurs ressources[17]. Des améliorations significatives ont été apportées au modèle de déploiement IaaS et de nombreuses entreprises, dont Amazon avec Elastic Compute Cloud [18], Rackspace's Cloud Offerings et IBM's BlueCloud, ont investi dans cet espace. En outre, il existe des offres de cloud public sous la forme d'applications ou de plateformes en tant que service, comme AppEngine de Google et Azure Services Platform[19], SimpleDB, Cloud Front et S3 Simple Storage.

I.5.3. Cloud communautaire

Le modèle du cloud collectif ou communautaire implique plusieurs organisations indépendantes ayant des intérêts communs et mettant en commun des ressources d'infrastructure. Ces

organisations peuvent aussi collaborer sur des tâches de gestion, comme la sécurité des données, le déploiement d'applications et l'authentification[20]. Le cloud communautaire offre l'avantage majeur de permettre à ces entités indépendantes de partager les bénéfices financiers d'un cloud privé tout en évitant les problèmes de sécurité et de conformité réglementaire souvent associés aux clouds publics génériques sans garanties spécifiques dans leurs SLA. Les cloud communautaires gagnent en popularité aux États-Unis et dans l'UE parmi les organismes gouvernementaux nationaux ou locaux, avec différents types de modèles envisagés.

Il existe deux types principaux de cloud communautaires. Le premier est le modèle fédéré, dans lequel toute organisation au sein de la communauté peut accéder aux ressources d'une autre. Le second est le modèle de tiers de confiance, dans lequel un "courtier" gère l'acquisition de services critiques et les met à la disposition de tous les membres.

I.5.4. Cloud Hybride

Le cloud hybride est une forme intégrée d'informatique en cloud qui combine deux ou plusieurs serveurs en cloud, tels que les cloud privés, publics ou communautaires, tout en conservant leurs identités individuelles. Ce type d'infrastructure peut transcender les frontières et l'isolement des fournisseurs, ce qui rend difficile sa classification en tant que cloud purement public, privé ou communautaire. L'avantage des cloud hybrides est que les utilisateurs peuvent améliorer leur capacité et leurs possibilités en intégrant, en agrégeant et en personnalisant un autre service ou paquet de cloud. La gestion des ressources peut être effectuée en interne ou par des fournisseurs externes. Les cloud hybrides offrent une solution optimale pour les échanges de charge de travail entre les cloud privés et publics, en fonction des besoins et des demandes de l'organisation. Par exemple, les tâches non critiques telles que les charges de travail de développement et de test peuvent être placées sur le cloud public d'un fournisseur tiers, tandis que les tâches sensibles ou critiques doivent être hébergées en interne. Les organisations peuvent tirer parti du modèle de cloud hybride pour traiter des ensembles de données volumineux, en profitant de caractéristiques telles que l'évolutivité, la flexibilité et la sécurité[20].

I.6. Architecture d’un système cloud

Selon l'architecture du Cloud, cinq éléments clés influencent et sont influencés par la technologie, ainsi que par ses implications en matière de sécurité. La Figure 4 fournit une représentation

visuelle de l'architecture du cloud, qui comprend une conception de référence complète décrivant les couches du modèle d'interconnexion des systèmes ouverts (OSI). En raison de sa complexité, le cloud présente de nombreux domaines de vulnérabilité[21]. Les composants du cloud computing sont les suivants :

Le consommateur du cloud : Il s'agit d'une personne ou d'une organisation qui utilise des services cloud pour ses besoins professionnels, personnels ou relationnels[21].

Fournisseur du cloud : Il s'agit d'une personne ou d'une organisation qui offre des services cloud aux parties intéressées[21].

Auditeur du cloud: Il s'agit d'un groupe qui peut effectuer des audits indépendants des services en cloud, y compris les opérations, la mise en œuvre et la sécurité des données et des systèmes des utilisateurs du cloud[21].

Courtier en cloud : Il s'agit d'une entité qui supervise l'utilisation, l'implémentation et la prestation de services dans le cloud, facilitant les interactions entre les utilisateurs et les prestataires de services cloud[22].

Cloud Carrier : Il s'agit d'un support qui fournit un réseau de services en cloud depuis les fournisseurs du cloud jusqu'aux consommateurs de cloud[21].

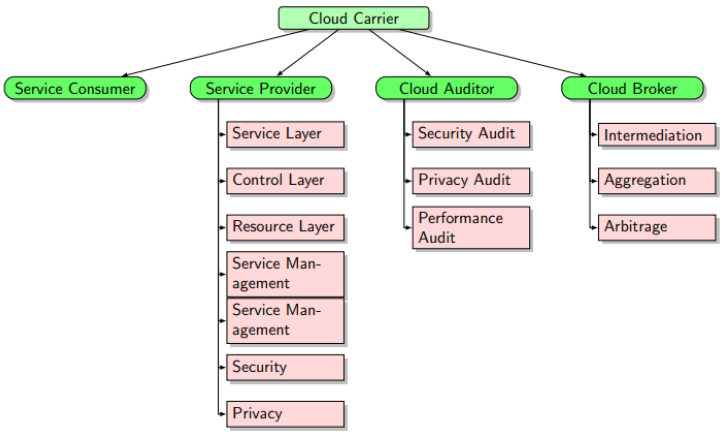

Figure 4 : Représentation visuelle de l'architecture du cloud[17]

I.7. Les différentes technologies utilisées dans le Cloud

Le CC offre aux utilisateurs des services informatiques personnalisés, fiables et dynamiques, tout en garantissant QoS. Il convient toutefois de noter que le cloud n'est pas un concept entièrement

nouveau. Il est né de la convergence de plusieurs technologies informatiques contemporaines, notamment l'informatique en grille et la virtualisation.

La complexité et les exigences des applications ne cessant de croître, de nouvelles technologies telles que l'informatique autonome et la virtualisation ont vu le jour. La virtualisation consiste à abstraire les différentes ressources informatiques des applications et des utilisateurs, offrant ainsi une flexibilité et une gestion optimisée des ressources dans les environnements informatiques. L'informatique autonome fait référence à la capacité des ressources informatiques distribuées à s'autogérer en s'adaptant aux changements imprévisibles(Ce point sera traité dans l'une des sections de ce chapitre). Ces différentes technologies de base ont ouvert la voie au développement d'autres technologies informatiques telles que les clusters, les grilles et le CC.

I.7.1. La virtualisation

Le cloud computing s'appuie fortement sur la virtualisation, qui est un élément fondamental de tous les modèles de services. La virtualisation a été introduite pour la première fois dans les années 1960 par IBM, comme un moyen d'optimiser les ressources matérielles en les divisant en plusieurs environnements virtuels[23]. Cette technologie permet de créer des versions virtuelles des serveurs, des applications, de dispositifs de stockage et de ressources réseau, ce qui simplifie la gestion des ressources et des infrastructures en cloud computing. Elle facilite également le partage des ressources entre plusieurs utilisateurs, améliore l'utilisation des ressources et réduit les coûts énergétiques et d'hébergement. Les méthodes de virtualisation varient dans leurs objectifs et peuvent être classées en trois types. La Figure 5 montre la taxonomie simplifié de de la technologie de la virtualisation[24].

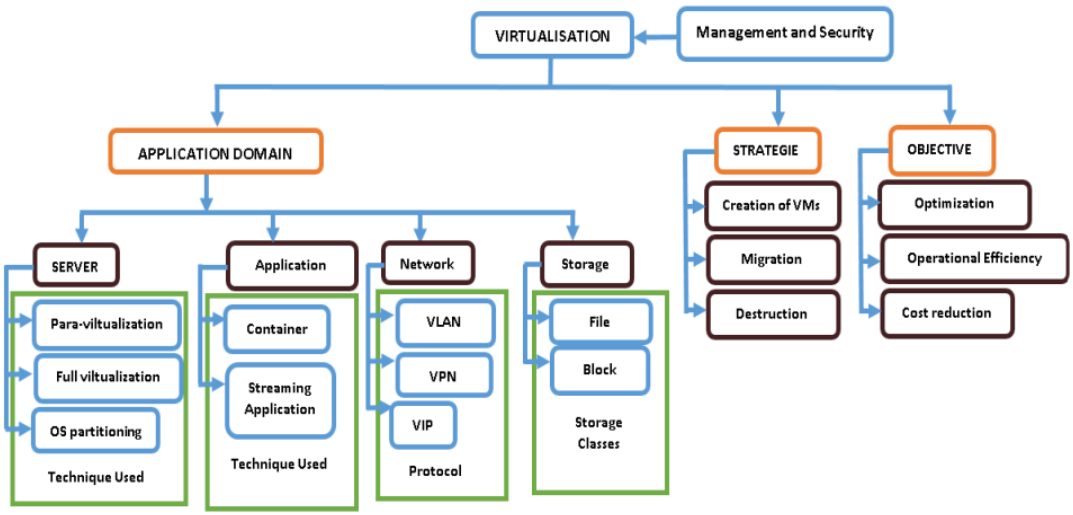

Figure 5 : Présentation taxonomique de la virtualisation[24]

I.7.1.1. Virtualisation des serveurs

La virtualisation des serveurs est une technologie qui permet à plusieurs systèmes de fonctionner simultanément sur un seul serveur physique. Ce résultat est obtenu par la consolidation de plusieurs serveurs virtuels, appelés machines virtuelles (VM), sur un seul serveur physique. La virtualisation des serveurs est principalement réalisée à travers l'utilisation d'un logiciel appelé hyperviseur. Cet hyperviseur, également connu sous le nom de moniteur de machines virtuelles, agit comme une couche intermédiaire entre le matériel physique du serveur et les machines virtuelles (VM) qui fonctionnent dessus. Son rôle principal est de créer et de gérer ces machines virtuelles en les isolant les unes des autres et en les faisant fonctionner de manière indépendante, comme si elles étaient des serveurs physiques distincts. Dans la virtualisation basée sur un hyperviseur, VM, également appelées instances, disposent de leurs propres ressources virtuelles, notamment des processeurs virtuels, des cartes réseau virtuelles, de la mémoire virtuelle et des disques virtuels, ainsi que de systèmes d'exploitation distincts appelés systèmes d'exploitation invités. Ces VM sont indépendantes et isolées les unes des autres, de sorte qu'un dysfonctionnement ou une panne dans une VM n'affectera pas les performances des autres VM. Un moniteur de machine virtuelle ou virtual Machine Manager en anglais (VMM) gère les VM, servant de couche d'abstraction qui découple la couche matérielle de la couche logicielle. La virtualisation basée sur un hyperviseur peut être classée en trois catégories : la virtualisation complète, la paravirtualisation et la virtualisation assistée par le matériel[25]. Ces catégories se

distinguent par la manière dont les systèmes d'exploitation hôtes et invités interagissent avec l'hyperviseur.

I.7.1.2. Virtualisation des stockages

Le principe de la virtualisation du stockage est très proche de celui de la virtualisation des serveurs. Il s'agit essentiellement de combiner la capacité de stockage de nombreux dispositifs de stockage interconnectés en un dispositif de stockage virtuel unifié qui peut être géré de manière centralisée à partir d'une console unique. Ce faisant, il devient beaucoup plus simple d'adapter la capacité de stockage en fonction des besoins et de migrer les données entre les dispositifs. En outre, les utilisateurs ont une vue d'ensemble de toute l'infrastructure de stockage, ce qui leur permet de gérer efficacement les données en toute simplicité.

I.7.1.3. Virtualisation de réseau

La virtualisation des réseaux permet à des réseaux logiques de fonctionner sur un réseau physique partagé. Elle fonctionne de manière similaire à la virtualisation des serveurs, qui virtualise les ressources telles que vCPU et vRAM. Dans le cas de la virtualisation de réseau, les cartes d'interface réseau, les commutateurs, les routeurs, les équilibreurs de charge et les pare-feu sont virtualisés pour fournir une gamme de fonctions réseau. Cette approche est souvent combinée à la virtualisation des ressources dans les environnements cloud pour créer des plateformes virtualisées pour les utilisateurs. De nombreuses technologies ont été développées pour faciliter la virtualisation du réseau, notamment Vlan, VPN et VXlan, entre autres.

I.7.2. La conteneurisation

Au niveau de l'application, on utilise un type de virtualisation qui crée plusieurs espaces utilisateurs isolés sur un noyau partagé, et chaque instance est appelée "conteneur". Cette approche est similaire à celle des modules d'application partitionnés, qui communiquent par le biais de services et d'applications web. Bien qu'indépendants, les conteneurs partagent un noyau et un espace mémoire communs, ce qui permet de créer un environnement normalisé contenant le code nécessaire, les paramètres d'exécution, les systèmes de fichiers réseau et les bibliothèques dont bénéficient les développeurs d'applications.

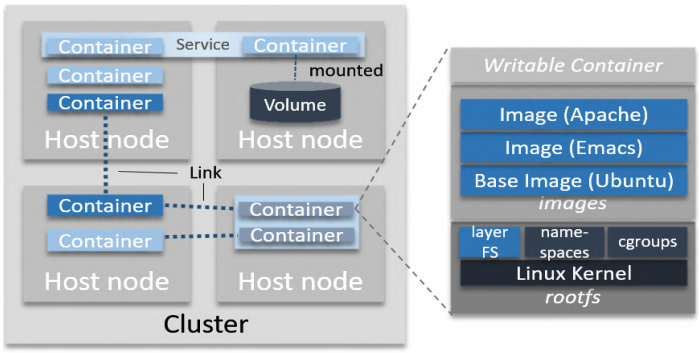

Figure 6 : Les composants d'un conteneur[26]

Ce passage décrit une forme de virtualisation qui opère au niveau de l'application. Le processus consiste à créer plusieurs espaces utilisateurs indépendants qui partagent un noyau commun. Ces espaces utilisateurs indépendants sont appelés "conteneurs". La séparation entre ces conteneurs est similaire aux modules d'application partitionnés, qui s'appuient sur des services web et des applications pour la communication. La Figure 6 montre les composants d'un conteneur.

I.7.2.1. Principes de la technologie des conteneurs

Un conteneur est un paquet autonome qui comprend toutes les parties nécessaires d'une application, ainsi que tout logiciel intermédiaire et logique d'entreprise requis dans des formats binaires et de bibliothèque. Les moteurs de conteneurs tels que Docker utilisent les conteneurs comme un moyen portable d'empaqueter les applications, ce qui entraîne la nécessité de gérer les dépendances entre les conteneurs dans les applications multi-niveaux. Pour résoudre ce problème, un plan d'orchestration peut être créé pour décrire les composants, leurs dépendances et leur cycle de vie dans un plan en couches. Ce plan peut être exécuté par des agents, tels qu'un moteur de conteneurs, au sein d'un cloud PaaS, ce qui permet de déployer des applications à partir de conteneurs. Dans ce contexte, l'orchestration implique la coordination de la construction, du déploiement et de la gestion continue de l'application.

Plusieurs solutions de conteneurs s'appuient sur les techniques Linux LXC, les distributions Linux récentes intégrant des mécanismes du noyau tels que les espaces de noms et les cgroups pour isoler les processus au sein d'un système d'exploitation partagé[26], [27], [28].

La solution de conteneur Docker est actuellement la plus populaire et sert d'exemple de conteneurisation. Une image Docker est construite à partir de systèmes de fichiers en couches, en utilisant les mécanismes LXC d'une manière similaire à la pile de virtualisation Linux. À l'aide

d'un montage en union, un système de fichiers inscriptible est ajouté à un système de fichiers en lecture seule, ce qui permet d'empiler plusieurs systèmes de fichiers en lecture seule les uns sur les autres. Cette fonctionnalité permet de créer de nouvelles images en s'appuyant sur des images de base existantes. Seule la couche supérieure, qui est le conteneur lui-même, est accessible en écriture. La conteneurisation simplifie la transition entre les applications individuelles et les grappes d'hôtes de conteneurs, ce qui permet aux applications conteneurisées de fonctionner de manière transparente sur les hôtes de la grappe. Dans cette configuration, les conteneurs hôtes individuels sont regroupés en grappes interconnectées, chacune étant constituée de plusieurs nœuds. Les services d'application sont des groupes logiques de conteneurs qui partagent la même image et qui permettent aux applications d'être mises à l'échelle sur différents nœuds hôtes. Les volumes sont des mécanismes utilisés pour les applications qui nécessitent une persistance des données, et les conteneurs peuvent monter ces volumes pour le stockage. Les liens permettent à deux ou plusieurs conteneurs de se connecter et de communiquer. Pour mettre en place et gérer ces clusters de conteneurs, un support d'orchestration est nécessaire pour permettre la communication inter-conteneurs, les liens et les assemblages de services[26], [28], [29].

I.7.2.2. Architectures de conteneurs basées sur le cloud

L'orchestration de conteneurs ne se limite pas au démarrage ou à l'arrêt des conteneurs et à leur déplacement entre les serveurs. Elle englobe la création et la gestion continue de grappes distribuées d'applications logicielles basées sur des conteneurs. Grâce à l'orchestration de conteneurs, les utilisateurs peuvent définir la manière dont les conteneurs sont coordonnés dans le cloud lors du déploiement d'applications multi-conteneurs[28].

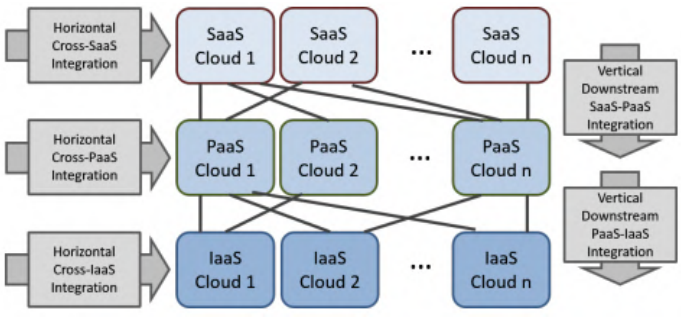

Figure 7 : Modèle d'architecture de référence du cloud[26]

Ce processus implique non seulement le déploiement initial des conteneurs, mais aussi la gestion de plusieurs conteneurs en tant qu'entité unique. L'orchestration des conteneurs gère la disponibilité, la mise à l'échelle et la mise en réseau des conteneurs, qui sont des facteurs critiques

dans un environnement distribué en cloud. L'architecture en cloud peut être considérée comme un système distribué et hiérarchisé, avec une infrastructure de base, une plateforme et des applications logicielles réparties dans des environnements multi-cloud[30]. La technologie des conteneurs est un outil précieux pour réaliser cette orchestration dans le cloud.

I.7.2.3. L'orchestration des conteneurs

Comme nous l'avons mentionné précédemment, la technologie des conteneurs Linux (LXC) est à la base du développement de l'architecture des microservices[31]. A ces technologies s'ajoutent d'autres tels que : OpenVZ[32], Docker[33], Singularity[34] et uDocker[35], ce qui a révolutionné la façon dont nous abordons le développement d'applications. Plutôt que de nous concentrer uniquement sur les opérations, nous avons désormais la possibilité de nous concentrer sur la programmation. Ce changement a été rendu possible par des outils d'orchestration populaires tels que Kubernetes[36], Docker Swarm[37] et Apache Mesos[36], qui fournissent une gestion du cycle de vie des conteneurs et des cadres pour gérer les conteneurs au sein d'une architecture micro-services. Ces cadres offrent une série de fonctionnalités, notamment l'ordonnancement des conteneurs avec des ressources limitées, la tolérance aux pannes et la mise à l'échelle automatique. Parmi les outils d'orchestration disponibles, Kubernetes[36] et OpenShift[38] sont de plus en plus adoptés dans divers systèmes informatiques, y compris dans les domaines industriels et scientifiques.

I.7.3. Centres de données

Un centre de données est un site où se trouvent divers équipements informatiques tels que serveurs, dispositifs de stockage, et infrastructures réseau. En plus, il comprend des équipements non informatiques comme des systèmes de refroidissement et des alimentations sans interruption. Cette colocation est nécessaire en raison des exigences environnementales partagées et des besoins de sécurité physique, ainsi que de la commodité de la maintenance. En regroupant les ressources informatiques en un seul lieu, plutôt que de les répartir géographiquement, les centres de données permettent de partager l'énergie, d'utiliser plus efficacement les ressources informatiques partagées et d'améliorer l'accessibilité. La taille des centres de données peut varier considérablement, allant de petites salles de serveurs pour les petites et moyennes entreprises à des fermes de serveurs à grande échelle pour les services en cloud[8].

I.8. Informatique Autonomique

La gestion de l'infrastructure des systèmes informatiques modernes peut s'avérer une tâche ardue en raison de leur complexité et de leur charge de travail dynamique. Pour assurer un fonctionnement harmonieux, les organisations doivent contrôler en permanence des centaines de composants et des milliers de paramètres de réglage[39]. Cependant, la quantité d'efforts humains nécessaires pour exploiter et maintenir ces systèmes peut rapidement devenir écrasante[40].

Pour relever ce défi, l'informatique autonome est apparue comme une solution qui élimine le besoin d'intervention humaine. Ce champ émergeant, inspiré par l'initiative d'informatique autonome d'IBM, cherche à créer des systèmes informatiques autonomes capables de s'auto-réguler en fonction des objectifs définis par les administrateurs. En prenant exemple sur le fonctionnement du corps humain, où des fonctions vitales comme la respiration, la digestion et les ajustements pupillaires sont régulées de manière inconsciente par le système nerveux autonome, IBM cherche à appliquer cette autonomie à des systèmes informatiques complexes.

Dans le domaine des systèmes informatiques, il est essentiel que les systèmes soient autonomes, capables de gérer eux-mêmes les tâches de maintenance et d'optimisation. Cette capacité permet de soulager la charge de travail des administrateurs système, tout en assurant le bon fonctionnement et l'efficacité globale du système[41].

I.9. Les réseaux virtuels programmables

Dans cette section, nous allons explorer deux concepts fondamentaux qui révolutionnent les réseaux : La mise en réseau définie par logiciel (SDN) et la virtualisation des fonctions de réseau (NFV). Ces paradigmes de pointe apportent une série de propriétés et de capacités nouvelles aux architectures de réseau conventionnelles, notamment la programmabilité et la virtualisation des réseaux. En adoptant SDN et NFV, les réseaux peuvent bénéficier d'une plus grande flexibilité et d'une optimisation des ressources.

I.9.1. Les réseaux définis par logiciels (Soft Define Networking : SDN)

Le réseau défini par logiciel (SDN) est un concept en développement qui vise à relever les défis de la mise en œuvre, de la gestion et de la modification de l'infrastructure de réseau en séparant les fonctions de contrôle (plan de contrôle) et de transfert de données du réseau (plan de données). Ce faisant, le SDN offre de nombreux avantages, notamment une flexibilité et une programmabilité accrues des réseaux, ce qui facilite la gestion et l'amélioration des performances

du réseau. Pour ce faire, on crée une division entre le plan de contrôle et les routeurs et commutateurs sous-jacents, ce qui permet d'obtenir des dispositifs de réseau qui fonctionnent comme de simples dispositifs de transfert gérés par le plan de contrôle[27], [42]. La fig 7 présente l'architecture de ce type de réseau.

I.9.1.1. Architecture SDN

L'architecture SDN sépare les fonctions du plan de contrôle et du plan de données du réseau, ce qui permet de centraliser l'intelligence du réseau et de fournir une infrastructure de réseau totalement indépendante de tout service de réseau. En se basant sur la Figure 8, cette architecture se compose de trois plans principaux[43]. Dans la meme figure, nous représentons l'architecture SDN avec ses différentes fonctions.

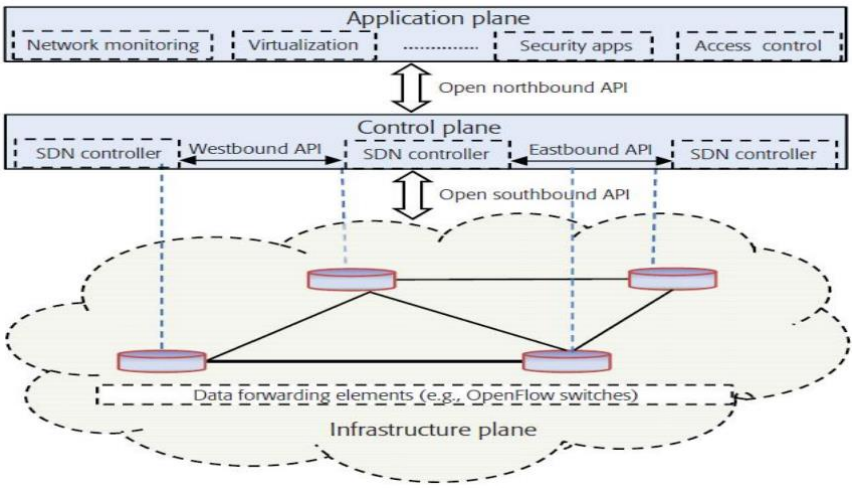

Figure 8 : Architecture SDN[42]

Le plan de données : Également appelé plan d'infrastructure, il s'agit du niveau le plus bas de l'architecture SDN. Il comprend des dispositifs de transmission tels que des commutateurs matériels et logiciels, des routeurs et des pare-feu, chargés de transmettre le trafic selon un ensemble de règles prédéfinies[44].

Plan de contrôle : Il s'agit du niveau central de l'architecture SDN et englobe tous les contrôleurs, qui constituent le cœur du réseau. Le contrôleur est une entité logique centralisée qui surveille et gère l'état du réseau. Il offre une vue globale et unifiée du réseau, ce qui simplifie l'administration et la configuration des dispositifs de bas niveau et la détermination du comportement du trafic. Le plan de contrôle orchestre la demande des applications en fournissant des règles d'acheminement spécifiques aux dispositifs d'acheminement. La communication entre le plan de contrôle et le plan

de données est établie par l'intermédiaire d'une API de sortie sud (SBI), qui est une interface descendante. L'OpenFlow est l'API la plus répandue, mais d'autres sont également disponibles[45].

Plan d'application : Il s'agit du niveau le plus élevé de l'architecture SDN, qui comprend diverses applications telles que les applications commerciales, la surveillance de la sécurité et le contrôle d'accès. Ces applications permettent une gestion souple et flexible du réseau[46], [47].

I.9.1.2. Comparaison des réseaux traditionnels et les réseaux SDN

Dans les réseaux traditionnels, le plan de contrôle est distribué et chaque appareil du réseau fonctionne indépendamment en utilisant certains des protocoles tels que l'Address Resolution Protocol (ARP), utilisé pour la résolution des adresses IP en adresses MAC dans les réseaux locaux. Le Spanning Tree Protocol (STP) est également crucial pour prévenir les boucles de données en désactivant les liens redondants dans un réseau de commutation Ethernet. Pour le routage interne, l'Open Shortest Path First (OSPF) et l'Enhanced Interior Gateway Routing Protocol (EIGRP) sont largement utilisés pour déterminer les chemins optimaux à travers un réseau IP. Enfin, pour le routage externe entre les systèmes autonomes, le Border Gateway Protocol (BGP) est indispensable, permettant l'échange de routes et la prise de décisions de routage entre les routeurs. Ces protocoles constituent des éléments fondamentaux dans la conception et le fonctionnement des réseaux modernes, assurant une connectivité fiable et efficace entre les périphériques et les réseaux[46], [48]. Cette approche décentralisée signifie qu'aucune machine centralisée ne gère que l'ensemble du réseau, contrairement au SDN qui est généralement basé sur des logiciels et offre un meilleur contrôle et une meilleure gestion des ressources à distance dans le plan de contrôle[49]. Les réseaux traditionnels utilisent du matériel physique comme des commutateurs et des routeurs pour établir des connexions et faire fonctionner le réseau. En revanche, les contrôleurs SDN utilisent une interface northbound qui communique avec des API, ce qui permet aux développeurs d'appareils de programmer explicitement le réseau au lieu de s'appuyer sur les protocoles requis[49].

Les réseaux conventionnels ont tendance à combiner les plans de données et les plans de contrôle dans une seule unité physique, ce qui peut entraîner une augmentation de la charge de trafic et de la charge sur l'unité centrale et la mémoire. En revanche, le SDN sépare les plans de contrôle et de données, qui peuvent être facilement gérés par le contrôleur pour prendre des décisions éclairées et configurer le réseau en réduisant la charge de trafic. En tant qu'alternative populaire

au réseau traditionnel, le SDN permet aux responsables informatiques d'étendre les services d'infrastructure et la bande passante du réseau sans investir dans du nouveau matériel. En revanche, la mise en réseau traditionnelle nécessite du nouveau matériel pour augmenter la puissance du réseau[42], [50].

I.9.2. La virtualisation des fonctions réseau (NFV)

L'évolution constante des demandes des clients nécessite la création de nouveaux services. Cependant, dans les réseaux traditionnels, l'hétérogénéité des équipements et l'incompatibilité entre les différentes plates-formes et technologies de fabrication ont rendu complexes le développement et le déploiement de nouveaux services, augmentant à la fois les certains coûts pour les fournisseurs de services et d'infrastructures. Pour relever ces défis, l'Institut européen des normes de télécommunications (ETSI) a proposé une technologie appelée "virtualisation des fonctions du réseau" (NFV)[27], [51]. Cette technologie s'appuie sur la virtualisation, permettant aux services de réseau tels que les routeurs, les pare-feu et les équilibreurs de charge d'être hébergés sur des VM. Cela offre une flexibilité sans précédent dans la création et la gestion des services de réseau, réduisant ainsi les coûts CAPital EXpenditures (CAPEX) et OPerational EXpenditures (OPEX). Par conséquent, les administrateurs de réseau n'ont plus besoin d'acheter du matériel dédié[51], [52].

I.9.2.1. Architecture du VNF

Selon l'ETSI, l'architecture NFV comprend trois éléments essentiels : L'infrastructure NFV (NFVI), les fonctions de réseau virtualisées (VNF) et la gestion et l'orchestration NFV (NFV MANO). Ces composants sont illustrés visuellement dans la Figure 9 et sont définis dans cette section[51], [53].

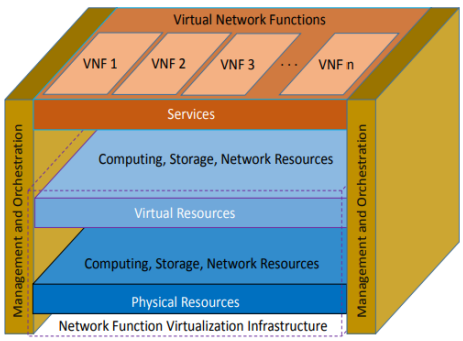

Figure 9 : L'architecture NFV[42]

NFV Infrastructure (NFVI) : Le NFVI est une combinaison de ressources matérielles et logicielles utilisées pour déployer les VNF. Les composants physiques tels que le matériel informatique COTS, le stockage et le réseau facilitent le traitement, le stockage et la connectivité des VNF. Les composants virtuels sont des abstractions de ressources informatiques, de stockage et de réseau créées à l'aide d'un hyperviseur pour découpler les ressources virtuelles des ressources physiques. Dans un centre de données, les ressources de calcul et de stockage sont représentées par des machines virtuelles, tandis que les réseaux virtuels utilisent des liens et des nœuds virtuels. Les nœuds virtuels fournissent une fonctionnalité d'hébergement ou de routage, et les liens virtuels sont des connexions logiques entre deux nœuds virtuels dont les propriétés changent

Virtual Network Functions and Services : Une fonction de réseau (NF) est un bloc fonctionnel avec des interfaces et un comportement défini, comme une passerelle résidentielle (Residential Gateway : RGW) ou un pare-feu. Une fonction de réseau virtualisée (VNF) est implimenté virtuellement sur des ressources virtuelles telles qu'une VMs, qui peut être composée de plusieurs éléments internes. Un service offert par le Telecom Service Provider (TSP) est composé d'une ou plusieurs NF, qui peuvent être virtualisées et déployées sur des VM dans le cas de la NFV. Cependant, du point de vue de l'utilisateur, les services, qu'ils fonctionnent sur des équipements dédiés ou sur des machines virtuelles, devraient avoir la même performance. Le comportement du service dépend des VNF qui le composent, lesquels sont déterminés par la spécification fonctionnelle et comportementale du service[53].

NFV Management and Orchestration (NFV MANO) : L'ETSI décrit le MANO comme étant la fonctionnalité requise pour l'approvisionnement et la gestion des fonctions de réseau virtualisées (VNF). Cela comprend l'orchestration et la gestion du cycle de vie des ressources physiques et/ou logicielles qui prennent en charge la virtualisation de l'infrastructure, ainsi que les bases de données pour le stockage des informations et des modèles de données. NFV MANO se concentre sur les tâches de gestion spécifiques à la virtualisation dans le cadre NFV et définit les interfaces pour la communication entre les différents composants, ainsi que la coordination avec les systèmes de gestion de réseau traditionnels tels que OSS et BSS. Cela permet de gérer à la fois les VNF et les équipements existants[51], [54].

L'architecture de référence NFV proposée par l'ETSI fournit des exigences fonctionnelles et décrit les interfaces requises, mais son champ d'application est limité et ne couvre pas le contrôle et la gestion des équipements existants. Il y a actuellement un manque de normes, de meilleures

pratiques et d'implémentations de référence pour les composants NFV. Les fournisseurs ont des idées divergentes sur la modélisation des NFVI et des VNF, et des questions restent en suspens concernant le déploiement des NF, les types de ressources nécessaires et les exigences opérationnelles. La deuxième phase des travaux de l'ETSI permettra de résoudre certains de ces problèmes, mais le temps presse, car les fournisseurs et les TSP investissent déjà beaucoup dans la NFV, ce qui pourrait donner lieu à des solutions spécifiques aux fournisseurs qui ne pourraient pas être inversées[51].

II.9.2.2. La relation entre cloud computing et la NFV

La NFV n'est pas limitée aux services de télécommunications et de nombreuses applications informatiques fonctionnent déjà sur des serveurs de base dans le cloud. Cependant, les fonctions de niveau transporteur ont des exigences de performance et de fiabilité plus élevées que les applications informatiques, ce qui fait que les performances acceptables de la NFV sont de niveau transporteur. Les modèles de services en cloud correspondent à une partie de l'architecture NFV, l'IaaS correspondant aux ressources physiques et virtuelles dans le NFVI, et les services et VNF dans le NFV étant similaires au modèle de service SaaS. La plupart des preuves de concept et des premières mises en œuvre de la NFV ont été basées sur le déploiement de fonctions sur des machines virtuelles dédiées dans le cloud en raison de sa flexibilité, du déploiement rapide de nouveaux services, de la facilité d'extensibilité et de la réduction de la duplication. Le cloud computing est le meilleur candidat pour atteindre l'efficacité et la réduction des dépenses qui motivent les TSPs vers la NFV.

Le déploiement de fonctions de réseau dans le cloud computing entraînera des changements importants dans le développement et la fourniture de services et d'applications. Bien que des progrès aient été réalisés dans les cloud en réseau et les réseaux inter cloud, les réseaux de télécommunications sont différents des environnements cloud computing pour trois raisons essentielles : les exigences de haute performance pour les charges de travail du plan de données, les exigences difficiles pour les topologies de réseau et la nécessité d'une gestion globale du réseau, ainsi que l'exigence d'évolutivité, de disponibilité cinq fois plus élevée et de fiabilité. Dans les réseaux de télécommunications traditionnels, ces caractéristiques sont fournies par l'infrastructure du site, mais si la NFV est basée sur le cloud computing, l'infrastructure cloud doit les reproduire d'une manière orchestrée qui peut être exposée par le biais d'abstractions appropriées, d'un support avancé pour la découvrabilité et la traçabilité, et d'un comportement de classe opérateur. Par conséquent, la NFV exige plus qu'un simple transfert de fonctions de réseau de classe opérateur

vers le CC ; elle nécessite également l'adaptation des environnements cloud afin d'obtenir un comportement de classe opérateur[51], [55]. La Figure 10 représente la comparaison entre les différentes couches des NFV et le CC.

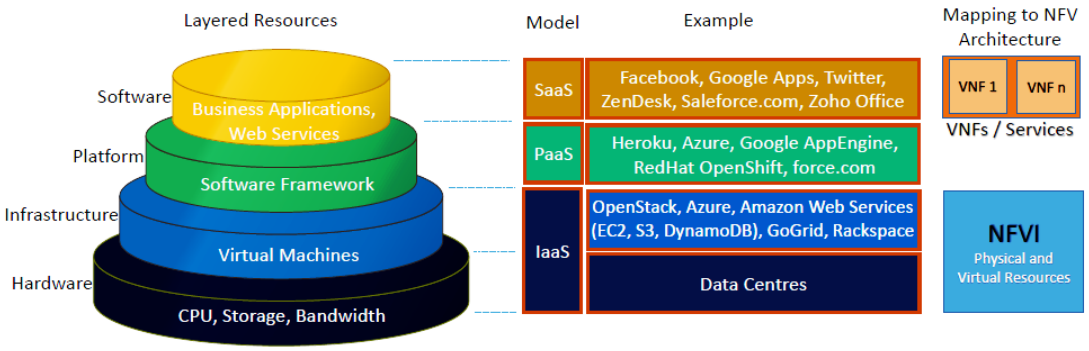

Figure 10 : Comparaison entre le NFV et le Cloud[42]

I.9.3. SDN/NFV

Le SDN et le NFV ont d'abord été développés indépendamment, mais ils ont des objectifs et des techniques communs, ce qui en fait des technologies complémentaires avec une forte synergie. Le SDN et le NFV peuvent tous deux améliorer leurs capacités respectives, ce qui se traduit par une flexibilité et une simplicité du réseau. Le SDN simplifie la gestion du réseau et accélère le déploiement du NFV en configurant les nœuds VNF par l'intermédiaire d'un contrôleur centralisé, tandis que le NFV peut mettre en œuvre un contrôleur SDN sur une machine virtuelle, ce qui permet aux contrôleurs SDN de tirer parti des caractéristiques du NFV telles que l'évolutivité et la fiabilité. Plusieurs études de recherche ont intégré NFV et SDN dans divers environnements, mais la complexité des composants du réseau et leurs interactions restent un défi pour les architectures SDN/NFV. Par conséquent, de nombreuses recherches universitaires et industrielles se concentrent sur la résolution des problèmes de fiabilité et de performance de l'architecture SDN/NFV[56].

I.10. La sécurité dans l'environnement Cloud

La sécurité des infrastructures du CC est une préoccupation majeure pour tous les intervenants impliqués dans leur gestion et leur utilisation. Pour garantir l'intégrité, la confidentialité et la disponibilité des données stockées dans le cloud, il est impératif de mettre en place des mesures de sécurité robustes et de suivre les meilleures pratiques en matière de gestion des risques. Dans le chapitre IV nous allons revenir sur la sécurité du CC pour donner beaucoup plus de détails.

I.11. Conclusion

Ce chapitre présente les fondamentaux d'un système du Cloud Computing. Le concept a été défini et ses principales caractéristiques ont été présentées. Le cloud computing offre une vaste gamme de services informatiques à la demande et payants sous la forme de logiciels, de plateformes ou d'infrastructures, répondant aux besoins spécifiques et aux attentes des différents utilisateurs. Ces services peuvent être déployés selon quatre modèles possibles, à savoir le cloud privé, le cloud public, le cloud communautaire et le cloud hybride.

Les utilisateurs finaux considèrent cette technologie comme une excellente option en raison de ses nombreux avantages. Cependant, malgré les divers avantages offerts par le cloud, il existe encore certaines limites. Comme nous allons le voir, il existe certains défis que doit relever cette technologie de pour améliorer la qualité des services fournis pour son adoption.

Dans le chapitre suivant, nous allons présenter les différentes technologies associées au cloud comme le Big Data pour l'analyse et le traitement des données ainsi que l'IoT utilisé souvent dans la collectée des données. Nous allons également voir certains vulnélabilité du cloud et comment associé la blockchain pour améliorer sa sécurisation.

CHAPITRE II : TECHNOLOGIES ASSOCIEES AU CLOUD: CONCEPTS DE BASE ET SECURITE

II.1. Résumé

Avec l'émergence de nouveaux éléments connectés tels que les villes intelligentes, les maisons intelligentes ont conduit à l'évolution remarquable de l'Internet des objets (IoT), permettant l'adoption de petits appareils pour gérer et effectuer des tâches complexes. Autour de ces équipements est apparue la collecte incessante d'une masse de données non structurées qui doivent parfois être analysées et stockées, ce qui a permis le développement de plusieurs applications pour un bon fonctionnement. Cela a favorisé l'émergissance et l'adoption du big data. L'intégration du cloud dans l'IoT est une nécessité pour tirer avantages de cette technologie. Le seul défi de l'utilisation du Cloud et de l'IoT est la sécurité des données et la disponibilité permanente des informations. La blockchain est aujourd'hui l'une des solutions prometteuses appliquées à divers problèmes de sécurité. Dans ce chapitre, nous présentons une revue de la littérature sur les différentes technologies associées au cloud tel que l'IoT et le Big data et l'utilisation de la Blockchain pour les sécurisées.

Mots-clés : Blockchain, IoT, Cloud Computing et Sécurité

II.2. Introduction

L'évolution de l'IoT en fait une technologie futuriste. Il permet à plusieurs objets différents situés dans le monde entier d'interagir les uns avec les autres dans un but parfois commun. Il s'agit d'un outil indispensable et très puissant qui permet de créer, de partager et d'analyser en temps réel d'innombrables informations. L'évolution remarquable de ces objets et le besoin émergent de leur utilisation montrent la croissance exponentielle de ces objets connectés, qui les pousse à communiquer directement avec les ordinateurs et à réduire l'interaction humaine. Dans[57], les auteurs ont montré que l'étude menée par l'institut Gartner, spécialisé dans la recherche sur les technologies de l'information, prévoyait que dès cette année, il y aurait déjà 50 milliards d'objets connectés sur le marché. L'IoT est donc une technologie qui révolutionner le monde numérique et changer radicalement nos modes de vie. Malgré cette évolution, l'IoT est un gadget mais apporte beaucoup plus de valeur à ses utilisateurs dans plusieurs domaines tels que la santé, les transports, l'industrie, etc. A l'heure actuelle, la profusion de l'utilisation de l'IoT est inexorable, il faut souligner qu'elle est consubstantielle à d'autres technologies qui révolutionnent le monde informatique, notamment le CC. Le CC n'est pas nouvelle, mais c'est un sujet qui offre des

possibilités illimitées et qui revêt une grande importance pour les chercheurs du monde entier. Basée sur la virtualisation, le cloud computing consiste à fournir des ressources informatiques à distance via un réseau, généralement l'internet. Il fournit des unités de calcul et de stockage de données à la demande et en libre-service. Le Cloud Computing offre plusieurs avantages de telle sorte qu'il est irréfutable et vital pour les autres technologies.

La croissance de ces objets connectés implique une grande masse de données, nécessitant un espace de stockage et d'analyse important. C'est à ce moment que l'IoT se tourne vers le Cloud Computing pour profiter de ces avantages et devenir inextricablement lié et inséparable pour alléger la pression que l'IoT fait peser sur l'infrastructure. Malgré le succès incontestable de ces technologies, le problème lié à la sécurité est encore loin d'être résolu. Pour le déploiement et l'usage de ces technologies doit prendre en considération l'aspect sécurité. Plusieurs mécanismes de sécurité peuvent être appliqués pour sécuriser les différents niveaux d'abstraction des deux technologies précitées. La nouvelle tendance en matière de sécurité des systèmes d'information est l'utilisation de la Blockchain. Cette dernière est un registre public qui repose sur des fonctions de hachage et ne peut être modifiée par un ordinateur. Une blockchain fonctionne sur un réseau peer-to-peer (P2P). Tous les nœuds participants exécutent un protocole pour parvenir à un consensus sur l'état de la chaîne de blocage. L'état fait référence au contenu, à l'exactitude et à l'ordre des transactions stockées dans la blockchain[58].

Dans ce chapitre, nous présentons l'état de l'art de la sécurité dans le Cloud et l'IoT avec l'utilisation de la Blockchain. Le reste du document est organisé comme suit. Dans la section II, nous présentons le concept de base des différentes technologies qui feront l'objet de notre étude. Dans la section III, nous passons en revue les différents travaux déjà réalisés sur l'utilisation de la Blockchain comme mécanisme de sécurité dans le Cloud et l'IoT. Dans la section IV, nous conclurons ce travail.

II.3. Concept de base

Avec l'omniprésence des NTIC dans nos activités, il n'est pas toujours facile de reconnaître ce que recouvrent ou non les technologies susmentionnées malgré les définitions données. Dans cette partie du travail, nous présentons les concepts de base de ces différentes technologies (IoT, la Blockchain et le Big datag) afin d'éclairer les utilisateurs avertis et non avertis sur le principe de fonctionnement de ces technologies. Pour le cas du cloud computing, ses concepts ont été présentés dans le chapitre II.

II.3.1. L'internet des objets

Dans l'ombre de différentes technologies, l'IoT continue d'étendre ses tentacules dans le monde de l'informatique. Il se distingue par sa capacité à toucher plusieurs domaines d'application et son efficacité à remplacer les ordinateurs qui jusque-là accomplissent toutes les tâches. Autrement dit, l'internet des objets est cette façon d'interagir avec des appareils mobiles interconnectés entre eux et entre des entités physiques via un réseau très étendu, générant une masse de données à l'aide de capteurs. Ces données sont ensuite récupérées, stockées et analysées.

Les appareils IoT sont identifiables de manière unique et se caractérisent principalement par une faible puissance, une petite mémoire et une capacité de traitement limitée. Des passerelles sont déployées pour connecter les dispositifs IoT au monde extérieur afin de fournir à distance des données et des services aux utilisateurs de l'IoT.

II.3.1.1. Domaines d'application de l'IoT

Comme nous l'avons mentionné, l'internet des objets a plusieurs domaines d'application. La Figure 11 en présente quelques-uns. Il touche une grande partie des différents domaines de notre vie. En raison des changements drastiques dans tous les aspects de la société, l'IoT est considéré comme la quatrième révolution industrielle. Cette tendance actuelle de l'industrie 4.0 favorise l'adoption de la maintenance prédictive, qui permet de détecter des anomalies sans intervention humaine et notamment grâce aux données collectées par des capteurs appelant l'IoT. D'autres applications sont abordées dans les publications, mais nous parlons ici quelques une qui sont représentées dans la Figure 11. En outre, nous avons distingué des catégories d'applications de l'IoT dans l'industrie médicale et l'environnement domestique intelligent, le transport intelligent et l'environnement intelligent, comme la ville intelligente, l'achat intelligent et la fabrication de l'IoT. Dans toutes ces applications diverses, l'IoT est essentiellement appliqué avec plusieurs capteurs, actionneurs, instruments d'analyse et de visualisation des données qui permettent à l'IoT d'être disponible à tout moment, n'importe où et pour n'importe quel objet. Nous présentons quelques domaines d'applications de l'IoT :

A. L'IoT au service de l'agriculture.

L'agriculture est en train de se démocratiser grâce à l'IoT. Pour optimiser la gestion des exploitations et les récoltes, le secteur s'équipe de nouveaux outils[59]. Ce phénomène est

influencé par certains facteurs tels que le réchauffement climatique et la croissance démographique. Des travaux ont touché le domaine pour y remédier[60], [61].

B. L'IoT pour les maisons intelligentes.

La domotique est la technologie qui permet d'automatiser certaines actions, comme l'allumage des lumières ou du chauffage, mais qui nécessite une intervention humaine pour assurer le confort des occupants. C'est ce que l'on appelle aussi la maison intelligente. Le développement de l'IoT a rendu ces progrès possibles. Beaucoup a été fait ces dernières années pour faciliter la gestion des maisons intelligentes[62].

C. L'IoT dans les soins de santé.

Alors que divers secteurs ont adopté l'IoT, le secteur des soins de santé rime avec IoT. Grâce à l'IoT, il est possible de surveiller, d'informer et de notifier les soignants ou les compagnies d'assurance de l'état de santé d'un patient. Avec l'IoT, il est facile de collecter différentes données à partir de différents tests (contrôle de la pression artérielle, de la glycémie, ...). Plusieurs travaux ont porté sur les applications de l'IoT dans le domaine de la santé[63].

D. L'IoT dans le sport

Aujourd'hui, l'IoT est utilisé dans le domaine du sport pour améliorer les performances. Aujourd'hui, cette technologie permet de collecter des données pour prédire les performances d'un athlète. Elle peut également signaler un but lors d'un match de football lorsque le ballon a franchi la ligne de but. L'IoT devient indispensable dans ce domaine et plusieurs travaux ont été réalisés sur son application.

E. L'IoT dans les transports

L'IoT révolutionne les transports dans le domaine de la gestion du trafic et de la sécurité routière. Ce domaine couvre plusieurs cas d'utilisation. Par exemple, il peut être utilisé pour réduire les coûts de maintenance grâce à une maintenance réactive[64].

F. L'IoT dans l'industrie

L'IoT révolutionne le monde industriel et en particulier l'industrie manufacturière. L'enjeu est d'améliorer la productivité, d'apporter plus de fiabilité en assurant la reproductibilité et la qualité

des produits, et d'optimiser les processus de production. La surveillance continue des équipements facilite également la maintenance et le contrôle de leur fonctionnement[64], [65].

Figure 11 : Quelques domaines d'applications de l'IoT[66]

II.3.1.2. Communication des capteurs

Comme indiqué dans les définitions, l'IoT est rendu possible par des capteurs qui collectent des informations en fonction du domaine d'application. Ces capteurs doivent également communiquer et échanger des informations. En se focalisant sur chaque domaine d'application, des facteurs tels que le champ d'application, les questions de sécurité et d'alimentation, les exigences en matière de données et les questions d'alimentation et de durée de vie de la batterie influenceront le choix de la technologie à utiliser ou l'adoption d'une combinaison de technologies. Le Tableau 1 montre la diversité de certains protocoles utilisés dans la communication IoT.

Type	**Standard**	**Portée**	**Vitesse de transmission**	**Fréquence**
Bluetooth	Bluetooth 4.2	50-150m	1Mbits/s	2,4GHz
ZigBee	ZigBee 3.0 based on IEEE802.15.4	10-100m	250Kbit/sZ-Wave	2,4GHz
Z-Wave	Z-Wave alliance ZAD12837/ ITU-T G.9959	30m	9,6/40/100Kbits/s	900MHz
6LowPAN	RFC6282	Variable	N/A	N/A

Thread	Thread, basé sur IEEE802.15.4 et 6LowPAN	N/A	N/A	2,4GHz
WiFi	802.11n	~50m	500Mbit/s à 1Gbit/s	
Technologie Cellulaire	GSM/GPRS/ EDGE, UMTS/ HSPA,LTE	35Km à + 200Km	30-170Kbit/s (GPRS), 120-384Kbit/s (EDGE), 384Kbit/s – 2Mbit/s (EDGE), 120-384Kbit/s (EDGE), 600 Kbit/s-10 Mbit/s (HSPA), 3-10 Mbit/s (LTE)	900/1800/1900 /2100MHz
NFC	ISO/CEI18000-3	10cm	100–420 Kbit/s	13,56MHz
Sigfox	Sigfox	0-50km (rural location), 3-10km(metropolitan area)	10-1 000 bit/s	900 MHz
Neul	Neul	10 km	Some bit/s to 100 Kbit/s	900 MHz (ISM), 458 MHz (UK), 470-790 MHz (White Space)
LoRaWAN	LoRaWAN	2-5km (Zone rurale), 15 km(zone suburbaine)	0,3-50 Kbit/s	Variable

Tableau 1 : Protocoles de communications des équipements IoT

II.3.3. Blockchain

Au sens strict, une blockchain est une structure de données en chaîne qui combine des blocs de données et d'informations dans l'ordre chronologique et stocke les blocs sous forme cryptée dans un grand livre distribué qui ne peut être falsifié ou contrefait. En général, la technologie Blockchain utilise des structures de données de type bloc pour valider et stocker les données et des algorithmes de consensus de nœuds distribués pour générer et mettre à jour les données et utilise également le cryptage pour assurer la transmission des données et la sécurité de l'accès. Le

code de script contient des contrats intelligents pour la programmation et la manipulation des données dans de nouvelles infrastructures distribuées et de nouveaux modèles informatiques[63]. Il est utilisé dans divers domaines tels que les services financiers[67], la transformation des entreprises[68], le marché financier P2P[69] et la gestion des risques[70]. Comme dans notre cas, ces contrats intelligents peuvent également être utilisés dans l'IoT et la sécurité. Ils peuvent être utilisés dans les services publics et sociaux qui sont divisés en cadastre, économie d'énergie[71], éducation[72], et le droit à la liberté d'expression. Enfin, ces technologies sont utilisées dans les systèmes de réputation qui sont regroupés dans les services académiques[73] et les services web[74]. Certains chercheurs ont résumé et illustré ces différents domaines dans la Figure 12[75].

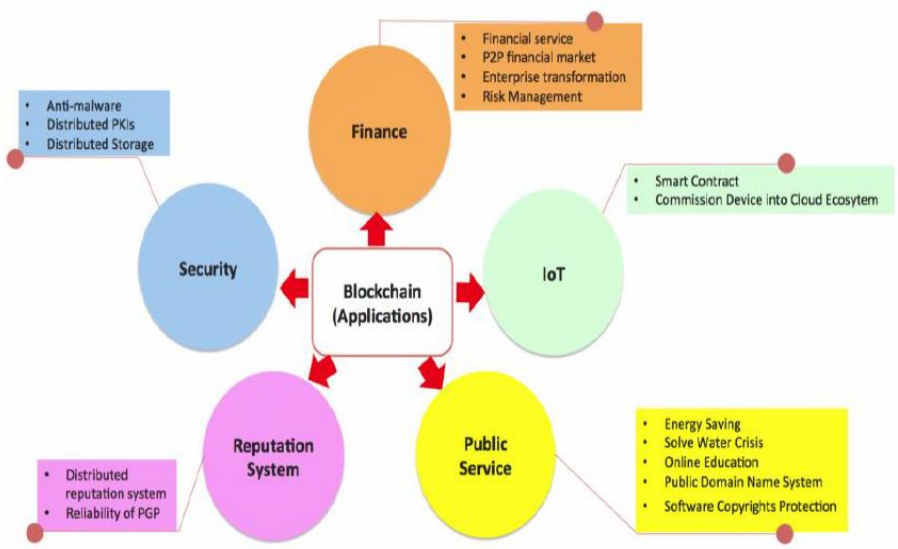

Figure 12 : Domaine d'application de la Blockchain[76]

II.3.3.1. Structure de base de la blockchain

Malgré le succès irréfutable de l'utilisation de la Blockchain dans divers secteurs socio-économiques, il ne s'agit pas d'une technologie inventée de toutes pièces, mais d'une technologie qui découle de celles qui existent déjà. Dans cette partie du chapitre, nous présentons les différentes couches qui caractérisent et normalisent l'architecture typique et les principaux composants des systèmes de blockchain et nous discutons des techniques clés de chaque couche.

La blockchain se compose de deux éléments principaux, à savoir les transactions, qui sont les actions générées par les participants au système, et les blocs, qui enregistrent les transactions et

garantissent qu'elles sont dans le bon ordre et qu'elles n'ont pas été altérées. Au-delà de cette structure, plusieurs composants sont intégrés dans différentes couches constituant la blockchain, comme le montre la Figure 13[77].

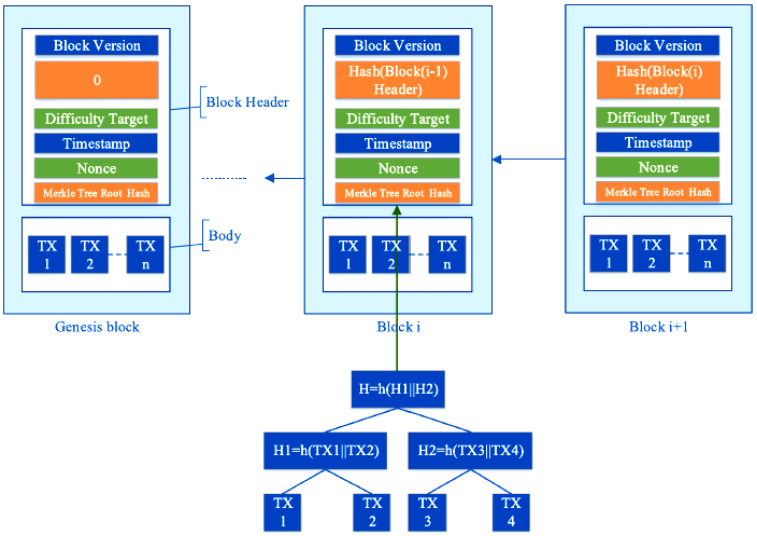

Figure 13 : Structure de la blockchain[77]

Couche de données - Cette couche fournit les blocs de données enchaînés, ainsi que les techniques connexes, notamment le cryptage, l'horodatage, les algorithmes de hachage et les arbres de Merkle[77]. Elle traite les données provenant de sources multiples.

Couche réseau - Cette couche spécifie des modèles de communication décentralisés et des mécanismes connexes pour la mise en réseau distribués, la transmission de données et la vérification. La blockchain est basée sur un environnement décentralisé et modélisée par la topologie du réseau P2P. Tous les nœuds du réseau ont les mêmes privilèges dans la blockchain, il n'y a pas d'autorité centrale.

Couche de consensus - La blockchain est un groupe de plusieurs algorithmes de consensus. Cette couche contiendra ces algorithmes pour assurer la cohérence et la tolérance aux pannes du registre partagé entre les nœuds distribués.

Couche d'application - C'est la couche qui contient les applications, les scénarios et les cas d'utilisation.

Couche contrat - Il s'agit d'une couche qui rassemble divers scripts d'algorithmes ainsi que des contrats intelligents qui s'auto-vérifient et s'auto-exécutent et qui sont stockés et sécurisés par la blockchain.

Couche réseau - Cette couche spécifie les modèles de communication décentralisés et les mécanismes de réseau distribué correspondants. La blockchain est basée sur un environnement décentralisé et modélisé par la topologie du réseau P2P. Tous les nœuds du réseau ont les mêmes privilèges dans la blockchain, il n'y a pas d'autorité centrale.

La blockchain présente plusieurs caractéristiques qui la rendent plus attrayante que les autres technologies.

Immutabilité - La construction de livres immuables est l'une des valeurs clés de la blockchain. Toutes les bases de données centralisées peuvent être corrompues et nécessitent généralement de faire confiance à un tiers pour préserver l'intégrité des informations. Une fois que vous vous êtes mis d'accord sur une transaction et que vous l'avez enregistrée, elle ne peut plus être modifiée[62].

Décentralisation - La blockchain est constituée d'un ensemble de blocs connectés dans un réseau P2P et peut enregistrer différentes transactions. Contrairement au système centralisé, chaque bloc participe à la prise de décision des différentes activités de la transaction.

La blockchain, en tant que système indépendant et autonome, permet à chaque nœud d'accéder, de transférer, de stocker et d'éffectuer des mises à jour des données en toute sécurité, ce qui représente sa plus grande résistance à toute intervention extérieure.

Capacité accrue - Un aspect important de la technologie Blockchain est qu'elle peut accroître la capacité d'un réseau entier. Le fait que des milliers d'ordinateurs travaillent ensemble peut être plus puissant que quelques serveurs centralisés[62].

Transparent - Une caractéristique essentielle de la blockchain est qu'elle permet à tous les participants de visualiser les blocs et les transactions qu'ils contiennent. Les données enregistrées dans la blockchain sont transparentes pour les utilisateurs et peuvent être mises à jour facilement. Cependant, cela ne signifie pas que le contenu des transactions est accessible à tous, car elles sont protégées par une clé privée. Chaque nœud de la même plateforme a les mêmes permissions et

obligations d'accès aux informations autorisées et permet aux autres nœuds du même réseau d'accéder à ces informations.

Anonymat - L'anonymat est un moyen efficace de dissimuler l'identité des utilisateurs et de préserver la confidentialité de leur identité.

Traçabilité et infalsifiabilité - La blockchain utilise des horodatages pour identifier et enregistrer chaque transaction ayant lieu dans la blockchain, renforçant ainsi la dimension temporelle des données. Cela permet au nœud de conserver l'ordre des transactions et de rendre les données traçables. Chaque transaction validée sera enregistrée dans chaque bloc et ne pourra jamais être répétée[78].

II.3.3.2. Différents types de la Blockchain

Nous illustrons par la Figure 14 les différentes types de la technologie blockchain jusque là existant. Par la suite, nous diffinissons cacune de ces types pour avoir une vue globale sur leurs utilisations.

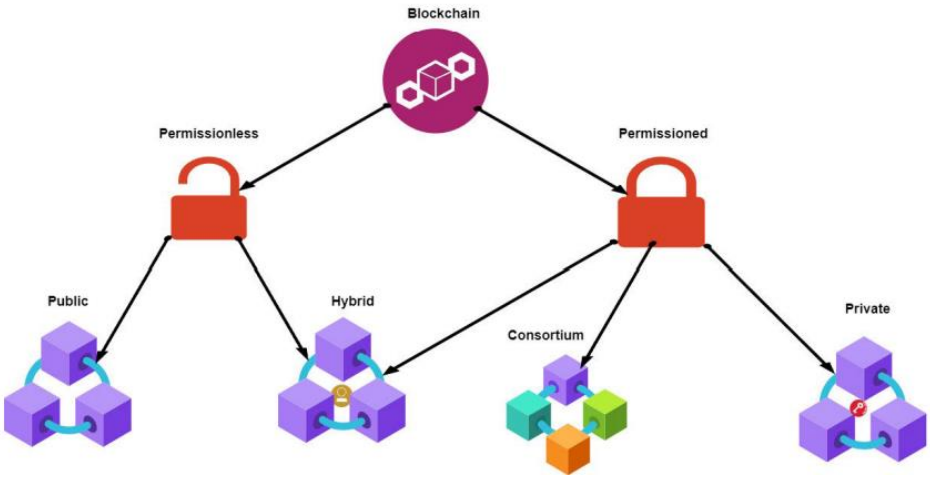

Figure 14 : Différents types de la blockchain[78]

Blockchain publique - Ce type de blockchain est ouvert au public et peut être accédé via un ordinateur et une connexion Internet. Cependant, certaines de ces blockchains restreignent l'accès à la lecture ou à l'écriture. La plupart sont développées et maintenues par des communautés de développeurs indépendants en Open Source, avec pour objectif d'éliminer complètement les intermédiaires de confiance traditionnels.

Blockchain privée - La blockchain privée est contrôlée par une ou un groupe d'entités. Pour accéder à ce type de Blockchain, une autorisation est délivrée par l'organisation qui l'a déployée. L'un des grands avantages des blockchains privées est qu'elles ne nécessitent pas d'argent réel pour stocker et accéder aux données. Cependant, elles ne sont pas entièrement dignes de confiance. L'entité qui contrôle sa gouvernance et ses opérations peut conserver un pouvoir supérieur pour tempérer les données[79].

Blockchain hybride- De nombreuses organisations cherchent à exploiter à la fois la technologie de la blockchain publique et celle de la blockchain privée. Pour ce faire, elles adoptent la blockchain hybride, une approche qui fusionne des aspects des deux technologies. En établissant un système de permission privé en parallèle avec un système public sans permission, une organisation peut exercer un contrôle sur les données de la blockchain accessibles au public et sur les entités autorisées à y accéder. Contrairement à une blockchain entièrement publique, une blockchain hybride ne rend généralement pas toutes les transactions et les enregistrements visibles à tous, mais elle peut être vérifiée au besoin via des contrats intelligents. La confidentialité des informations au sein du réseau peut toujours être vérifiée. Bien qu'une entité privée puisse être propriétaire de la blockchain hybride, elle ne peut pas altérer les transactions enregistrées. Lorsqu'un utilisateur rejoint une blockchain hybride, il bénéficie d'un accès complet au réseau. Sauf si d'autres utilisateurs s'engagent dans une transaction avec lui, son identité reste protégée, bien que dans ce cas, l'identité des autres parties soit révélée[80].

Blockchain de consortium - La blockchain de consortium, ou chaîne de blocs fédérée, est similaire aux chaînes de blocs privées dans la mesure où il s'agit d'un réseau autorisé. Les réseaux de consortiums rassemblent plusieurs organisations et contribuent à maintenir la transparence entre les parties concernées. Une blockchain consortium est utilisée comme une base de données distribuée, vérifiable et synchronisée de manière fiable qui suit les échanges de données entre les membres du consortium participants. À l'instar des blockchains privées, une chaîne de consortium n'implique pas de traitement et la publication de nouveaux blocs et n'est pas coûteuse en termes de calcul. Bien qu'elle offre une contrôlabilité et une latence réduite dans le traitement des transactions, elle n'est pas entièrement décentralisée ni résistante à la censure[68], [81].

Le Tableau 2 présente différents avantages et désavantages des différents types de blockchain :

Types	Avantages	Désavantages	Applications
Publique	• Indépendants • Transparence • Confiance	• Performances • Évolutivité • Sécurité	• Crypto-monnaie • Validation des documents
Privée	• Contrôle d'accès • Performance de l'entreprise	• Confiance • Auditabilité	• Chaîne d'approvisionnement • Propriété des actifs
Hybride	• Contrôle d'accès • Performance de l'entreprise • Évolutivité	• Transparence • Mise à niveau	• Dossiers médicaux • Immobilier
Consortium	• Contrôle d'accès • Sécurité • Évolutivité	Transparence	• Banque • Recherche • Chaîne d'approvisionnement

Tableau 2 : Avantages et inconvénientss de s différents tpes de la Blockchain

II.3.3.3. Contrats intelligents

Les contrats intelligents sont des protocoles informatiques qui facilitent, exécutent et vérifient un accord de volonté entre deux personnes (ou organisations) pour créer une obligation légale. Aujourd'hui, les contrats intelligents sont l'élément clé de la blockchain. Les clauses contractuelles sont converties en logiciel informatique exécutable. L'exécution de chaque déclaration de contrat est enregistrée comme une transaction immuable stockée dans la Blockchain. La Figure 15[82] illustre le cycle de vie d'un contrat intelligent dans la blockchain, qui se compose de quatre phases consécutives.

Création de contrats intelligents - Cette étape implique d'abord une négociation sur les obligations, les droits et les interdictions des contrats. Un accord est conclu après plusieurs cycles de négociation. Des avocats aideront les différentes parties à rédiger un premier accord contractuel. Les ingénieurs logiciels convertissent ensuite cet accord rédigé en langage naturel en un contrat intelligent écrit dans un programme informatique exécutable[76].

Déploiement des contrats intelligents - Une fois les contrats intelligents validés, ils peuvent être déployés dans la Blockchain. Les contrats stockés dans la blockchain ne peuvent pas être modifiés en raison de l'immuabilité de la blockchain. En outre, les actifs numériques des deux parties impliquées dans le contrat intelligent sont verrouillés en gelant les portefeuilles numériques correspondants[76].

Exécution des contrats intelligents - Après les deux premières étapes du cycle de vie d'un contrat intelligent et après vérification des conditions contractuelles atteintes, les procédures (ou fonctions) du contrat seront automatiquement exécutées. Lorsqu'une condition est déclenchée, la déclaration correspondante est automatiquement exécutée, c'est-à-dire qu'une transaction est exécutée et validée par les mineurs dans la blockchain[83].

Achèvement des contrats intelligents - Toutes les parties concernées sont mises à jour après l'achèvement du contrat intelligent. À ce moment-là, toutes les transactions effectuées pendant l'exécution des contrats intelligents ainsi que les mises à jour sont stockées dans la blockchain (par exemple, le transfert d'argent de l'acheteur au fournisseur).

Malgré certains défis, l'utilisation de contrats intelligents dans la blockchain est inéluctable. L'utilisation de ces contrats intelligents a permis de révolutionner les différentes applications réelles des solutions Blockchain comme la finance, qui comprend les services financiers[84], la transformation des entreprises[67], le marché financier P2P, et la gestion des risques[69]. Comme dans notre cas, ces contrats intelligents peuvent également être appliqués à l'IoT et à la sécurité. Ils peuvent être utilisés dans les services publics et sociaux, subdivisés en cadastre, économie d'énergie[70], éducation[71], et droits d'expression. Enfin, ces technologies sont utilisées dans le système de réputation, qui est regroupé en académique[72] et service web[73]. Ajoutons à ces solutions le vote, les soins de santé, la sécurité alimentaire et l'immobilier, pour n'en citer que quelques-unes.

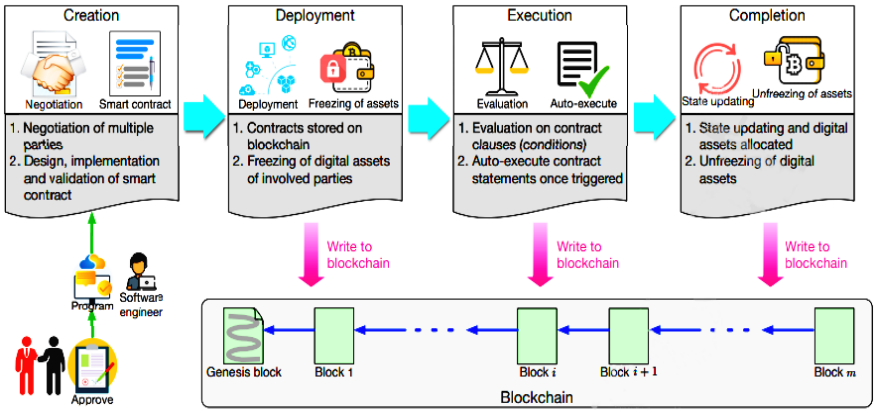

Figure 15 : Cycle de vie d'un contrat intelligent dans la Blocchain[76]

L'origine de la Blockchain souvent associée à son origine qui est la crypto monnaie, freine sa démocratisation. Il est cependant nécessaire de s'intéresser à son impact potentiel vers d'autres cas d'usage, notamment dans la sécurisation des transactions vers d'autres cas d'usage. Dans notre cas, nous nous concentrerons sur l'utilisation de la blockchain pour sécuriser ces technologies indissociable du Cloud Computing.

II.3.4. Le Big Data

La prolifération exponentielle des données générées et stockées constitue un phénomène remarquable dans le paysage contemporain. Cette expansion massive engendre à la fois des opportunités inédites pour l'exploitation innovante des données en vue de générer de la valeur ajoutée, ainsi que des défis substantiels du fait de la gestion et de l'analyse de volumes de données considérables. Un aspect crucial de cette évolution réside dans la prépondérance croissante des données non structurées. Traditionnellement, les analyses commerciales se concentraient principalement sur les données structurées, gérées au moyen de modèles relationnels. Cependant, ces dernières années, la quantité de données non structurées, telles que des fragments textuels, des pages web, des données multimédias et des flux de relations, a connu une expansion fulgurante, marquant une tendance à l'intégration croissante de ces données dans le processus de création de valeur[85].

L'avantage majeur de l'analyse Big Data réside dans sa capacité à traiter des volumes massifs et une variété de données de natures diverses. Il importe de souligner que le Big Data ne se réduit pas simplement à une augmentation quantitative des volumes de données, ni à une taille dépassant les capacités des méthodes d'analyse traditionnelles. La nécessité d'améliorations constantes en termes de performances et d'efficacité demeure une exigence perpétuelle. Toutefois, le Big Data

représente un changement fondamental dans l'architecture requise pour la gestion efficiente des ensembles de données contemporains[86].

II.3.4.1. Les caractéristiques et architecture du Big Data

Une compréhension approfondie de la vaste étendue des données s'avère indispensable. Ces caractéristiques sont essentielles pour appréhender les mégadonnées, ainsi que pour élaborer des stratégies efficaces de traitement de volumes massifs et hétérogènes de données dans des délais appropriés. Ceci permet l'extraction de valeurs et la réalisation d'analyses en temps réel. Au départ, le Big Data était caractériser par 3V qui on ont été cités dans différentes définitions. La Figure 16 montre qu'aujordhui le Big Data peut se caractériser par plus de de 3V, dans notre cas nous présentons 8V.

La connaissance des attributs du Big Data permet une meilleure appréhension de ses cas d'utilisation spécifiques et de ses applications. Ainsi, explorons les aspects critiques de l'analyse du Big Data[87], [88]:

Figure 16 : Les 8V du Big Data[87]

Volume : Dans le contexte actuel, les entreprises font face à une quantité considérable de données. Pour analyser ces données massives, il est nécessaire de traiter des volumes substantiels de données structurées et non structurées. La source de ces données est surtout variées telles que les ensembles de données de réseaux sociaux (Facebook et Instagram par exemple), ou encore des données issues de nombreuses applications web ou mobiles. Selon les tendances du marché, il est prévu que le volume de données augmente de manière significative dans les années à venir, offrant ainsi de vastes opportunités pour une analyse approfondie et la découverte de modèles.

Vélocité : Elle désigne la rapidité avec laquelle les données sont traitées. Un haut débit de traitement des données est essentiel pour garantir l'évaluation et les performances en temps réel de tout processus Big Data. Alors que davantage de données seront disponibles à l'avenir, la vitesse de traitement demeurera tout aussi cruciale pour permettre aux entreprises de tirer profit de l'analyse des mégadonnées.

Variété : Les données variés font référence à la diversité des catégories de mégadonnées. Ce défi majeur auquel fait face l'industrie du Big Data a un impact significatif sur sa productivité. Avec l'essor croissant des mégadonnées, de nouveaux types de données émergent. Ces différentes catégories de données, comprenant notamment le texte, l'audio et la vidéo, nécessitent un prétraitement supplémentaire afin de garantir la préservation des métadonnées et d'en exploiter pleinement la valeur ajoutée.

Valeur : Elle représente les bénéfices que votre entreprise tire du traitement et de l'analyse des données. Elle mesure dans quelle mesure les données répondent aux objectifs définis par votre entreprise et contribuent à son amélioration. Il s'agit là d'une caractéristique fondamentale au sein du domaine du Big Data.

Véracité : Désigne l'exactitude de vos données. Cet aspect revêt une importance capitale car une faible véracité peut compromettre la précision de vos résultats d'analyse de données massives.

Validité : Elle renvoie à l'efficacité et à la pertinence des données pour les objectifs et les buts définis par une entreprise. C'est un critère essentiel pour évaluer la qualité et l'utilité des données à exploiter.

Volatilité : Les mégadonnées sont sujettes à des variations continues. Les données recueillies à partir d'une source précise peuvent évoluer en peu de temps, ce qui engendre des incohérences et affecte la capacité d'hébergement et d'adaptation des données.

Visualisation : La visualisation des données consiste à présenter les analyses et les informations produites par le Big Data à travers des représentations visuelles telles que des tableaux et des graphiques. Ce procédé revêt désormais une importance majeure, car il permet aux experts en Big Data de communiquer leurs analyses et leurs idées à des destinataires non techniques.

L'architecture Big Data qui est présentée dans la Figure 17, constitue un cadre définissant les composants, les processus et les technologies requis pour capturer, stocker, traiter et analyser les données massives. Typiquement, elle se compose de quatre strates principales : la collecte et l'ingestion des données, le traitement et l'analyse des données, la visualisation et le reporting des

données, ainsi que la gouvernance et la sécurité des données. Chaque strate possède son propre ensemble de technologies, d'outils et de procédures[89].

Les avantages d'une architecture reposant sur Hive dans le domaine du Big Data incluent la capacité à prendre des décisions meilleures et plus rapides, à traiter et analyser davantage de données, et à améliorer l'efficacité opérationnelle. Toutefois, les défis inhérents à une architecture de type stack pour le Big Data comprennent le besoin de compétences et de connaissances spécialisées, des investissements coûteux en matériel et logiciel, ainsi qu'un niveau élevé de sécurité.

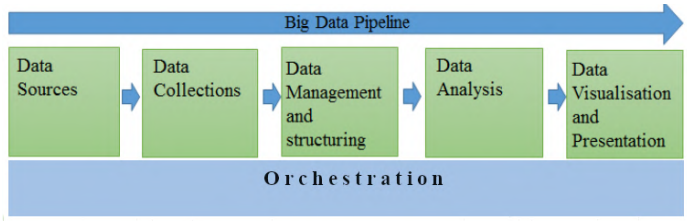

Figure 17 : Architecture du Big Data[90]

Dans le sous section suivante, nous allons présenter les différents défis du Big Data. Se basant sur différents travaux, on constate que les différents défis se base sur l'architecture du Big Data[91].

II.3.4.2. Les enjeux et défis liés aux Big Data

Alors que le big data s'impose progressivement dans le milieu entrepreneurial, les professionnel de l'IT ainsi que les cadres dirigeants seront confrontés à une série de défis préalables à surmonter pour assurer la réussite d'un projet de big data. Dans le contexte de la gestion des données, plusieurs défis émergent, nécessitant une attention particulière pour garantir le succès de l'adoption du big data[92]:

Premièrement, l'incertitude règne quant au choix des technologies appropriées parmi une multitude de solutions concurrentes, dans un paysage où chaque domaine technique compte de nombreux acteurs. Il est crucial de faire des choix éclairés afin de minimiser les risques potentiels inhérents à l'intégration du big data. Ensuite, la transparence joue un rôle essentiel. Il est primordial de rendre les données accessibles aux parties prenantes concernées dans des délais appropriés, favorisant ainsi une prise de décision éclairée et opportune.

L'expérimentation constitue également un aspect crucial. En permettant des tests et des analyses approfondis, les organisations peuvent identifier les besoins, mettre en lumière la variabilité des

données et améliorer les performances de manière significative. Avec la transition croissante vers la numérisation des données transactionnelles, les entreprises ont accès à des données de performance plus précises et détaillées. Par ailleurs, le recours à des algorithmes automatisés pour appuyer voire remplacer la prise de décision humaine peut non seulement minimiser les risques, mais aussi révéler des informations précieuses qui seraient autrement restées cachées.

Enfin, le big data ouvre la voie à l'innovation. Les entreprises peuvent ainsi concevoir de nouveaux produits et services, améliorer ceux déjà existants, voire réinventer leurs modèles d'affaires répondant ainsi à la forte demande du marché en constante évolution.

II.4. Utilisation de la blockchain pour sécuriser le cloud et l'IoT

La sécurité est le plus grand défi qui secoue le monde de l'IT, en particulier dans les domaines du CC et de l'IoT. En termes de sécurité des données, les enjeux sont élevés pour les données sensibles et critiques en transit et stockées dans le Cloud. Toute conception de la sécurité doit prendre en compte trois exigences de sécurité principales, à savoir la CIA[74]. La blockchain pourrait renforcer la sécurité tout en limitant les attaques potentielles provenant de l'extérieur des systèmes d'informatique dématérialisée basés sur des appareils utilisant l'IoT. Dans cette partie du travail, nous analysons les travaux liés à la mise en œuvre de la chaîne de blocs sécurisée de l'IoT et du cloud computing.

II.4.1. Sécurité de l'IoT avec la blockchain

L'IoT lui-même possède des caractéristiques qui le poussent à de bonnes performances, tout comme la blockchain. La combinaison de ces deux technologies procure plusieurs avantages, pour la sécurisation de l'IoT. Dans les sections précédentes, nous avons présenté le concept de ces deux technologies qui empêcheront la défaillance d'un seul nœud d'un réseau.

La blockchain pourrait renforcer la sécurité tout en limitant certaines attaques potentielles provenant de l'extérieur des dispositifs IoT et de la gestion des données qui y sont collectées. La sécurité de l'IoT doit se concentrer sur cinq axes différents, comme le montre la Figure 18[93].

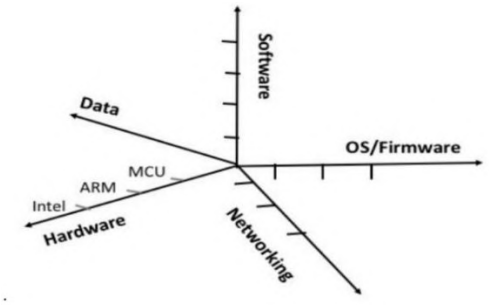

Figure 18 :Différents axe pour sécuriser l'IoT[94]

II.4.1.1. Les attaques dans l'IoT

La sécurité dans l'IoT peut être assurée à plusieurs niveaux. Dans[75], [95], les auteurs ont montré la vulnérabilité des dispositifs utilisés dans l'IoT et ont proposé des mécanismes de sécurité relatifs. Ces auteurs ont montré que la sécurité dans l'IoT doit être abordée sur cinq axes différents (logiciel, OS/firmware, réseau, matériel et données).

Plusieurs travaux ont montré et détaillé les différentes attaques qui peuvent être menées contre la technologie de l'IoT en tant que telle, selon les principes susmentionnés. Nous résumons certains de ces travaux dans le Tableau 3, tout en détaillant les attaques qui peuvent être menées dans chaque domaine et en fonction des couches de l'IoT. Dans ce cas, nous fusionnons à la fois l'OS et le logiciel dans un seul domaine qui est le soft. Dans ce même tableau, nous montrons également les contre-mesures qui peuvent être appliquées à chaque couche impliquée dans l'IoT.

	Attaques physiques	**Attaques sur le réseau**	**Attaques sur le Soft**	**Attaques sur les données**	**Contre-mesure**
Couche perception	Dégradation[94] Code malveillant[93] Interférence RF/brouillage[65] Injection de faux nœuds[96] Attaques par Sleep Denial [65] Attaque par le Side channel [93] DoS permanent [65]	-	-	-	Conception physique sécurisée Amorçage sécurisé IDT Authentification Canal de sécurité IPSec

Couche Application	Attaque par le Side channel [93]	-	Virus, Worms, Trojan Horses, Spyware and Adware[94]	-	Authentification biométrique Logiciel de protection ACL Sensibilisation à la sécurité Fonctions de hachage cryptographiques Firewalls
Couche réseaux	-	Attaque par analyse de trafic[97] Spoofing RFID et accès non autorisé[99] Attaque des informations de routage[94] Transfert sélectif[100] Attaque par Sinkhole [101] Replay Attack[100] Man in Middle Attack[103] Wormhole Attack[104] Sybil Attack[105]	-	Data Inconsistency[98] Data Breach[102]	Routing Security Système de localisation GPS Authentification du réseau Routage ad hoc tenant compte de la sécurité Chiffrement P2P Chiffrement des tables de routage
Couche de traitement	-	DoS/DDoS Attack[106]	Malware[100]	Unauthorized Access[107]	Scanner d'applications web Fragmentation redundancy scattering Encryption Hyper Safe

Tableau 3 : Les attaques dans l'IoT[97], [108]

II.4.1.2. Utilisation de la blockchain pour sécuriser l'IoT

Les caractéristiques de la blockchain, grâce à sa stratégie de décentralisation, en font une solution de sécurité très prometteuse pour l'IoT. La combinaison de ces technologies permet d'atteindre un certain niveau élevé de sécurité et d'améliorer la QoS. Ces dernières années, plusieurs recherches se sont focalisées sur l'utilisation de la blockchain pour sécuriser l'IoT.

Dans[109] les auteurs ont montré que ces appareils produisent une énorme masse de données, qui doivent être stockées de manière appropriée et analysées à des fins utiles. Pour chaque opération (création, mise à jour, suppression, lecture) des appareils IoT, les données peuvent être traitées comme une transaction dans des blocs. Des informations sur l'identité de l'appareil peuvent être stockées dans un bloc, comme l'identité du fabricant et l'état de fonctionnement de l'appareil à l'endroit où il se trouve. Les contrats intelligents sont utilisés pour mettre en œuvre des politiques de contrôle d'accès sur les appareils IoT qui peuvent identifier et détecter les accès non autorisés. Il n'est pas nécessaire de recourir à une autorité de stockage, telle que la configuration du cloud, pour protéger l'IoT. La blockchain est également utilisée pour garantir l'authenticité, l'intégrité et la traçabilité des données tout en permettant de sécuriser le canal entre les appareils IoT.

Certains auteurs ont montré l'utilisation de la blockchain pour sécuriser le SDN pour l'IoT. Dans[110], les auteurs explorent le potentiel de l'intégration de la blockchain et du réseau défini par logiciel (SDN) pour atténuer certains des défis. Plus précisément, ils proposent une architecture sécurisée et économe en énergie, basée sur la blockchain, de contrôleurs SDN pour les réseaux IoT utilisant une structure en grappe avec un nouveau protocole de routage. L'architecture utilise des blockchains publiques et privées pour la communication Peer to Peer (P2P) entre les dispositifs IoT et les contrôleurs SDN, ce qui élimine la preuve de travail (POW), et utilise une méthode d'authentification efficace avec une confiance distribuée, ce qui rend la blockchain adaptée aux dispositifs IoT à ressources limitées.

Les auteurs de ce travail ont présenté une solution possible en tirant parti de la négociation distribuée, de la prise de décision et des capacités d'apprentissage au niveau de l'appareil en utilisant des systèmes multi-agents compatibles avec l'IA et des contrats intelligents en temps réel entre les voitures et les péages à l'aide de la Blockchain. La solution mise en œuvre met également en évidence la monétisation temporelle des données provenant de divers dispositifs IoT. Alors que la chaîne de conservation garantit la confidentialité des participants, elle fait également office

de couche économique transactionnelle et de couche de gouvernance entre les dispositifs du réseau[111].

Dans ce travail, les auteurs ont introduit un nouveau concept de confidentialité appelé SVM sécurisé. Ils ont relevé les défis de la confidentialité et de l'intégrité des données en utilisant des techniques de provisionnement de Blockchain pour construire une formation sécurisée de l'algorithme SVM dans des scénarios multipartites où les données IoT sont collectées auprès de plusieurs fournisseurs de données. Ils utilisent un système de cryptographie homomorphe pour atténuer et construire un algorithme de formation SVM efficace et précis qui préserve la vie privée[112].

Dans ce travail, les auteurs soutiennent que si la blockchain est une technologie efficace pour garantir la sécurité et la confidentialité dans l'IoT, son application dans le contexte de l'IoT présente plusieurs défis importants. Pour relever ces défis, ils ont proposé une blockchain légère et évolutive (LSB) pour l'IoT. Le LSB dispose d'un algorithme de consensus pour l'IoT. Le LSB intègre une méthode de confiance distribuée selon laquelle le temps de traitement pour la validation de nouveaux blocs par les OBM diminue progressivement au fur et à mesure qu'ils obtiennent la confiance des autres. Les résultats qu’ils ont obtenus après la simulation prouvent que l'architecture proposée réduit la bande passante et le temps de traitement par rapport à l'architecture classique de la blockchain[113].

Dans ce travail, les auteurs ont présenté un système basé sur la Blockchain, garantissant ainsi la fiabilité des enregistrements stockés et permettant aux autorités de valider si une vidéo a été modifiée ou non. Cela permet de distinguer les fausses vidéos des vidéos originales et de garantir l'authenticité des caméras de surveillance. Étant donné que la blockchain enregistre les métadonnées de la vidéo de vidéosurveillance, elle fait également obstacle au risque de falsification des données. Cet enregistrement immuable réduit le risque de violation des droits d'auteur pour les organismes d'application de la loi et les clients en sécurisant la possession et l'identité[114].

Ces quelques travaux présentés ci-dessus, montrent l'importance de l'utilisation de la Blockchain pour la sécurité des différents domaines de l'IoT.

II.4.2. Sécurité du Cloud computing

Bien que le CC apporte plusieurs avantages, son adoption est encore ralentie par des problèmes de sécurité. Bien que cette technologie soit convoitée, elle comporte d'importantes menaces pour la sécurité. Celles-ci doivent être bien connues de tous les utilisateurs, qu'il s'agisse d'individus ou d'organisations, pour qu'elle soit adoptée. Comme précisé dans les sections précédentes, le cloud est basée sur la virtualisation. La sécurité dans le CC doit donc prendre en compte à la fois les ressources virtuelles et physiques. Dans cette partie du travail, nous discuterons des différentes menaces qui pèsent sur le Cloud Computing et des contre-mesures à adopter pour les différents types de ressources. Nous présenterons également les différents travaux qui montrent l'importance de la Blockchain dans l'amélioration de la sécurité du CC.

II.4.2.1. Attaques dans le Cloud Computing

L'accès aux ressources du Cloud Computing se fait via des réseaux, qu'ils soient locaux ou via Internet. Ces accès sont rendus possibles par les protocoles utilisés dans la communication. Le Cloud Computing est un ensemble de technologies qui présente une certaine vulnérabilité et ceci s'applique aussi bien aux ressources physiques que virtuelles. Dans cette partie, nous présentons les différentes attaques internes ou externes qui peuvent affecter la sécurité du Cloud. Ces attaques et les contre-mesures sont résumées dans le Tableau 4[115].

Attaques de sécurité	**Contre-mesure**	**Niveaux de sécurité**
Attaques par SQL Injection	Évitez d'utiliser du code SQL généré dynamiquement dans le code. Utiliser un filtrage approprié pour assainir l'entrée de l'utilisateur. Utiliser une architecture basée sur un proxy pour détecter et extraire dynamiquement l'entrée de l'utilisateur.	Niveau de base/ Niveau d'application
Attaques de phishing	Identifier les spams	Niveau de base
Attaques de type XSS	Utilisation du filtrage actif du contenu.	Niveau de base/

	Utilisation de la collaboration entre navigateurs. Utilisation d'une technologie de prévention des fuites de données basée sur le contenu. Utilisation d'une technologie de détection de la vulnérabilité des applications web. Approche basée sur un plan d'action pour minimiser la dépendance à l'égard du navigateur Configuration SSL. Utilisation d'un logiciel anti-malware.	Niveau d'application
Attaques DNS	Utiliser les mesures de sécurité du DNS	Niveau réseaux
Attaques MITM	Configuration correcte du protocole SSL. Utilisation d'outils de cryptage	Niveau de base
Attaques DOS	Utiliser de meilleurs systèmes d'authentification et d'autorisation. Utiliser des IDS/IPS	Application Level
Attaques DDOS	Utiliser de meilleurs mécanismes d'authentification et d'autorisation. IDS/(IPS).	Application Level
Adresse IP réutilisée	Utiliser de meilleurs mécanismes d'authentification et d'autorisation. IDS/(IPS).	Niveau réseau
Attaques par des Zombies	Utiliser de meilleurs mécanismes d'authentification et d'autorisation. IDS/(IPS).	Niveau réseau
Attaques par Wrapping	Utilisation d'une technique de détection par renifleur basée sur l'ARP Utilisation d'une technique de détection par renifleur basée sur le RTT.	Niveau réseau Niveau VM

Cookie Poisoning	Utilisation d'un mécanisme de signature approprié. Utilisation de la bonne configuration de SSL	Niveau application
Violation du CAPTCHA	Mettre en œuvre de meilleurs systèmes de cryptage. Nettoyer régulièrement les données des cookies. Utiliser la politique de sécurité du navigateur. Créer des sessions et utiliser d'autres mécanismes d'authentification.	Niveau application
Piratage de Google	Utiliser des mesures de sécurité standard pour les vulnérabilités associées au web. Utilisant de meilleurs systèmes d'authentification et d'autorisation Mettre en œuvre des politiques de sauvegarde pour éviter les problèmes de récupération des données.	Niveau application
Attaques d'hyperviseurs	Utilisation d'un hyperviseur sécurisé et surveillance adéquate de l'hyperviseur. Isolation des VM.	Niveau VM

Tableau 4 : Attaques et contre mesures dans le cloud[115]

II.4.2.2. Utilisation de la blockchain pour sécuriser le cloud

Dans ce travail, les auteurs ont utilisé le modèle d'agent d'une VM distribuée en la déployant dans le cloud avec la technologie d'agent mobile. L'agent de la VM permet à plusieurs locataires de coopérer pour fournir une vérification de confiance des données. Les tâches de stockage, de surveillance et de vérification des données sont effectuées par un mécanisme d'agent de machine virtuelle. Il s'agit également d'une condition préalable à la mise en œuvre d'un mécanisme de protection de l'intégrité de la chaîne. La protection de l'intégrité basée sur la blockchain est construite par le modèle proxy de la machine virtuelle et la valeur de hachage unique correspondant au fichier généré par l'arbre de hachage de Merkel. Elle est utilisée pour surveiller l'évolution des données à l'aide d'une application basée sur un contrat intelligent appliqué dans une Blockchain, et les données sont conservées dans le temps[116].

Pour superviser le partage des données dans un Cloud, les auteurs de[117] ont présenté un nouveau système de cryptage des signes basé sur les attributs de la blockchain (BABSC). Ce nouveau système a l'avantage d'utiliser deux modes de cryptage, l'un utilisant le cryptage Blockchain et l'autre le cryptage basé sur les attributs, garantissant ainsi la confidentialité et l'infalsifiabilité des données.

Les auteurs de [118] ont proposé un système basé sur la blockchain pour remplacer les TPA tout en concevant un contrat de paiement intelligent basé sur la blockchain pour l'audit du stockage dans le cloud public. Pour s'assurer que le fournisseur de services cloud soumet régulièrement des preuves de possession aux propriétaires de données, un contact intelligent basé sur la blockchain a été mis en place dans le système entre les deux parties. Le CSP n'est payé que si la vérification est réussie ; dans le cas contraire, il ne reçoit aucune rémunération mais doit payer les pénalités. Un contrat intelligent basé sur une Blockchain pour l'audit du stockage public en cloud basé sur la notion de possession de données publiques prouvables et non interactives a également été mis en place.

Dans ce travail, un système de partage des connaissances de fabrication basé sur le cloud pour la reconception des moules a été proposé pour construire un réseau sécurisé et de confiance dans l'alliance des moules. Les auteurs ont proposé un système distribué et sécurisé pour réaliser le partage des connaissances sur la préconception des moules d'injection, basé sur une Blockchain et un cloud privé. La méthode de recherche de documents k-Nearest Neighbor a également été appliquée dans le système pour améliorer l'efficacité du processus de recherche de connaissances[119].

II.5.Une architecture basée sur la blockchain pour sécuriser l'IoT et le cloud

Pour renforcer la sécurité du cloud et de l'IoT, la blockchain peut être utilisée pour garantir l'intégrité et l'authenticité des informations. Nous proposons une architecture basée sur la Blockchain pour sécuriser l'IoT et le Cloud Computing. Pour sécuriser l'IoT et le cloud computing. Cette architecture se compose de quatre couches, comme le montre la Figure 19.

Dans l'architecture IoT, chaque objet connecté est éventuellement une cible d'attaque qui peut être facilement compromise. Par exemple, en ce qui concerne l'appariement, les appareils IoT permettent généralement à tout contrôleur de proximité de se connecter à l'appareil IoT pour sécuriser les configurations. Par exemple, la sécurité basée sur la blockchain pourrait renforcer les

mécanismes d'authentification de ces appareils. Dans notre architecture, à la première couche, nous proposons que chaque appareil du réseau IoT (quelle que soit la topologie adoptée) soit considéré comme un bloc de la Blockchain qui peut résoudre le problème d'authentification par le biais de citations ou d'attaques DDoS (Distributed Denial of Service).

Les données collectées par les appareils IoT sont traitées et stockées dans le cloud. Pour ce faire, ces appareils se connectent au cloud computing via l'internet s'ils se trouvent à proximité ou via les diverses autres technologies mentionnées dans la couche 2 de la Figure 19. Ces appareils doivent s'identifier et s'authentifier par l'intermédiaire des serveurs du cloud computing. Notre architecture, dans sa troisième couche, propose de considérer chaque VM (machine virtuelle) hébergée dans un serveur Sn comme un bloc de la blockchain afin de renforcer la sécurité du service en cloud. Le seul problème lorsque nous créons une Blockchain dans différents serveurs cloud pourrait être lié à l'optimisation de ces ressources virtuelles (VM).

Dans le modèle proposé, nous avons proposé un modèle global pour les projets Big Data dans le contexte de l'architecture basée sur la blockchain pour sécuriser l'IoT et du CC. Ce cadre nous permet de collecter tous les types de données entrantes provenant de la Blockchain, des capteurs IoT ou de toute autre source de Big Data.

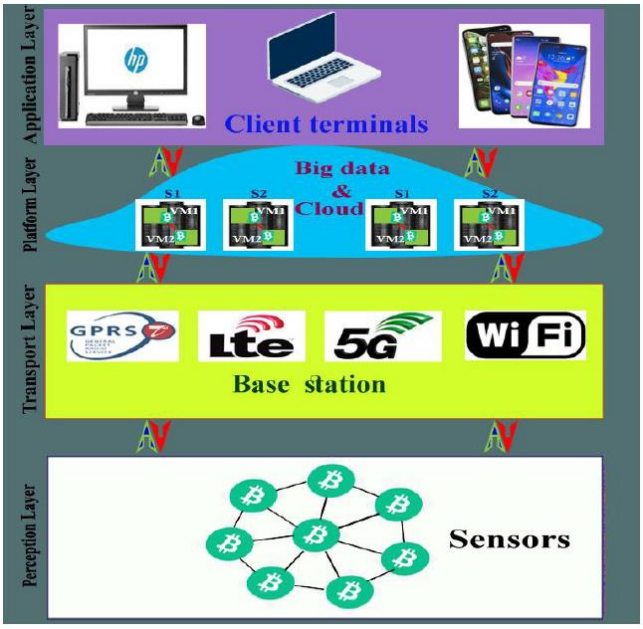

Figure 19 : Architecture pour renforcer la sécurité dans l'IoT et le cloud

Ensuite, il nous donne la possibilité de choisir le système de niveau par rapport à notre architecture. Après cela, nous devons appliquer un modèle de sécurité ou d'autres protocoles sécurisés dans nos

stockages pour assurer l'intégrité des données, car si nous découvrons des modèles, par exemple, nous avons nos blocs intégraux dans le magasin de données ou nous avons dans le HDFS (Hadoop Distributed File System) tous les nœuds de nom accédant à chaque ensemble de données pour leur appliquer un algorithme de haut niveau dans l'apprentissage automatique, l'intelligence artificielle, ou de tout autre domaine de la science pour éviter le gaspillage et la perte de données. Ce modèle basé sur la blockchain a été conçu dans le cadre d'un projet d'analyse de Big Data lier au stockage HDFS et au cloud computing, où toutes les ressources d'information sont facilement accessibles.

II.6. Conclusion

L'avènement des NTIC a favorisé l'adoption de l'IoT. Il en résulte un flux important de données collectées par les appareils IoT, traitées et stockées dans le CC. Le plus grand défi qui menace l'IoT et le cloud est la sécurité de ces technologies. L'utilisation de la blockchain pour renforcer la sécurité de ces technologies est parfois essentielle de nos jours. Dans ce chapitre, nous avons proposé une architecture basée sur la blockchain pour sécuriser l'IoT. Tout d'abord, nous avons présenté les travaux existants. Ensuite, nous avons analysé la portée de l'IoT et de la blockchain. Troisièmement, nous avons présenté les travaux sur l'utilisation de la Blockchain comme bonne pratique pour la sécurité de l'IoT, nous avons proposé une architecture liée à l'amélioration de la sécurité de divers dispositifs IoT et de Cloud Computing. Dans les travaux futurs, nous avons l'intention d'analyser l'impact direct de l'architecture proposée sur l'utilisation des ressources virtuelles dans le CC.

CHAPITRE III: APPROCHES POUR L'OPTIMISATION DU CLOUD COMPUTING

III.1. Introduction

Les problèmes d'optimisation sont essentiels dans divers domaines comme les mathématiques discrètes, la recherche opérationnelle, l'informatique, ainsi que divers problèmes de l'ingénieries. Ces problèmes peuvent impliquer des variables continues ou discrètes (combinatoires), être mono-objectifs ou multi-objectifs, statiques ou dynamiques, et peuvent comporter ou non des contraintes. Les systèmes de recommandation peuvent être appliqués suivant différents type de problème et du domaine concerné. Il est important de noter qu'un problème d'optimisation peut combiner plusieurs de ces caractéristiques, comme être à la fois continu et dynamique.

Un problème d'optimisation est défini par un ensemble de variables, une fonction de coût, et un nombre limité de contraintes. L'espace de recherche, qui inclut toutes les solutions possibles, a une dimension par variable et est généralement fini pour des raisons pratiques et de calcul. La fonction objective, qu'on cherche à minimiser ou maximiser, indique le but à atteindre. Les contraintes, constituées d'égalités et d'inégalités, restreignent cet espace de recherche.

Dans ce chapitre nous nous intéressons aux différentes méthodes qui sont utilisées pour optimisés les différentes ressources dans le cloud. En premier lieu, nous nous intéressons sur les approches métaheuristiques utilisées dans le VMP, en deuxième lieu nous allons aborder les systèmes de recommandations.

III.2. Optimisation du placement des Machines Virtuelles

III.2.1. Les Approches métaheuristiques pour le VMP

Dans le domaine du cloud computing, la problématique du placement des machines virtuelles constitue l'une des questions les plus cruciales. Il s'agit d'assigner des machines virtuelles à des serveurs physiques tout en optimisant certains objectifs tels que l'utilisation des ressources, la consommation d'énergie et le temps de réponse. Son objectif principal est de pouvoir maximiser l'utilisation des ressources tout en respectant les accords de niveau de service des machines virtuelles.

Résoudre efficacement le problème du placement des machines virtuelles dans les grands centres de données est une tâche complexe et chronophage. Les métaheuristiques, comme les algorithmes

génétiques, le recuit simulé, les colonies de fourmis et les essaims de particules, sont des techniques d'optimisation avancées utilisées pour trouver des solutions approximatives à ce défi NP-difficile. Ces méthodes itératives sont conçues pour gérer une variété de contraintes et de types de problèmes, offrant ainsi une approche flexible pour résoudre le problème de placement de VM.

Dans la littérature, on recense plus de 500 métaheuristiques utilisées pour optimiser divers problèmes[120]. Dans cette partie de notre travail, nous allons présenter certaines des métaheuristiques les plus anciennes et classiques pouvant etre appliqués à l'optimisations du cloud.

III.2.1.1. Algorithme génétique

L'algorithme génétique (AG) est une méthode de recherche basée sur les principes de la sélection naturelle et de la génétique. Il fonctionne en créant une population initiale de solutions potentielles, puis en les combinant pour former des individus, représentés sous forme de chromosomes. Ces individus sont évalués et classés en fonction de leur pertinence par rapport à la meilleure solution, qui est inconnue au départ. À travers un processus de sélection naturelle similaire à celui proposé par Darwin, l'algorithme évolue progressivement vers une solution optimale pour le problème donné, grâce à des échanges d'informations structurés et une stratégie de "survie du plus fort"[121]. Les individus les plus aptes de la population sont plus susceptibles de se reproduire et de transmettre leur information génétique à la génération suivante[122].

Dans un algorithme génétique, une nouvelle génération est créée par la combinaison des gènes parentaux, avec l'espoir que les descendants héritent des meilleures caractéristiques des deux parents. Ce processus se répète jusqu'à ce que tous les individus partagent les mêmes informations génétiques, généralement celles conduisant à la meilleure solution. Les opérations de sélection, mutation et croisement sont utilisées pour créer cette nouvelle génération, fonctionnant de manière similaire à des opérations algébriques.

La sélection

La sélection est une étape cruciale dans les algorithmes génétiques, visant à choisir les individus les plus performants pour la reproduction et la création de la génération suivante. Il existe deux types de sélection : parentale pour la reproduction et de remplacement pour maintenir la taille de la population. L'objectif est de favoriser les individus performants tout en évitant une convergence prématurée. Les méthodes de sélection incluent la sélection aléatoire, élitiste, par tournoi, etc., tout en tenant compte de la nature du problème et la stratégie de l'algorithme.

Sélection par roulette (RWS : Roulette Wheel Selection) : La sélection par RWS consiste à attribuer à chaque individu une probabilité de sélection proportionnelle à sa valeur d'évaluation. Cette technique est utilisée pour maximiser la fonction d'évaluation de la population. Dans le but de maximiser la fonction d'évaluation, la formule mathématique ci-dessous présente la probabilité de sélection d'un individu[123]:

$$P_{SP} = \frac{F_p}{\sum_P F_P} \quad (1)$$

Avec : F_p qui évalue l'idividue p et P_{SP} la probabilité de sélection de l'individue p

Pour minimiser la fonction d'évaluation, la probabilité de sélection d'un individu se calcule par la formule suivante[124] :

$$P_{SP} = \frac{F_p}{\sum_P \frac{1}{F_P}} \quad (2)$$

En d'autres termes, plus la valeur d'évaluation d'un individu est élevée (F_p), plus sa probabilité d'être sélectionné est grande (P_{SP}).

La sélection des individus dans un processus de reproduction génétique peut se faire via une roulette, où la probabilité de sélection est proportionnelle à l'évaluation de l'aptitude de chaque individu. Cependant, cette méthode présente le risque d'un "Super individu" bénéficiant d'une sélection disproportionnée. Pour éviter cet inconvénient, des méthodes comme la sélection par tournoi sont préférées car elles assurent une sélection équitable sans favoriser un individu particulier.

Sélection par Tournoi :

La sélection par tournoi est une méthode aléatoire où deux individus sont choisis et comparés pour déterminer un gagnant parmi une population de N chromosomes. Cette approche évite la domination d'un seul individu, mais peut ne pas toujours choisir le meilleur, limitant ainsi l'exploration[125].

Sélection par Stochastic Universal Sampling (SUS) :

La méthode de sélection dont il est question ici repose sur le même principe de base que la méthode de sélection de la roulette, mais au lieu de s'appuyer sur un seul indicateur, elle utilise plusieurs indicateurs régulièrement espacés. Toutefois, ces méthodes de sélection proportionnelle peuvent

rencontrer des problèmes de mise à l'échelle. Heureusement, la méthode de sélection par rang peut être utilisée comme alternative pour éviter ce problème[126].

Sélection par rang :

La sélection par rang attribue des probabilités de reproduction basées sur le rang des chromosomes plutôt que sur leurs valeurs d'évaluation. Ce processus commence par trier les chromosomes selon leur aptitude, puis attribue des rangs de 1 pour le moins adapté à N pour le plus adapté, où N est la taille de la population. Cette méthode est similaire à la sélection à la roulette mais utilise les rangs pour déterminer les proportions[125].

Le Croisement :

L'opérateur de croisement, essentiel en génétique, crée deux chromosomes enfants à partir de deux parents, basé sur une probabilité P_c, définie par l'utilisateur selon le problème d'optimisation. Ce processus permet aux enfants d'hériter de portions des gènes de leurs parents, générant ainsi de nouvelles séquences génétiques.

Le type de codage et la nature du problème traité sont les principaux facteurs qui déterminent l'opérateur de croisement spécifique employé. Il existe plusieurs opérateurs de croisement pour le codage binaire, notamment le croisement en un point et le croisement en plusieurs points.

A. Croisement à un point

Pour effectuer ce type de croisement, un point de croisement aléatoire est sélectionné pour chaque paire, comme illustré à la Figure 20. Le croisement se produit au niveau binaire plutôt qu'au niveau du gène et peut traverser le milieu d'un gène.

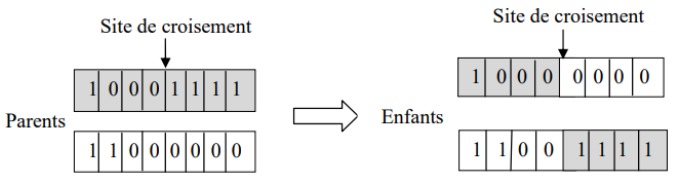

Figure 20 : Croisement à un point

B. Croisement en deux points

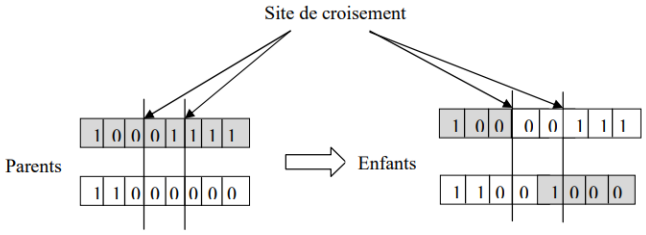

Figure 21 : Type de croisement à deux points

Dans la littélature, il existe d'autres types de cloisement tels que le croisement JOX (job-based order crossover)[127], le croisement PMX(partial-mapped crossover)[125], ER (edge recombination crossover)[128], OX (order crossover)[129], etc.

La mutation

C'est un processus qui consiste à apporter des changements aléatoires aux génotypes sur la base d'une règle probabiliste prédéfinie, où la probabilité que le changement se produise est généralement fixée par l'utilisateur à une faible valeur P_m. La mutation modifie un ou plusieurs allèles d'un chromosome(deux copies d'un même gène), maintenant ainsi la diversité des individus pour éviter les optima locaux et la perte permanente de caractéristiques. Dans les chromosomes binaires, elle convertit un 1 en 0 ou vice versa.

Différents opérateurs de mutation incluent : la transposition de deux allèles adjacents (échanger deux allèles consécutifs au hasard), la transposition de deux allèles quelconques (échanger deux allèles choisis aléatoirement), et l'inversion d'allèles (inverser l'ordre de deux allèles choisis aléatoirement). La probabilité de mutation P_m peut se calculer comme suit :

$$P_m = 1 - \sigma^{\frac{-1}{l}} \quad (3)$$

Ou l désigne la longueur du chromosome et σ représente la pression sélective qui peut etre 1,8.

III.2.2.2. L'algorithme de la fourmi

Le terme "colonie de fourmis" fait référence à une classe d'algorithmes "Ant System" développé par Colorni, Dorigo et Maniezzo en 1992[130]. Cet algorithme est basé sur le comportement de communication des fourmis connu sous le nom de stigmergie, où les fourmis déposent une substance volatile appelée phéromone pour construire un chemin lorsqu'elles explorent un

environnement. Les autres fourmis de la colonie "lisent" cette information avec leurs antennes et ont tendance à choisir le chemin ayant la plus forte concentration de phéromone. La Figure 22 montre l'expérience menée par Goss et al. en 1989[131] sur le comportement d'auto-organisation des fourmis d'Argentine illustre ce phénomène. Au départ, une fourmi choisit un chemin entre son nid et une source de nourriture et dépose de la phéromone tout en retournant au nid par un autre chemin. La même Figure 22 montre comment la colonie renforce le chemin le plus court avec de la phéromone, ce qui le rend plus attrayant, et finit par choisir le chemin le plus court tout en abandonnant les chemins plus longs en raison de l'évaporation de la phéromone.

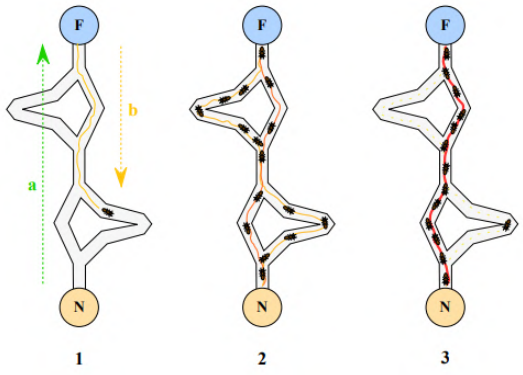

Figure 22 : La sélection du chemin le plus court par une colonie de fourmis[133]

Au départ, l'ant system a été développé pour parier au problème du voyageur de commerce (TSP). Cette métaheuristique est unique en ce sens qu'elle utilise une métaphore de créatures sociales dotées d'une mémoire centralisée, à savoir la phéromone.

Supposons une colonie composée de m fourmis et de n villes à relier. À chaque itération t, chaque fourmi k construit une tournée en reliant une ville i à une autre ville j non visitée. Ce processus repose sur différents facteurs, notamment une liste de villes à visiter par la fourmi k lorsqu'elle se trouve en i, une variable qui représente la visibilité d'une ville j depuis une ville i, et la quantité de phéromone entre deux villes, qui indique l'intensité de l'utilisation de ce chemin. En outre, ce processus est déterminé par trois paramètres : $\alpha, \beta \; et \; Q$.

La formule ci-dessous décrit la modélisation mathématique de la règle du déplacement :

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{\tau_{ij}(t)^{\alpha}.\eta_{ij}^{\beta}}{\sum_{l \in J_i^k} \tau_{il}\,(t)^{\alpha}.\eta_{il}^{\beta}} & si\ j \in J_i^k \\ 0 & Si\ non \end{cases} \quad (4)$$

Une fois que la fourmi a terminé le chemin, elle laisse une quantité spécifique de phéromone déterminée par $\Delta\tau_{ij}^{k}(t)$, qui varie en fonction de la qualité de la solution. Elle est déterminée par la longueur totale du chemin découvert, $L^{k(t)}$[134]. Cette valeur peut être exprimée comme suit :

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} \frac{Q}{L^{k(t)}} & si\ (i;j) \in T^{k}(t), \\ 0 & Si\ non \end{cases} \quad (5)$$

Dans ce contexte, $T^{k}(t)$ désigne le chemin que la fourmi k a déjà parcouru après t itération, tandis que Q est une valeur fixe dans l'algorithme. Au fur et à mesure que la longueur du chemin augmente, la concentration de phéromone déposée par les fourmis diminue. À la fin de chaque cycle, la phéromone sur chaque chemin est évaporée pour éliminer les mauvaises solutions. Le taux d'évaporation est déterminé par un facteur de réduction constant, qui est calculé par:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t) \quad (6)$$

Dans cette formule m représente le nombre de fourmi et ρ est un paramètre de l'algorithme représentant le taux d'évaporation.

III.2.2.3. Le recuit simulé

Le concept de recuit simulé tire son inspiration de la métallurgie, s'appuyant sur un processus de refroidissement et de réchauffement progressif d'un matériau dans le but de réduire son énergie globale, conformément aux principes de la thermodynamique de Boltzmann. À partir de 1983, Kirkpatrick, C.D. Gelatt et M.P. Vecchi ont transposé cette méthode au domaine de l'optimisation. Dans cette adaptation, la fonction objectif f est assimilée à l'énergie du matériau, tandis que les paramètres de l'algorithme incluent des éléments tels que le critère d'arrêt, la température initiale T_o (dont la détermination peut varier selon différentes approches), le seuil de variation de température ainsi que la méthode de décroissance de la température[135]. L'algorithme débute par une marche aléatoire, avec une température décroissante qui réduit progressivement l'acceptation des mauvaises solutions, convergeant ainsi vers une solution optimale. Un ajustement équilibré de la décroissance de la température est crucial pour éviter un piégeage dans un optimum local[136]. Pour obtenir un nouvel état, un atome est soumis à un déplacement aléatoire minime par rapport à n'importe quel autre atome, ce qui entraîne un changement d'énergie (ΔE). Le nouvel état est accepté si l'énergie du système diminue ($\Delta E < 0$). En revanche, si l'énergie reste inchangée ($\Delta E = 0$), le nouvel état est accepté avec une certaine probabilité :

$$prob(\Delta E, t) = exp(\frac{-\Delta E}{k_B.t}) \quad (7)$$

La température du système est représentée par t, et une constante physique, appelée constante de Boltzmann, est notée k_B. À chaque étape du processus, un nouvel état est accepté au hasard, même si son énergie n'est pas inférieure à celle de l'état actuel. Pour décider d'accepter ou non le nouvel état, un nombre q est généré au hasard dans l'intervalle [0,1[. Si q est inférieur ou égal à prob$(\Delta E, t)$, le nouvel état est accepté, sinon l'état actuel est maintenu. En appliquant cette règle de manière répétée, les recherches ont démontré que le système finira par atteindre un état d'équilibre thermique[137].

III.2.2.4. Algorithme de recherche tabou

Fred Glover a proposé la méthode de recherche tabou en 1986[138], qui prend en compte les solutions passées en utilisant une mémoire des solutions explorées pendant la recherche d'une solution optimale. Cette mémoire est connue sous le nom de liste tabou car elle comprend des solutions qui ont déjà été visitées et qui ne doivent pas être revisitées[139]. L'objectif de cette mémoire est d'éviter que l'algorithme se bloque dans un optimum local en explorant des solutions initialement moins aptes. La taille de la mémoire, déterminant le nombre d'éléments mémorisés, permet d'équilibre la phase de diversification ainsi que celle de l'intensification. Une petite mémoire favorise l'intensification en limitant les solutions interdites, tandis qu'une grande mémoire favorise la diversification en permettant une exploration plus large[140].

La gestion de la mémoire dans le processus de recherche est influencée par la portée de la liste tabou. Pour favoriser l'approfondissement, il est possible de restreindre cette liste. Cependant, l'élargissement de cette dernière contraint le processus de recherche à explorer de nouveaux horizons, ce qui encourage la diversification. Il est donc envisageable d'adapter la taille de la liste tabou au fur et à mesure de l'avancement de la recherche[141]. Une avancée significative dans l'amélioration de la méthode de recherche tabou consiste à intégrer des mécanismes de stockage de données à moyen et long termes. Cette intégration vise à enrichir les notions d'intensification et de diversification.

III.2.2.5. Algorithme d'Optimisation par Essaim Particulaire

L'algorithme d'optimisation par essaim de particules (PSO) a été proposé par Russel Eberhart et James Kennedy en 1995 en tant qu'algorithme basé sur la population[142]. Contrairement aux

algorithmes génétiques, l'algorithme PSO ne repose pas sur la sélection et le croisement des individus, mais met plutôt l'accent sur la coopération entre eux[143]. Le PSO utilise une population d'agents ou de particules qui, bien que peu intelligents et ayant une connaissance limitée de leur environnement, communiquent largement selon une architecture de voisinage spécifique, semblable aux bancs de poissons ou volées d'étourneaux. Contrairement aux ACO, le PSO ne repose pas sur une mémoire centralisée. Les particules, appelées solutions, explorent l'espace de recherche et forment un essaim, représentant la population, tout en respectant des contraintes de direction et de vitesse pour maintenir une organisation cohérente.

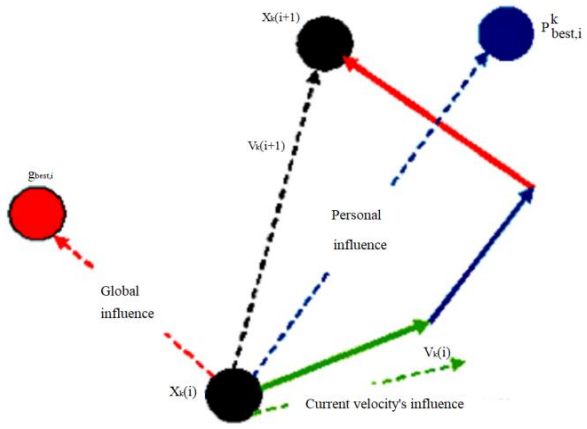

Figure 23 : Plan de génération des particules[131]

Chaque particule est caractérisée par une position X_i et une vitesse V_i (Vi). L'objectif est de déplacer les positions des particules vers la solution optimale. Pour ce faire, le mouvement de chaque particule est influencé par trois composantes : physique, cognitive et sociale[144]. La composante physique prend en compte la position actuelle de la particule, la composante cognitive considère la meilleure solution trouvée par la particule jusqu'à présent (P_i) et la composante sociale considère la meilleure solution trouvée par l'ensemble de l'essaim (P_g). Mathématiquement, cela peut s'exprimer comme suit[144] :

$$\overrightarrow{V_i(t+1)} = \omega \times \overrightarrow{V_i(t)} + cC_1 \times r_1 \times \left(\overrightarrow{P_i(t)} - \overrightarrow{X_i(t)}\right) + C_2 \times r_2 \times \overrightarrow{P_g(t)} - \overrightarrow{X_i(t)} \qquad (8)$$

Soit (C_1, C_2) deux constantes qui représentent une accélération positive. Nous générons deux nombres aléatoires (r_1, r_2) à partir d'une distribution uniforme dans l'intervalle [0 ; 1]. Ici, i varie de 1 à N, où N est la taille de l'essaim, et ω est le coefficient d'inertie[145], [146]. La nouvelle position peut être calculée à l'aide de la formule suivante :

$$\overrightarrow{X_\iota(t+1)} = \overrightarrow{X_\iota(t)} + \overrightarrow{V_\iota(t+1)} \quad (9)$$

Pour réguler la vitesse des particules dans l'espace de recherche et équilibrer la diversification et l'intensification, il est possible d'établir une limite de vitesse maximale qui restreint la vitesse des particules à un intervalle de [-Vmax ; Vmax].

III.2.2.6. Algorithme d'Optimisation par Essaim Particulaire

En 1986, Fred Glover a proposé la méthode de recherche tabou [47]. Cette méthode utilise une mémoire, appelée liste taboue, pour éviter les optima locaux en interdisant de revisiter certaines solutions. Cela permet à l'algorithme d'explorer de nouvelles solutions potentiellement meilleures. La taille de la mémoire est cruciale pour équilibrer diversification et intensification : une mémoire petite favorise l'intensification, tandis qu'une grande favorise la diversification. L'algorithme commence par une solution initiale aléatoire et sélectionne ensuite itérativement la meilleure solution voisine tout en évitant le cyclage.

Il existe une variante à mémoire adaptative de cette métaheuristique, qui permet à la mémoire d'adapter sa taille en fonction du contexte de recherche et même de générer de nouvelles solutions.

III.2.3. Les métriques pour évaluer le placement de machines virtuelles

Le placement des machines virtuelles (VM) en cloud computing consiste à choisir un hôte physique pour une VM en fonction de la disponibilité des ressources, la charge de travail et les contraintes système. L'objectif est d'optimiser l'utilisation des ressources, de minimiser la consommation d'énergie et d'assurer des performances élevées et une tolérance aux pannes. Les performances des algorithmes de placement de VM sont évaluées à l'aide de diverses métriques, dont plusieurs seront examinées dans les deux derniers chapitres de ce travail.

III.2.3.1. Utilisation des ressources

L'utilisation des ressources est cruciale pour évaluer les algorithmes de placement des VMs, mesurant l'efficacité de l'utilisation des ressources physiques comme le processeur, la mémoire, le disque et la bande passante. Un taux élevé d'utilisation des ressources indique une meilleure performance et des coûts réduits, tandis qu'un faible taux conduit à un gaspillage et une sous-utilisation du matériel. Les chapitres VI et VII se concentreront sur la modélisation de cette métrique pour minimiser les coûts liés au gaspillage des ressources dans un système cloud.

III.2.3.2. Équilibrage de la charge

L'équilibrage de la charge est une autre mesure importante pour évaluer les algorithmes de placement de VM. Il mesure l'homogénéité de la répartition de la charge de travail entre les hôtes physiques. Un placement bien équilibré garantit qu'aucun hôte n'est surchargé alors que d'autres sont sous-utilisés. Cela permet d'améliorer les performances et de réduire le risque de défaillance du matériel. L'équilibrage de la charge est particulièrement important dans les environnements dynamiques où la charge de travail peut changer rapidement.

III.2.3.3. Coût de migration

Le coût de migration est un indicateur qui mesure le coût de la migration des machines virtuelles d'un hôte physique à un autre. Le coût comprend le temps d'arrêt, la bande passante du réseau et l'utilisation des ressources pendant le processus de migration. Un coût de migration faible indique un meilleur placement. Cette mesure est particulièrement importante dans les systèmes qui nécessitent une migration fréquente des machines virtuelles, comme l'équilibrage de la charge ou la tolérance aux pannes.

III.2.3.4. Tolérance aux pannes

La tolérance aux pannes est un indicateur qui mesure la capacité de l'algorithme de placement de VM à gérer les pannes matérielles. Un placement tolérant aux pannes garantit que même si un ou plusieurs hôtes physiques tombent en panne, les machines virtuelles continuent de fonctionner sans interruption. Le placement tolérant aux pannes doit être capable de détecter rapidement les pannes et de migrer automatiquement les machines virtuelles concernées vers d'autres hôtes. Cette mesure est essentielle pour les systèmes qui exigent une disponibilité et une fiabilité élevées.

III.2.3.5. Efficacité énergétique

L'efficacité énergétique est un indicateur qui mesure la consommation d'énergie du CD. Les algorithmes de placement efficaces sur le plan énergétique visent à minimiser la consommation globale d'énergie du CD en consolidant les machines virtuelles sur un plus petit nombre d'hôtes physiques. Cette mesure est particulièrement importante pour les CDs dont les coûts énergétiques sont élevés et qui se préoccupent de leur empreinte carbone.

L'évaluation des algorithmes de placement de VM est essentielle pour assurer des performances optimales, la tolérance aux pannes et l'efficacité énergétique dans les systèmes de cloud computing. Les mesures d'évaluation incluent l'utilisation des ressources, l'équilibrage de la charge, les coûts de migration, la tolérance aux pannes et l'efficacité énergétique. Un bon

algorithme doit équilibrer ces critères pour fournir les meilleures performances en fonction de la charge de travail et des contraintes du système, avec des métriques adaptées aux objectifs d'optimisation.

III.3. Les systèmes de recommandation pour l'optimisation du Cloud

Dans le monde d'aujourd'hui, l'internet a facilité un accès sans précédent à l'information grâce à l'échange de quantités massives de données stockées dans des CDs basé sur le cloud. Cependant, cette abondance d'informations pose également un défi de taille, car elle nécessite des processus de filtrage étendus pour garantir que les utilisateurs reçoivent rapidement des données pertinentes. Il n'est pas possible pour les humains d'analyser toutes les informations disponibles de manière exhaustive. Pour les aider dans cette tâche, des systèmes de recommandations aux consommateurs ont été développés. Cette section, commence par définir ce qu'est un système de recommandation, suivi d'un aperçu des trois principales approches de filtrage utilisées pour la recommandation. Nous nous pencherons ensuite sur la factorisation matricielle, ses modèles et son importance. Enfin, nous discuterons des limites et des perspectives d'avenir de ces systèmes.

III.3.1. Définition et terminologies des systèmes de Recommandation

L'objectif principal d'un système de recommandation est de proposer aux utilisateurs des articles pertinents en fonction de leurs préférences. Cela permet de réduire le temps passé par l'utilisateur à rechercher les articles qui l'intéressent le plus et d'identifier les articles qu'il est susceptible d'apprécier mais qu'il aurait pu négliger. Les systèmes de recommandations ont été définis de différentes manières. L'une des définitions les plus courantes et les plus générales, citée dans les travaux de Robin Burke[147], est la suivante :

Un système de recommandation personnalise les suggestions ou guide les utilisateurs vers des ressources pertinentes dans un vaste ensemble de données. Il se base sur les préférences exprimées par les utilisateurs sous forme de scores, souvent entre 1 et 5, ou binaires ("j'aime" ou "je n'aime pas"), formant ainsi une matrice de scores. Les deux principales tâches du système sont la prédiction des scores inconnus (prédire les préférences pour des éléments non notés) et la recommandation (générer une liste ordonnée de n suggestions, appelée liste Top-n).

Dans la partie suivante, nous adopterons une série de notations pour identifier les divers éléments constitutifs d'un modèle de système de recommandation. Nous désignerons ainsi par U l'ensemble des utilisateurs du système, tandis que l'ensemble des éléments sera représenté par la lettre I. La

matrice scores[147] stocke les scores des utilisateurs. $I_u$représente l'ensemble des éléments que l'utilisateur u a noté, tandis que $U_i$représente l'ensemble des utilisateurs qui ont noté l'élément i. Le score que l'utilisateur u a attribué à l'élément i est noté $r_{u,i}$.

III.3.2. Les différentes approches des systèmes de recommandations

III.3.2.1. Les recommandations basées sur le contenu

Cette section décrit les méthodes de recommandation basées sur le contenu qui analysent le contenu des articles ou leurs descriptions. Ces approches exploitent des méthodologies qui s'inspirent de l'exploration d'informations, sans toutefois exiger de demandes explicites de la part des utilisateurs. Elles se fondent plutôt sur l'analyse des inclinations de l'utilisateur pour suggérer des articles présentant des similitudes avec ceux qui ont été appréciés précédemment[148], [149], [150]. Par conséquent, lorsque de nouveaux éléments sont ajoutés au système, ils peuvent être recommandés immédiatement sans nécessiter de temps d'intégration, comme c'est le cas pour les systèmes de recommandation basés sur le filtrage collaboratif.

Pour recommander des éléments basés sur le contenu, il faut créer des profils d'éléments et d'utilisateurs. Cela inclut les articles et les attributs descriptifs des utilisateurs. Une approche basée sur le contenu analyse les éléments évalués par un utilisateur pour construire un profil de ses intérêts, divisé en profil "positif" (articles aimés) et "négatif" (articles non aimés). Le système compare les attributs des articles candidats aux profils de l'utilisateur pour recommander ceux qui correspondent au profil "positif" et diffèrent du profil "négatif". L'efficacité du système augmente avec la précision du profil utilisateur.

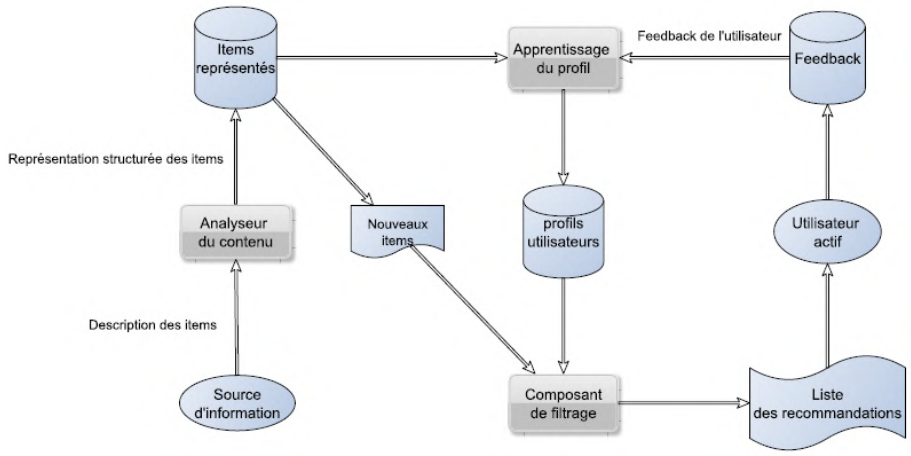

Figure 24 : Architecture d'une recommandation basée sur le contenu[151]

La mise en place d'un système de recommandation centré sur le contenu requiert l'utilisation de méthodes visant à représenter de manière efficace les éléments en question ainsi que le profil de l'utilisateur. Pour les comparer, Lops et al.[151] ont proposé une architecture de haut niveau dans laquelle le processus de recommandation est réalisé en trois étapes gérées par des composants spécifiques.

Pour suggérer des contenus en fonction de leur nature, il est nécessaire de créer deux ensembles distincts : les profils d'éléments et les profils d'utilisateurs. Dans ce contexte, le contenu englobe non seulement le texte des articles, mais également les caractéristiques des utilisateurs. Une méthode d'analyse basée sur le contenu consiste à examiner un ensemble d'articles précédemment évalués ou consultés par un utilisateur afin de créer un modèle ou un profil de ses intérêts. Ce modèle est élaboré en se basant sur les caractéristiques des articles que l'utilisateur a aimés ou n'a pas aimés. Le profil utilisateur comporte généralement deux volets : un volet "positif" qui regroupe les articles appréciés, et un volet "négatif" qui regroupe ceux qui n'ont pas été appréciés. La recommandation consiste alors à comparer les attributs des articles potentiels avec ceux des volets "positif" et "négatif" du profil utilisateur. Ainsi, les contenus recommandés sont ceux qui ressemblent au volet "positif" du profil utilisateur et diffèrent de son volet "négatif". L'efficacité du système de recommandation s'améliore à mesure que le profil utilisateur reflète de manière plus précise ses préférences.

Pour qu'un système de recommandation basé sur le contenu puisse recommander efficacement des articles, il doit utiliser des techniques pour représenter efficacement les articles et le profil de l'utilisateur afin de permettre la comparaison. Pour ce faire, une architecture qui a été représentée dans la Figure 24 se compose de trois éléments, chacun étant responsable d'une étape spécifique du processus de recommandation :

- Le premier composant est l'analyseur de contenu qui est utilisé pour prétraiter les informations non structurées, telles qu'un article textuel, et extraire les informations pertinentes. Ces informations extraites sont ensuite structurées et représentées sous une forme appropriée, telle qu'un vecteur de mots-clés.

- La deuxième phase, axée sur l'apprentissage de profil, a pour objectif de synthétiser les données reflétant les préférences de l'utilisateur et de créer son profil correspondant. Des approches de machine learning, comme les arbres de décision, les réseaux neuronaux et la classification naïve de Bayes, peuvent être mises en œuvre pour élaborer le profil de l'utilisateur en se basant sur ses interactions avec les articles, qu'il les ait appréciés ou non.

- Le composant de filtrage, troisième élément fondamental du processus de recommandation, joue un rôle crucial. Son objectif est de sélectionner les éléments les plus pertinents en comparant le profil de l'utilisateur avec les candidats à la recommandation. Pour évaluer cette pertinence, une métrique de similarité est employée, mesurant la proximité entre chaque élément et le profil de l'utilisateur. Ainsi, plus la similarité avec le profil "positif" est élevée et moins elle est avec le profil "négatif", plus l'élément a de chances d'être recommandé.

Pour construire et tenir à jour le profil de l'utilisateur actuel, la composante "feedback" recueille et stocke ses réponses aux éléments (évaluations). Dans le processus d'apprentissage, ces évaluations sont employées pour estimer avec précision la pertinence probable d'un élément non encore évalué par l'utilisateur. Les utilisateurs peuvent également créer leurs domaines d'intérêt à l'avance en tant que profil initial, mais ce scénario est assez peu fréquent.

III.3.2.1.1. Représentation d'un item

Les systèmes de recommandations basés sur le contenu s'appuient souvent sur des données non structurées telles que des descriptions d'articles sous forme de texte, de pages web, d'e-mails ou de descriptions de produits sur un site web de commerce électronique. Contrairement aux données structurées, les données non structurées n'ont pas d'attributs bien définis avec des valeurs claires. L'identification des caractéristiques des utilisateurs à partir de données non structurées peut se révéler ardue, principalement en raison des défis inhérents au langage naturel. En effet, ce dernier est sujet à des ambiguïtés telles que la polysémie, où un mot peut avoir plusieurs significations, ainsi que la synonymie, où plusieurs mots peuvent exprimer la même idée. Par conséquent, une étape de prétraitement peut être nécessaire pour extraire les informations pertinentes des données non structurées et les structurer en un ensemble d'attributs.

III.3.2.1.2. Recommandation basée sur les mots-clés

La majorité des approches pour les systèmes de recommandation basés sur le contenu adoptent une structure courante, connue sous le nom de modèle de l'espace vectoriel (VSM), associée au schéma de pondération TF-IDF (Term Frequency-Inverse Document Frequency) pour représenter les descriptions des éléments. Dans ce modèle, chaque élément est défini par un vecteur possédant un nombre déterminé de dimensions, chaque dimension étant liée à un terme présent dans le vocabulaire du document. Un vecteur de poids est ensuite assigné à chaque document pour

exprimer le degré d'association entre l'élément et le terme spécifique. Par exemple, considérons $D = \{d_1, d_2, \dots, d_n\}$ comme l'ensemble des documents et $T = \{t_1, t_2, \dots, t_N\}$ comme le dictionnaire des termes, représentant tous les mots du corpus. Ce dictionnaire est établi en appliquant diverses techniques de traitement du langage naturel telles que la tokenisation, l'élimination des mots vides et le stemming[152]. Chaque document est représenté par un vecteur dans un espace à n dimensions, tel que $d_j = \{w_{1j}, w_{2j}, \dots, w_{nj}\}$, où w_{kj} est le poids du terme tk dans le document d_j.

Selon Lops et al.[151], Il est possible d'observer que certains termes apparaissent fréquemment dans un document par rapport au reste du corpus, suggérant ainsi que ces termes soient centraux pour le sujet traité dans ce document. En outre, il est important de normaliser les vecteurs de résultats pour s'assurer que les documents plus longs n'ont pas plus de chances d'être retrouvés que les documents plus courts. La fonction TF-IDF répond efficacement à ces deux questions :

$$TFIDF(t_k, d_j) = TF(t_k, d_j) \times \log\left(\frac{N}{n_k}\right) \tag{10}$$

Dans la formule TF-IDF, N représente le nombre total de documents dans le corpus, et $n_k$représente le nombre de documents dans lesquels le terme $t_k$apparaît au moins une fois avec :

$$TFIDF(t_k, d_j) = \frac{f_{k,j}}{max_z f_{z,j}} \tag{11}$$

Pour s'assurer que tous les poids se situent dans l'intervalle [0,1] et que tous les documents sont représentés par des vecteurs de même longueur, les poids TF-IDF sont généralement normalisés à l'aide de la normalisation cosinus :

$$w_{k,j} = \frac{TFIDF(t_k, d_j)}{\sqrt{\sum_{s=1}^{|T|} TFIDF(t_k, d_j)^2}} \tag{12}$$

Après avoir achevé le processus de pondération et de normalisation des termes, il est essentiel d'établir une méthode pour évaluer la similitude des vecteurs de caractéristiques. Cette évaluation est cruciale pour évaluer la proximité entre deux documents. Parmi les différentes méthodes disponibles, la mesure de similitude cosinus est largement privilégiée dans la documentation académique :

$$sim(d_i d_j) = \frac{\sum_k w_{ki} w_{kj}}{\sqrt{\sum_k w_{ki}^2} \times \sqrt{\sum_k w_{kj}^2}} \tag{13}$$

Dans les systèmes de recommandation de contenu, basés sur l'espace vectoriel, les profils d'utilisateurs et les éléments sont représentés par des vecteurs de termes pondérés. Le profil de l'utilisateur, appelé ContentBasedProfile(u), est créé en analysant le contenu des éléments que l'utilisateur a précédemment évalués. Généralement, ce profil est défini comme un vecteur de mots-clés, en utilisant des techniques de recherche d'informations. Plus formellement, le profil basé sur le contenu (u) est représenté par un vecteur de poids (w_{u1} , w_{u2}, ..., w_{ku}), où chaque poids w_{iu} indique l'importance de l'attribut k_i par rapport aux préférences de l'utilisateur u. Diverses techniques peuvent être employées pour obtenir ce vecteur de poids à partir des vecteurs de contenu des éléments évalués.

III.3.2.1.3. Formes particulières de recommandation basée sur le contenu

A. Recommandation basée sur la connaissance

L'idée qui sous-tend la recommandation basée sur la connaissance est de rassembler des informations complètes sur un utilisateur afin de lui fournir des recommandations pertinentes[153]. Cette approche est analogue à celle d'un ami qui recommande quelque chose en se basant sur sa connaissance intime de la personne, plutôt que sur ses préférences. Pour l'essentiel, les systèmes de recommandation basés sur la connaissance s'appuient sur les informations explicites fournies par l'utilisateur.

B. Recommandation basée sur l'utilité

Les recommandations fournies par les systèmes de recommandation basés sur l'utilité reposent sur le calcul de l'utilité de chaque élément pour l'utilisateur. La principale difficulté consiste à créer une fonction d'utilité qui représente précisément les préférences de l'utilisateur[154]. Cette fonction est essentiellement le profil de l'utilisateur que le système obtient de l'utilisateur. Une approche courante consiste à demander aux utilisateurs de remplir des formulaires qui reflètent leurs préférences[148].

III.3.2.2. Recommandation par filtrage collaboratif

Le filtrage collaboratif repose sur l'échange d'avis entre utilisateurs pour élaborer des recommandations. Ce concept s'inspire du principe du "bouche à oreille", une pratique humaine courante pour se faire une idée sur des produits ou des services inconnus. L'hypothèse clé du filtrage collaboratif est que si deux utilisateurs ont les mêmes préférences pour un ensemble d'articles, ils sont susceptibles d'avoir des préférences similaires pour d'autres articles qu'ils n'ont

pas encore évalués. Par exemple, si les voisins de Mohammed ont eu une expérience positive dans un nouveau restaurant asiatique de son quartier, il est probable que Mohammed l'appréciera également. Les techniques de filtrage collaboratif s'appuient sur l'identification d'utilisateurs similaires pour fournir des recommandations à l'utilisateur actuel. Le filtrage collaboratif se divise généralement en deux grandes catégories : les approches qui s'appuient sur l'historique des interactions et celles qui construisent un modèle prédictif.

Les algorithmes basés sur un modèle, quant à eux, construisent une représentation réduite de la matrice des notes hors ligne pour traiter les notes manquantes, et réduire la complexité des calculs. Ces algorithmes passent d'abord par une phase d'apprentissage pour identifier des modèles dans les données, puis utilisent ces modèles pour formuler des recommandations. Certains des algorithmes de recommandation basés sur des modèles les plus performants sont des méthodes de réduction des dimensions telles que la décomposition de la valeur singulière (SVD : Singular Value Decomposition), des approches probabilistes, des approches basées sur le regroupement et des approches basées sur les règles d'association.

III.3.2.2.1. Filtrage collaboratif basé sur les voisins

Les algorithmes de filtrage collaboratif basés sur les voisins visent à formuler des recommandations en utilisant l'ensemble de la matrice des évaluations des utilisateurs. Une approche courante est celle des k-voisins les plus proches (ou kNN), dans laquelle l'algorithme identifie les k-utilisateurs (ou éléments) les plus similaires à l'utilisateur actif sur la base de leurs évaluations passées. Ces approches peuvent être regroupées en deux familles : celles basées sur l'utilisateur (filtrage collaboratif utilisateur-utilisateur) et celles basées sur l'élément (filtrage collaboratif élément-élément).

Pour les algorithmes basés sur l'utilisateur, l'évaluation estimée d'un élément pour un utilisateur est prédite en utilisant les évaluations de ses utilisateurs similaires. Les utilisateurs similaires sont ceux qui partagent les mêmes préférences que l'utilisateur actif. En revanche, les algorithmes basés sur les éléments, tels que ceux décrits dans [155] prédisent l'évaluation d'un élément pour un utilisateur en se basant sur les évaluations des éléments voisins.

Pour désigner la moyenne des notes attribuées par un utilisateur aux éléments qu'il a notés, on utilise la notation $\bar{r}_u$, qui est calculée à l'aide de la formule 14. De même, la moyenne des notes reçues par un élément est désignée par $\bar{r}_i$ et est calculée à l'aide de la formule 15.

$$\bar{r}_u = \frac{\sum_{i \in I_u} r_{u,i}}{|I_u|} \quad (15)$$

$$\bar{r}_i = \frac{\sum_{u \in U_i} r_{u,i}}{|U_i|} \quad (16)$$

La fonction mesurant la similarité entre deux utilisateurs u et v est représentée par $sim(u, v)$, tandis que $sim(i, j)$ mesure la similarité entre deux objets i et j. Nous définissons $I_{uv} = I_u \cap I_v$ comme l'ensemble des objets qui ont été évalués par les utilisateurs u et v. De même, $U_{ij} = U_i \cap U_j$ représente l'ensemble des utilisateurs qui ont évalué à la fois les objets i et j.

A. Filtrage basé sur les utilisateurs

Le concept de filtrage collaboratif basé sur l'utilisateur a été initialement introduit dans le système GroupLens[156]. Cette méthode suit une approche simple consistant à identifier les utilisateurs similaires à l'utilisateur actif, puis à calculer une valeur de prédiction pour chaque élément sur la base des évaluations exprimées par les voisins de l'utilisateur actif sur cet élément.

Calcul de la similarité

Il existe deux méthodes couramment utilisées pour mesurer la similarité entre deux utilisateurs u et v dans le filtrage collaboratif basé sur l'utilisateur : La similarité cosinus et le coefficient de corrélation de Pearson. Malgré les avantages et les inconvénients propres à chacune de ces approches, il est courant de constater que le coefficient de Pearson se distingue comme le plus prévalent dans l'état de l'art. Selon les conclusions d'une recherche menée par Schafer et al.[157], il est réputé pour sa capacité à fournir des recommandations pertinentes.

Calcul de la prédiction

Pour estimer la prédiction d'un utilisateur u, une méthode basique implique le calcul de la moyenne des évaluations attribuées par tous les utilisateurs voisins de u, comme illustré dans l'équation suivante :

$$pred(u, i) = \frac{\sum_{w \in voisins(u) \cap U_i} r_{w,i}}{|voisins(u) \cap U_i|} \quad (17)$$

La formule (17) a été critiquée du fait qu'elle considére tous les voisins de la même manière et ne prend pas en compte les différents degrés de similarité qui existent entre l'utilisateur actuel u et chacun de ses voisins Schafer et al.. Pour résoudre ce problème, un système de pondération est utilisé pour évaluer la similarité entre l'utilisateur actuel et ses voisins. Les scores des voisins les

plus similaires sont davantage pris en compte. Pour obtenir une prédiction, le score pondéré est divisé par la somme des similarités. L'équation (18) décrit cette formule:

$$pred(u,i) = \frac{\sum_{w \in voisins(u) \cap U_i} sim(w,u) \times r_{w,i}}{\sum_{w \in voisins(u) \cap U_i} | \sum_{w \in voisins(u) \cap U_i} sim(w,u) |} \quad (18)$$

En outre, il est important de reconnaître que les utilisateurs ont des tendances différentes en matière d'évaluation des éléments. Certains utilisateurs ont une échelle d'évaluation plus large, où ils attribuent une valeur de 5 pour indiquer leur satisfaction, tandis que d'autres ont des modèles d'évaluation plus stricts et peuvent attribuer une valeur de 3 pour exprimer leur satisfaction à l'égard d'un élément. Pour tenir compte de ces variations dans les jugements des utilisateurs, le score de chaque utilisateur w est ajusté en fonction de sa note moyenne $\bar{r}_u$. La formule finale, adoptée par les auteurs de GroupLens[156], est présentée dans l'équation ci-dessous pour le calcul de la prédiction :

$$pred(u,i) = \bar{r}_u + \frac{\sum_{w \in voisins(u) \cap U_i} sim(w,u) \times (r_{w,i} - \bar{r}_w)}{\sum_{w \in voisins(u) \cap U_i} |sim(w,u)|} \quad (19)$$

III.3.2.2.2. Filtrage basé sur les items

Le concept de filtrage collaboratif basé sur les éléments a été introduit par Sarwar et al.[155]. Dans cette méthode, la prédiction du score u d'un utilisateur pour un élément spécifique i est calculée sur la base des scores des éléments voisins ou similaires. Le principe de fonctionnement du filtrage collaboratif basé sur les éléments consiste à trouver les éléments voisins ou similaires les plus proches de l'élément candidat i en mesurant sa similarité avec les autres éléments disponibles. Après avoir identifié les éléments similaires, la prédiction de la note u de l'utilisateur pour l'élément i est calculée sur la base des notes attribuées par l'utilisateur aux éléments voisins de i.

Calcul de la similarité : La similarité entre deux éléments i et j peut être calculée à l'aide du cosinus, du coefficient de Pearson ou du cosinus ajusté. Toutefois, une étude menée par Sarwar et al.[155] a révélé que le cosinus ajusté est la méthode la plus efficace en termes de précision de la prédiction.

Calcul de prédiction : Pour prédire le score u de l'utilisateur actuel pour un élément recommandé i, une moyenne pondérée de ses scores sur tous les éléments similaires à i est calculée. Chaque score, $r_{u,j}$, est multiplié par la similarité entre l'élément j et l'élément i. Pour que la prédiction se situe dans la même fourchette de valeurs que les scores, elle est divisée par la somme des similarités. Comme il s'agit du même utilisateur, il n'est pas nécessaire d'ajuster le score. La formule utilisée par Sarwar et al.[155] pour le calcul de la prédiction est donnée par l'équation 20:

$$pred(u,i) = \frac{\sum_{i \in I_u} sim(i,j) \times r_{u,j}}{\sum_{i \in U_i} |sim(i,j)|} \quad (20)$$

L'importance du nombre de voisins dans le filtrage collaboratif basé sur les articles est comparable à celle du filtrage basé sur l'utilisateur, car il influence considérablement la précision des prédictions. Afin d'améliorer la qualité des recommandations et de réduire la complexité des calculs, Sarwar et al.[155] suggèrent de ne prendre en compte que les k plus proches voisins de l'article i parmi ceux qui ont été évalués par l'utilisateur u. La valeur optimale de k peut être déterminée par l'expérimentation.

III.3.2.2.3. Différents types de calcul de la similarité

L'objectif du calcul de similarité est de déterminer le degré de similitude entre deux utilisateurs ou deux éléments. Il existe plusieurs méthodes de calcul de la similarité, mais les méthodes les plus couramment utilisées et les plus efficaces sont présentées ci-dessous.

Similitude *cosinusienne* : La similarité cosinus est une mesure largement utilisée pour mesurer la similarité entre deux objets, a et b, dans le cadre de la recherche d'informations[158]. Elle consiste à représenter les deux objets sous forme de vecteurs, $\vec{x}_a$et $\vec{x}_b$et à mesurer le cosinus de l'angle formé par les deux vecteurs :

$$sim(a,b) = \cos(\vec{x}_a, \vec{x}_b) = \frac{\vec{x}_a . \vec{x}_b}{\| \vec{x}_a \| \| \vec{x}_b \|} \quad (21)$$

Dans le cadre du filtrage collaboratif, chaque utilisateur u est représenté par un vecteur $\vec{x}_u$, où $\vec{x}_u$= $r_{u,i}$. Pour calculer la similarité entre deux utilisateurs u et v, le cosinus est évalué sur l'ensemble des éléments évalués par les deux utilisateurs. La formule 21 est utilisée à cette fin :

$$sim(u,v) = \cos(\vec{x}_u, \vec{x}_v) = \frac{\sum_{i \in I_{uv}} r_{u,i} \times r_{v,i}}{\sqrt{\sum_{i \in I_{uv}} r_{u,i}^2} \cdot \sqrt{\sum_{i \in I_{uv}} r_{v,i}^2}} \quad (22)$$

Le cosinus peut également être utilisé pour calculer la similitude entre deux éléments. Pour calculer la similarité entre deux articles i et j, la formule 23 peut être appliquée en remplaçant les utilisateurs par les vecteurs d'articles respectifs xi et xj, où xij est l'évaluation de l'utilisateur u pour les deux articles i et j.

$$sim(i,j) = \cos(\vec{x}_i, \vec{x}_j) = \frac{\sum_{u \in U_{ij}} r_{u,i} \times r_{u,j}}{\sqrt{\sum_{u \in I_{ij}} r_{u,i}^2} \cdot \sqrt{\sum_{i \in I_{ij}} r_{u,j}^2}} \quad (23)$$

Le score de similarité cosinus varie de 0 à 1, un score de 1 indiquant que deux utilisateurs ont les mêmes préférences et un score de 0 indiquant qu'ils n'ont rien en commun. Cependant, l'une des limites de l'utilisation du score de similarité cosinus dans le filtrage collaboratif est qu'il ne prends pas en considération de la variabilité des évaluations des utilisateurs.

Le coefficient de corrélation de *Pearson* est une autre méthode couramment utilisée pour calculer la similarité entre deux utilisateurs ou deux éléments. Il a été initialement introduit dans le système GroupLens par Resnick et al. en 1994. Le coefficient de corrélation de Pearson évalue la concordance entre les évaluations de deux utilisateurs ou éléments en examinant la covariance par rapport au produit de leurs écarts-types. Cette mesure quantifie la similitude entre les éléments évalués par les deux parties concernées. Plus deux utilisateurs ont tendance à noter les mêmes éléments de manière similaire, plus ils sont semblables, comme le montre la formule mathématique ci-dessous[159] :

$$sim(i,j) = Pearson(u,v) = \frac{\sum_{i \in I_{uv}} (r_{u,i} - \bar{r}_u).(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{v,i} - \bar{r}_u)^2} . \sqrt{\sum_{i \in I_{uv}} (r_{v,i} - \bar{r}_v)^2}} \quad (24)$$

Le coefficient de corrélation de Pearson peut également être utilisé pour mesurer la similarité entre deux éléments i et j. La formule ci dessous fournit la similarité de Pearson entre deux éléments :

$$sim(i,j) = Pearson(i,j) = \frac{\sum_{i \in U_{ij}} (r_{u,i} - \bar{r}_i).(r_{u,j} - \bar{r}_j)}{\sqrt{\sum_{i \in U_{ij}} (r_{u,i} - \bar{r}_i)^2} . \sqrt{\sum_{i \in U_{ij}} (r_{u,i} - \bar{r}_j)^2}} \quad (25)$$

La méthode du *cosinus ajusté* est utilisée pour calculer la similarité entre deux éléments dans le cadre du filtrage collaboratif. Alors que la méthode de similarité de Pearson centre les scores d'un même utilisateur par rapport à la moyenne de leurs scores, la méthode du Cosinus ajusté ajuste les scores par rapport à la moyenne des utilisateurs. En effet, la variation entre les différents utilisateurs est plus importante que la variation des scores pour un même utilisateur. Le cosinus ajusté a été introduit par Sarwar et al. et est considéré par beaucoup comme l'une des méthodes les plus efficaces et les plus populaires pour calculer la similarité entre deux éléments dans le filtrage collaboratif, comme l'indiquent Schafer et al.[157], [160].

$$sim(i,j) = Adjustde\_cosine(i,j) = \frac{\sum_{u \in U_{ij}} (r_{u,i} - \bar{r}_u).(r_{u,j} - \bar{r}_u)}{\sqrt{\sum_{u \in U_{ij}} (r_{u,i} - \bar{r}_u)^2} . \sqrt{\sum_{u \in U_{ij}} (r_{u,j} - \bar{r}_u)^2}} \quad (26)$$

III.3.2.3. Méthode pour réduire la dimension

Les techniques de réduction de dimensions servent à adresser les défis liés à la sensibilité des données manquantes et à la mise à l'échelle. Elles opèrent en projetant les éléments (ou les utilisateurs) dans un espace de dimensions réduites, caractérisé par des variables latentes. Cet espace plus dense et plus petit permet de comparer les utilisateurs (ou les éléments) entre eux et de découvrir de nouvelles relations, même s'ils n'ont aucun élément en commun[161], [162]. Les méthodes de factorisation matricielle, telles que l'analyse en composantes principales (ACP)[163] ou la décomposition en valeurs singulières (SVD) [164], [165], sont utilisées pour extraire les variables latentes. Ces techniques sont principalement employées dans le but soit de diminuer la taille de la matrice, soit de réduire la dimension de la matrice de similitude[166].

L'analyse sémantique latente (LSA), également connue sous le nom d'indexation sémantique latente (LSI), est une technique de réduction de la dimension largement utilisée dans le domaine de la recherche d'informations et du filtrage collaboratif[167]. La LSA a été appliquée pour réduire la dimensionnalité de la matrice R-score dans les algorithmes de filtrage collaboratif . LSA approxime la matrice de rang $R_{|U|,|I|}$ par une matrice de rang $R' = PQ^t$avec $k \ll r$, où $P_{|U|,k}$, k est une matrice représentant les utilisateurs dans l'espace réduit et $Q_{|U|,k}$ est une matrice représentant les éléments dans l'espace réduit, P et Q étant deux matrices orthonormées. Pour calculer les matrices , P et Q, une décomposition en valeurs singulières (SVD) est appliquée à la matrice de score R, qui la décompose en trois matrices , $D, \sum$ et T.

$$R \approx R' = D_{|U|,k} \times \sum_{k,k} \times T^t_{k,|I|} \tag{27}$$

Après avoir obtenu les matrices P et Q par décomposition en valeurs singulières (SVD), chaque utilisateur de la matrice $P_{|U|,k}$ est représenté par un ensemble de k variables latentes, au lieu de tous ses scores, tandis que chaque élément de $Q_{|I|,k}$est représenté par k variables latentes au lieu des scores des utilisateurs. La prédiction du score u de l'utilisateur pour l'élément i est alors donnée par :

$$pred(u,i) = p_u q_i^t \tag{28}$$

L'application de la SVD à la matrice R-score peut s'avérer difficile en raison des données manquantes. En effet, la SVD ne peut pas être appliquée directement à une matrice comportant des valeurs manquantes. Pour résoudre ce problème, une solution consiste à remplacer les valeurs

manquantes par des valeurs par défaut. Toutefois, cela peut introduire un biais dans les données, comme l'ont noté Desrosiers et Karypis en 2011[166].

III.3.2.4. Modèles probabilistes

Ces approches sont basées sur le concept de modélisation du comportement de l'utilisateur à l'aide d'un modèle probabiliste pour prédire ses actions futures. Le principe sous-jacent de ces algorithmes consiste à calculer la probabilité $P(v|u,i)$, qui est la probabilité que l'utilisateur u attribue la note v à l'élément i sur la base de ses évaluations précédentes. La prédiction, soit au score avec la probabilité la plus élevée, soit le score attendu, est déterminée à l'aide de la formule 28 chafer et al.[157].

$$pred(u,i) = E(v|u,i) = \sum_{u \in V} v.P(v|u,i) \tag{29}$$

Cette catégorie d'algorithmes utilise des modèles probabilistes pour prédire le comportement des utilisateurs. L'idée principale de ces algorithmes est de calculer la probabilité $P(v|u,i)$, qu'un utilisateur u attribue une note v à un élément i, sur la base de ses notes précédentes. Le score prédit, désigné par $pred(u,i)$, peut être soit le score le plus probable, soit le score attendu calculé à l'aide de la formule 28[160]. Les valeurs de score v sont généralement tirées d'un ensemble de valeurs possibles, souvent comprises entre 1 et 5.

Un exemple de ce type d'algorithme est Cross Sel[168], qui utilise un classificateur bayésien naïf pour recommander des articles. Il estime la probabilité qu'un utilisateur achète l'article j s'il a déjà acheté l'article i, sur la base de son historique d'achat. On peut également donner l'exemple du travail de Breese et al[169], qui propose un modèle basé sur les réseaux bayésiens et les arbres de décision pour calculer les probabilités. Il s'agit de l'un des premiers travaux à appliquer un modèle probabiliste aux algorithmes de filtrage collaboratif.

III.3.2.3. Recommandation Hybride

Les systèmes basés sur le contenu sont efficaces pour fournir des recommandations dans les cas d'interaction limitée de l'utilisateur. Cependant, leur précision est inférieure à celle des méthodes de filtrage collaboratif, qui ne tiennent pas compte des détails du contenu. Un système hybride intégrant ces approches offre une solution prometteuse, combinant les avantages des deux méthodes. Ces systèmes, intégrant des informations sur les films telles que le genre, le résumé et les acteurs, au profil de l'utilisateur, visent à améliorer la pertinence des recommandations[150],

[170]. D'autres modèles utilisent une somme pondérée pour intégrer les recommandations de différents modèles[171], un système de vote où chaque modèle recommande un élément[172], ou alternent la responsabilité du système de recommandation entre les modèles basés sur le contenu et les modèles de filtrage collaboratif[173]. Une autre approche consiste à reclasser la liste des recommandations d'un modèle en utilisant les résultats d'un autre modèle pour affiner les recommandations[147]. Enfin, une approche plus sophistiquée utilisant des machines de Boltzmann a été proposée pour apprendre automatiquement l'hybridation afin d'optimiser la précision du système de recommandation[174].

Comme les plateformes recueillent continuellement de nouvelles informations sur les utilisateurs, telles que leurs connexions sur les réseaux sociaux, leur localisation géographique et autres, l'ajout de nouvelles fonctionnalités au cours du processus d'apprentissage devient crucial. Cette flexibilité dans l'intégration de nouvelles caractéristiques est une force importante des réseaux neuronaux, qui peut être considérée comme une forme d'hybridation en soi[175].

III.3.2.4. Systèmes de recommandation sensibles au contexte

La nécessité d'intégrer les informations contextuelles dans les systèmes de recommandations est de plus en plus reconnue par les chercheurs dans divers domaines. Les données contextuelles telles que l'heure, la localisation et l'environnement social peuvent considérablement améliorer le processus de recommandation, en particulier dans des domaines comme le tourisme. Alors que les systèmes de recommandation traditionnels se concentrent généralement sur les utilisateurs et les articles, l'intégration des informations contextuelles est cruciale pour améliorer leur efficacité dans certains contextes.

III.3.2.4.1. Définition

Il est difficile de définir le contexte de manière exhaustive et pratique[176] présentent l'une des définitions les plus largement acceptées : "Les données de contexte englobent toutes les informations utiles pour définir la situation entourant une entité. Cette dernière peut être une personne, un lieu ou un objet, jugé pertinent pour l'interaction entre un utilisateur et une application. Cela inclut non seulement l'utilisateur et les applications elles-mêmes, mais également tout ce qui influe sur leur interaction".

Cette définition a toutefois été critiquée par certains chercheurs[177], ont émis des critiques à l'égard de cette définition, la trouvant trop abstraite et difficile à mettre en pratique. Pour remédier

à cela, ils avancent un modèle de contexte qui s'appuie sur la définition d'Abowd et al., mais qui détaille cinq catégories spécifiques d'informations contextuelles : l'individualité, l'activité, la relation, la temporalité et l'emplacement. Ces différentes dimensions du contexte sont illustrées dans la Figure 25.

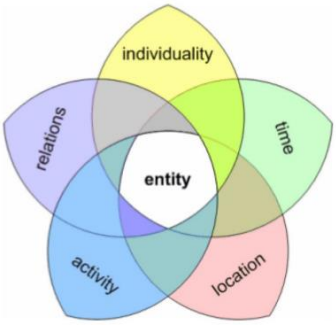

Figure 25 : Eléments définissant le Contexte[177]

III.3.2.4.3. L'importance de la modélisation du contexte

Les systèmes de recommandation traditionnels opèrent généralement dans un espace bidimensionnel, se concentrant sur les utilisateurs et les objets. Cependant, les systèmes de recommandation contextuels ajoutent une dimension supplémentaire : le contexte. En intégrant le contexte, ces systèmes cherchent à améliorer la qualité des recommandations en utilisant des fonctions de score adaptées:

$$R : User \times Item \times Contexte \rightarrow Rating \qquad (30)$$

Pour illustrer ce concept prenons un exemple de recommandation d'un match de Karaté dans des salle à plusieurs tapis(tatami), où on peut décrire les attributs des utilisateurs et les items comme suit :

Match : (ID_Match, Catégorie, Durée, Année, Genre)

Spectateurs : (ID_spectateur, Nom, Sexe,Adresse, Age)

Les informations contextuelles peuvent être classées en trois types, chacun décrit par les attributs correspondants : explicites, implicites et déduites :

Tatami : (ID_tatami, ID_Match, Date_match)

Temps : (JourSemaine, Weekend)

Compagnon : (Type_compagnon : prend une des valeurs " seul ", " amis ", " famille ", " collègues ")

Palmisano et al. [178] décrivent le contexte comme un ensemble de dimensions contextuelles K, où chaque dimension $k \in K$ possède un ensemble de q attributs, $k = \{k^1, k^2, \ldots, k^q\}$. Ces dimensions capturent différents aspects du contexte et ont une structure hiérarchique.

Selon Adomavicius et Tuzhilin[179], Trois approches principales sont utilisées pour intégrer des informations contextuelles dans un système de recommandation, selon le moment où le contexte est pris en compte. Ces stratégies sont dénommées pré-filtrage contextuel, post-filtrage contextuel et modélisation contextuelle.

III.4. Conclusion

La recherche sur l'efficacité énergétique et l'optimisation des ressources virtuelles dans les centres de données cloud est cruciale pour répondre aux défis croissants de consommation énergétique. Les efforts visent à minimiser l'utilisation des serveurs et à maximiser l'efficacité énergétique. Une variété de techniques est explorée pour atteindre cet objectif, allant des méthodes d'analyse de trafic simples aux approches plus sophistiquées qui intègrent le contrôle énergétique. L'accent est mis sur la virtualisation des serveurs pour exploiter leur puissance croissante, mais les résultats varient en fonction des environnements de test. En définitive, l'objectif est de minimiser les serveurs actifs tout en maximisant l'utilisation de l'énergie à travers l'ensemble du centre de données.

Ces dernières années, les systèmes de recommandations ont gagné en importance dans plusieurs domaines, aidant les utilisateurs à trouver des ressources adaptées à leurs besoins parmi une multitude d'options. Ils se divisent en deux catégories principales : basés sur le contenu et collaboratifs. D'autres approches peuvent être regroupées selon cette classification. Cette étude offre un aperçu des diverses approches pouvant être appliquées à l'optimisation du cloud.

CHAPITRE IV : OPTIMISATION DU CLOUD PAR LES SYSTEMES DE RECOMMANDATIONS : ARCHITECTURE BIG DATA

IV.1. Résumé

Les systèmes de recommandation se sont imposés comme des outils incontournables pour offrir un contenu pertinent aux utilisateurs, qu'il s'agisse de réseaux sociaux, de santé, d'éducation ou d'élections. Par ailleurs, avec l'essor fulgurant du Cloud Computing, du Big Data et de l'Internet des objets (IoT), il est primordial que les élections soient supervisées par des organismes de gestion électorale transparents, responsables, neutres et autonomes. L'utilisation de la technologie dans les procédures de vote peut les rendre plus rapides, plus efficaces et moins sensibles aux failles de sécurité. La technologie peut garantir la sécurité de chaque vote, un comptage et un décompte automatiques meilleurs et plus rapides, ainsi qu'une plus grande précision. Les données électorales ont été combinées par différents sites web et applications. En outre, elles ont été interprétées à l'aide de nombreux algorithmes de recommandation tels que les algorithmes d'apprentissage automatique, les algorithmes de représentation vectorielle, les algorithmes de modèle à facteurs latents et les méthodes de voisinage, et partagées avec les organes de gestion des élections afin de fournir des recommandations appropriées. Dans ce chapitre, nous procédons à une étude comparative approfondie des algorithmes employés dans les recommandations avec une architecture du Big Data. Les résultats nous montrent que le modèle K-NN fonctionne le mieux avec une précision de 96%. En outre, nous avons élaboré un système de recommandation hybride intégrant à la fois le filtrage basé sur le contenu et le filtrage collaboratif. Ce système exploite les analogies entre les utilisateurs et les objets pour proposer des suggestions d'une pertinence optimale.

Mots-clés : Cloud computing, Big Data, IoT, système de recommandation et algorithme KNN.

IV.2. Introduction

Aujourd'hui, de nombreux pays tentent de mettre en place des lois pour permettre au public d'exprimer plus facilement son opinion sur une question donnée et surtout de choisir ses dirigeants par le biais du vote. La procédure conventionnelle est le vote manuel ou sur papier qui, parfois, ne facilite pas le décompte et n'est pas pratique pour les électeurs, ce qui diminue parfois le taux de participation. Avec l'évolution des TIC, le vote électronique est devenu une nécessité pour surmonter ces problèmes et faciliter le vote dans toute zone géographique. Cette évolution technologique a généré une explosion d'une grande masse de données numériques, ce qui rend

impératif de trouver des solutions pour permettre un stockage et une analyse adéquats, donnant ainsi naissance au concept de Big Data. En effet, le Big Data nécessite d'énormes capacités de stockage et de traitement que les méthodes conventionnelles ne peuvent offrir. Pour favoriser le déploiement du Big Data, la technologie Cloud Computing permet l'accès et l'utilisation de ressources informatiques telles que les unités de calculs, le stockage et le réseau à la demande. En d'autres termes, il fournit aux utilisateurs des services informatiques qui leur permettent de sauvegarder et de manipuler des données, d'installer des modèles et d'accéder à de multiples ressources sur le web, en fonction de leurs besoins.

L'une des caractéristiques les plus avantageuses du cloud est son élasticité, car elle permet d'adapter la capacité de calcul à la demande et de n'allouer que les ressources nécessaires[64]: Les services cloud les plus populaires sont Amazon web services, SAP (System Applications and Products), Oracle, Cloudera, Data-bricks, etc. L'augmentation rapide du volume de données numériques sur les sites web et les applications dans le monde entier a conduit au problème de la surcharge de données, qui rend difficile l'extraction d'informations pertinentes. En réponse à ce problème, des algorithmes d'optimisation et des systèmes de recommandation ont été proposés et les différents services mentionnés ci-dessus servent de plateforme pour le développement et l'évaluation des systèmes de recommandation.

Lorsque les moteurs de recherche ne suffisent pas à traiter cette grande quantité d'informations en ligne, les systèmes de recommandation tentent de deviner les informations ou les biens et services dont nous avons réellement besoin. Tout SR doit connaître le profil de l'utilisateur, qui contient par exemple ses préférences. Ce profil peut être acquis implicitement en analysant le comportement passé de l'utilisateur ou explicitement en demandant à l'utilisateur quelles sont ses préférences. Dans ce cas, les SR qui prennent en compte une communauté d'utilisateurs sont souvent appelés approches collaboratives[180], tandis que ceux qui se basent sur la description des articles sont appelés approches basées sur le contenu[181]. L'approche hybride[182] est une combinaison de deux méthodologies ou systèmes différents qui vise à créer un nouveau modèle et qui est fortement ancrée dans la recherche et le filtrage d'informations. Les préférences de l'utilisateur et les descriptions des articles sont en quelque sorte compatibles, de sorte qu'il est possible de les comparer et de trouver un article qui corresponde aux préférences de l'utilisateur.

L'une des questions centrales des systèmes de recommandation est de savoir comment modéliser et comparer les préférences des utilisateurs et les descriptions d'articles de manière souple et

naturelle. Les préférences des utilisateurs, tout comme les descriptions des articles, peuvent être imprécises, incomplètes, subjectives et d'importance différente. La première approche est basée sur l'utilisation de la théorie des ensembles flous adoptée par Yager[183]. Une autre approche, qui utilise le regroupement flou comme outil de recommandation, a été présentée par d'autres auteurs[184]. Lors des élections politiques, les électeurs doivent décider pour qui voter. Cette décision est devenue particulièrement difficile ces derniers temps, car on accorde beaucoup plus d'attention aux personnalités, aux images et aux événements de la campagne qu'aux manifestes des partis et aux questions politiques. En outre, diverses applications ont été créées pour aider les électeurs à choisir facilement leurs candidats préférés. L'idée générale de ces systèmes est de mettre x sur les symboles des deux candidats. Enfin, ces choix sont comparés pour sélectionner les candidats accessibles.

Dans ce chapitre, nous proposons une vue panoramique des divers algorithmes employés dans les systèmes de recommandation ainsi que des multiples instruments exploités dans l'analyse du Big data. De surcroît, nous effectuons une analyse comparative de certains de ces algorithmes lorsqu'ils sont appliqués aux systèmes de recommandation associés à un processus de vote. Le reste du chapitre est organisé comme suit. La section 2 présente les différents outils utilisés dans l'analyse des Big Data et un bref aperçu des systèmes de recommandation. La section 3 présente certains des travaux réalisés sur ce sujet. La section 4 présente la méthodologie utilisée dans notre travail. La section 5 présente les résultats obtenus dans ce chapitre. Nous terminons par une conclusion.

IV.3. Revue de la littérature

IV.3.1. Cadre des Big Data

HDFS : Le système de fichiers distribués Hadoop (HDFS) est une architecture composée d'un NameNode, un serveur qui régule l'accès aux fichiers par les clients et gère l'espace de noms du système de fichiers. HDFS permet de stocker les données des utilisateurs dans des fichiers par l'intermédiaire d'un espace de noms de fichiers. Dans HDFS, l'ouverture, la fermeture et le renommage des fichiers et des répertoires sont des opérations effectuées par le Na-meNode et déterminent également le mappage des blocs aux Datanodes[182]. HDFS peut s'adapter à des applications avec des ensembles de données dont la taille varie de quelques gigaoctets à quelques téraoctets. Il peut également s'adapter à des centaines de nœuds dans un seul cluster et peut fournir une large bande passante pour les données agrégées[185].

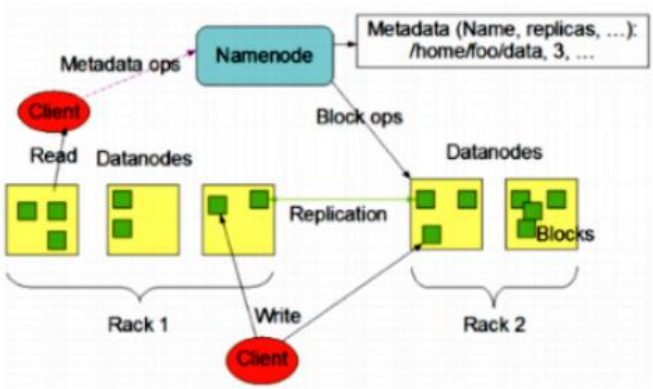

Figure 26 : Architecture de HDFS[185]

Spark : Dans le domaine de l'analyse avancée des données (Big Data), Spark est le framework de traitement distribué en mémoire le plus populaire. Ce succès s'explique en grande partie par son API conviviale et son ensemble de fonctionnalités robustes, permettant l'interrogation de vastes volumes de données via SQL et la création de modèles complexes d'apprentissage automatique (ML) grâce aux algorithmes les plus couramment utilisés. Spark réduit également le temps d'exécution d'une application en réduisant le nombre d'opérations de lecture/écriture sur le disque[183]. Malgré ces avantages, il peut masquer certains des problèmes causés par un mécanisme d'exécution de code distribué, ce qui rend les tests inefficaces. La création de conteneurs Dockers est le moyen le plus efficace de créer un environnement de test qui ressemble à l'environnement du cluster. L'avantage de cette approche est que les applications peuvent être testées sur différentes versions du Framework en changeant simplement quelques paramètres dans le fichier de configuration[184].

Un autre avantage est qu'il est conçu pour être exécuté sur différents types de clusters. Les gestionnaires de clusters tels que Yet Another Resource Negotiator (YARN), Hadoop Platform Resource Manager, Mesos ou Kubernetes sont pris en charge par spark[182]. La figure montre l'architecture de HDFS[181].

MapReduce. L'énorme quantité de données offre aux entreprises de nombreuses opportunités. L'un des plus grands problèmes est de pouvoir les traiter efficacement sur les systèmes traditionnels et donc de se tourner vers de nouvelles solutions spécialement conçues à cet effet.MapReduce facilite le traitement des données. MapReduce facilite le traitement des données. Il décompose les volumes massifs de données en plus petits morceaux. Ces morceaux de données

sont ensuite traités en parallèle sur les serveurs Hadoop. Il envoie ensuite le résultat consolidé à l'application[186].

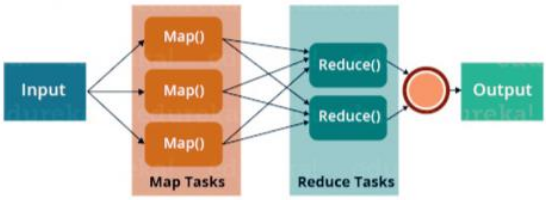

Figure 27 : Exécution de la tâche MapReduce2[186]

IV.3.2. Systèmes de recommandation

Avec l'évolution de la technologie et en particulier l'avènement du web, la quantité de données à exploiter et à traiter est devenue très importante, ce qui rend la recherche et la découverte difficiles. Les systèmes de recommandation constituent l'une des techniques informatiques mises au point pour extraire des informations pertinentes. Dans les systèmes de recommandations, la valeur d'un élément est souvent exprimée par un score reflétant le niveau d'appréciation de l'utilisateur pour cet élément. Le défi consiste à estimer les scores des éléments que l'utilisateur n'a pas encore évalués. Lorsqu'il est possible d'estimer les scores des éléments qui n'ont pas encore été évalués, les utilisateurs se voient recommander des éléments dont le score estimé est plus élevé[187].

Les systèmes de recommandations sont des systèmes qui produisent des recommandations personnalisées pour guider l'utilisateur de manière personnalisée vers des éléments intéressants ou utiles lorsque de nombreuses options sont disponibles. Ils comportent trois phases : la première est la phase de collecte des données, qui consiste à recueillir des informations pertinentes sur un utilisateur ; la deuxième est la phase d'apprentissage, qui utilise les données collectées pour filtrer et exploiter les caractéristiques de l'utilisateur ; et la dernière est la phase de recommandation, qui prédit ou recommande le type de produit que l'utilisateur pourrait préférer.

Filtrage collaboratif : Ce type de recommandation aspire à automatiser les suggestions que des utilisateurs aux affinités communes peuvent se transmettre mutuellement Ils permettent à un utilisateur de trouver les informations qui l'intéressent en se basant sur les mêmes goûts et préférences que les autres utilisateurs. Dans ce type de recommandation, deux méthodes peuvent être utilisées, à savoir la recommandation basée sur l'utilisateur et la recommandation basée sur l'élément.

Par exemple, la recommandation basée sur l'utilisateur est une technique utilisée pour prédire les articles qu'un utilisateur pourrait aimer sur la base des notes attribuées à cet article par d'autres utilisateurs ayant des goûts similaires à ceux de l'utilisateur cible. Dans ce type de recommandation, le poids peut être déduit comme la corrélation entre les utilisateurs, qui est calculée par la formule suivante :

$$Sim(a,b) = \frac{\sum_p (r_{ap} - \bar{r}_a)(r_{bp} - \bar{r}_b)}{\sqrt{\sum_p (r_{ap} - \bar{r}_a)^2}\sqrt{\sum_p (r_{bp} - \bar{r}_b)^2}} \quad (31)$$

Avec :

- r_{ap} et r_{bp} représentent les évaluations de l'utilisateur p pour les items a et b respectivement.
- $\bar{r}_a$ et $\bar{r}_b$ sont les moyennes des évaluations des items a et a respectivement.
- La somme est effectuée sur tous les utilisateurs p ayant évalué à la fois les items a et b.

Dans certains cas, un utilisateur peut présenter des similitudes avec certains utilisateurs et peu de similitudes avec d'autres. Cela signifie que les notes attribuées à un élément particulier par les utilisateurs les plus similaires doivent avoir plus de poids que celles attribuées par les utilisateurs moins similaires, et ainsi de suite. Dans ce cas, l'évaluation de chaque utilisateur est multipliée par un facteur de similarité, qui est donné par la formule 32 :

$$r_{up} = \bar{r}_b + \frac{\sum_{i \in users} sim(u,i) * \bar{r}_{ip}}{\sum_{i \in users} |Sim(u,i)|} \quad (32)$$

Filtrage basé sur le contenu : Pour générer des recommandations, ces types de système de recommandation s'appuient sur les profils des utilisateurs et les descriptions des articles, tout en suggérant à l'utilisateur les articles qui correspondent le mieux à ses préférences sur la base de la similarité. Des profils sont établis pour les utilisateurs ainsi que pour les produits. Ainsi, ces descriptions (profils, utilisateurs) sont représentées sous forme de vecteurs de termes pondérés. La formule (33) calcule les similitudes pour prédire les intérêts des utilisateurs[188], [189].

$$Sim(a,b) = cos\theta = \frac{\vec{a}.\vec{b}}{\|\vec{a}\|.\|\vec{b}\|} = \sum_{i=1}^{n} \frac{a_i b_i}{\sqrt{a_i^2 . b_i^2}} \quad (33)$$

Où :

- $\vec{a}$ et $\vec{b}$: sont des vecteurs représentant les évaluations de deux utilisateurs ou de deux

items.

- $cos\theta$: Représente le cosinus de l'angle θ entre les deux vecteurs $\vec{a}$ et $\vec{b}$. Le cosinus de cet angle donne une mesure de la similarité entre les deux vecteurs : plus il est proche de 1, plus les vecteurs sont similaires.
- $\vec{a}.\vec{b}$: Produit scalaire des vecteurs $\vec{a}$ et $\vec{b}$. Il est calculé en multipliant les éléments correspondants des deux vecteurs et en additionnant les résultats.
- $\|\vec{a}\|.\|\vec{b}\|$: Normes (ou longueurs) des vecteurs $\vec{a}$ et $\vec{b}$. La norme d'un vecteur $\vec{a}$ est calculée comme la racine carrée de la somme des carrés de ses élément. Il en est de même pour le vecteur $\vec{b}$.

Certains auteurs [190] ont utilisé des méthodes probabilistes dans les recommandations basées sur le contenu. La formule ci-dessous représente le modèle classifié bayésien qui vérifie l'appartenance d'un exemple à une classe en tenant compte des attributs.

$$P(C_i)\prod_k P(A_k V_{kj}\, C_i) \tag{34}$$

Où :

- $P(C_i)$: Il s'agit de la probabilité a priori de la classe C_i. En termes de recommandation, cela peut représenter la probabilité initiale qu'un utilisateur soit intéressé par une certaine catégorie ou classe d'items avant de prendre en compte des preuves supplémentaires
- $\prod_k$: Il indique que nous multiplions les termes $P(A_k V_{kj} C_i)$ pour toutes les valeurs de k.
- $P(A_k V_{kj} C_i)$: Probabilité conditionnelle de $A_k V_{kj}$donné par C_i. Cela représente la probabilité de l'attribut A_k renant la valeur V_{kj}, étant donné la classe C_i.

L'autre méthode utilisée dans ce type de recommandation est la Term Frequency-Inverse Document Frequency (TF-IDF), qui permet de déterminer dans quelles proportions certains mots d'un site web, par exemple, ou d'un document, sont évalués par rapport au reste du texte. Par exemple, le critère de fréquence permet d'augmenter la pertinence d'un site, sans que la densité des mots-clés ne joue un rôle majeur. Le TF est défini comme suit[191] :

$$TF = \frac{NumberOfoccurenceOfterm}{TotalNumberOfterms} \tag{35}$$

L'autre mesure se concentrera sur la mesure du mot en fonction de l'utilisation du document plutôt que sur la mesure en fonction de la fréquence. La formule ci-dessous définit l'IDF[191]:

$$IDF = log \frac{TotalNumberOfdocuments}{NumberOfdocumentscontainingterm} \quad (36)$$

La formule 37 permet de calculer le poids du TF :

$$W = IDF * TF \quad (37)$$

Les systèmes de recmmandations hybrides : combinent deux techniques ou plus pour obtenir de meilleures performances, comme le filtrage basé sur le contenu et le filtrage collaboratif. Aujourd'hui, les systèmes de recommandation ont été étudiés dans plusieurs domaines tels que les villes intelligentes[192], l'éducation[193], le commerce électroniqu[194] et l'apprentissage en ligne[195].

IV.4. Les différents travaux de l'état de l'art

Dans le passé, de nombreux chercheurs ont consacré une grande partie de leurs travaux au développement de systèmes électoraux ou de technologies basées sur l'IoT. Les travaux antérieurs étaient principalement axés sur un nouveau système de recommandation de l'IoT qui utilise des données d'application accessibles au public comme source de personnalisation. Dans[186], les auteurs ont proposé un système qui déduit les objets physiques de l'utilisateur à partir de l'exploration des applications installées sur leurs appareils mobiles tels que les smartphones et les tablettes, et qui construit un inventaire numérique des objets physiques de chaque utilisateur.

Ces inventaires peuvent ensuite être utilisés pour créer des recommandations personnalisées. Dans[196], les auteurs ont utilisé des systèmes de recommandation pour les utilisateurs de l'IoT. Ils ont proposé un système de recommandation qui fonctionne sur la base du graphe créé entre les utilisateurs, les objets et les services, tout en recommandant un appareil IoT adapté aux clients en fonction de leurs besoins et de leurs intérêts. Pour recommander la meilleure option aux utilisateurs, ils ont également pris en compte les caractéristiques des utilisateurs et des services. Pour faire face à la tâche difficile de la sélection des services dans CC, les auteurs dans [197] ont proposé un SR pour ces services afin d'aider les différents utilisateurs du CC à mieux trouver ce qu'ils recherchent. Pour générer des recommandations fiables, ils ont adopté l'analyse formelle des concepts flous à l'aide de la représentation en treillis. Cette méthode leur a permis de transformer

le référentiel des différents services cloud en un ensemble de petites grappes, où les relations entre les services de haute qualité et les services de haute qualité sont mises en évidence.

IV.5. Méthodologies appliquées

Le processus de conception du système de recommandation avec la progression de l'Internet des objets (IoT), en l'occurrence le cloud computing, évolue rapidement en utilisant des clusters. Il donne de l'élasticité aux utilisateurs des services de technologie de l'information (IT) associés pour sauvegarder et gérer les données, installer des modèles, et fournir facilement des recommandations au fur et à mesure des meilleurs besoins. D'autre part, nous avons développé une approche hybride de recommandation contextuelle pour optimiser la sélection des candidats. Cette méthode fusionne deux techniques distinctes :

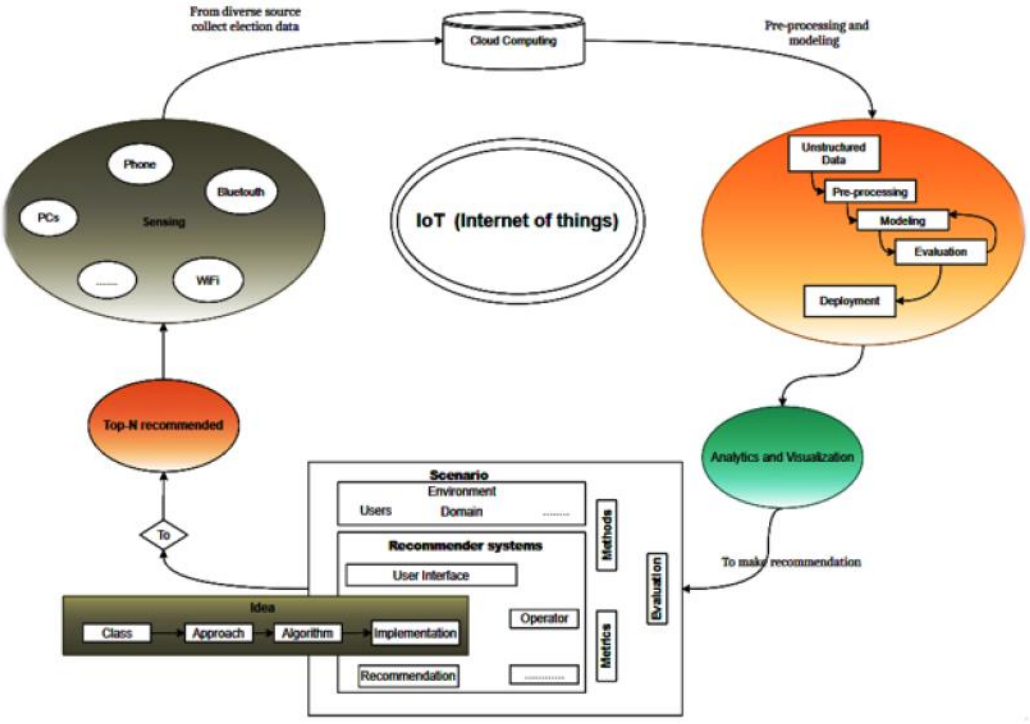

Figure 28 : Architecture proposée

Chaque méthode a été adaptée à une étape spécifique de l'électeur. Tout d'abord, l'approche contient des tentatives de création d'un profil utilisé pour prédire les évaluations sur des éléments non vus. Deuxièmement, l'approche collaborative utilise les produits à l'aide de notes d'utilisateurs (explicites ou implicites) à partir de l'historique donné. Elle fonctionne dans une base de données des préférences des utilisateurs pour les articles (Items) qui sont similaires. Enfin, nous proposons

un SR plus efficace basé sur la recommandation hybride afin d'éliminer certains inconvénients. La Figure 28 montre la méthodologie de notre travail.

En outre, nous avons mis en œuvre quatorze algorithmes, mais les meilleurs d'entre eux sont respectivement K-NN, K-NN avec GridSearchCV, SVM et Decision Tree (DT) Classifier avec des précisions de 96%, 96%, 92% et 89% : 96%, 96%, 92%, et 89%, de sorte que les procédures de chaque algorithme sont écrites :

L'algorithme des K-Nearest Neighbors(K-NN) est une méthode d'apprentissage automatique remarquable par sa simplicité et la facilité de son implémentation. Classé parmi les algorithmes supervisés, il se montre tout aussi efficace pour les tâches de classification que pour les problèmes de régression.

Algorithm: K-NN

Input: the training set D, test objet x, category label set C.
Output: the category c(x) of test object x, c(x) belongs to C.

1. Begin
2. For each y belongs to D calculate the distance D(y,x) between y and x.
   end for
3. Select the subset N from the data set D, the N contains k training samples which are the k narest neighbors of the test sample x.
4. Calculate the category of x:
   $c_k = \arg max_{c \in C} \sum_{y \in N} \mathbf{1}(c = class(y))$
5. End

L’algorithme Support Vector Machine (SVM) est un modèle d'apprentissage supervisé avec des algorithmes d'apprentissage associés qui analysent les données pour la classification et l'analyse de régression.

Algorithm: SVM

Input: Determine the various training and test data
Output: Determine the calculate accuracy
Select the optimal value of cost and gamma for SVM
While (stopping condition is not met) do
1. Implement SVM train step for each data point.
2. Implement SVM classify for testing data point.

End while

Return accuracy

Algorithme de classification par arbre de décision appartient à la famille des algorithmes d'apprentissage automatique supervisé.

Algorithm: Decision Tree

Input: S, where S= set of classified instances
Output: Decision Tree
Require: $S \neq \emptyset$, ***num_attibutes*** **> 0**

```
1.  Procedure Build Tree
2.  Repeat
3.    maxGain ← 0
4.    splitA ← null
5.    e ← Entropy(Attributes)
6.    For all Attributes a in S do
7.        gain ← InformationGain(a,e)
8.        If gain > maxGain then
9.            maxGain ← gain
10.           splitA ← a
11.       End if
12.   End for
13.   Partition(S,splitA)
14.   Until all partitions processed
15. End procedure
```

Nous avons évalué et comparé les performances de plusieurs algorithmes de recommandation. Nous avons mené une étude expérimentale sur Databricks et Jupyter Notebook, un ensemble de données électorales accessibles au public, afin d'analyser et d'évaluer plusieurs algorithmes de

recommandation, à savoir : Fréquence des termes-fréquence inverse des documents[198], Rocchio[199], analyse sémantique latente[200], voisins les plus proches[201], Naïve Bayes[202], arbre de décision[203], SVM[204], régression linéaire (LR)[205], K-means[206], Slope One[207], factorisation non matricielle[208], vecteur de décomposition singulière[188], co-clustering[209], corrélation de Pearson[210] et réseau neuronal artificiel (ANN)[211].L'évaluation des algorithmes est basée sur la précision, le rappel, le f1- score, l'exactitude, l'aire sous la courbe, la RMSE, la MSE et la MAE.

Définitions :

1. Vrai positif (tp) = nombre d'instances correctement prédites comme étant un vote.
2. Faux positifs (fp) = nombre d'instances incorrectement prédites comme étant des votes.
3. Vrai négatif (tn) = nombre d'instances correctement prédites comme n'ayant pas voté.
4. Faux négatifs (fn) = nombre d'instances incorrectement prédites comme n'ayant pas voté.

Précision : c'est la mesure de la qualité. Il s'agit de la mesure de la correspondance entre les observations positives prédites et la somme totale des observations positives prédites. Elle répond à la question suivante : Parmi toutes les instances étiquetées comme des votes, combien de fois cela s'est-il avéré vrai ? Elle peut être calculée comme indiqué dans l'équation 38 :

$$Precision = \frac{tp}{tp + fp} \tag{38}$$

Recall : il s'agit de la mesure de l'exhaustivité. Elle détermine le rapport entre les observations positives prédites de manière adéquate et l'ensemble des observations de la classe définie - oui. Dans le cas contraire, elle répond à la question 38 :

Parmi toutes les vraies prédictions de vote, combien en avons-nous étiquetées ? L'équation 39 montre comment calculer cette valeur :

$$Recall = \frac{tp}{tp + fn} \tag{39}$$

F1 – Score : il s'agit de la moyenne pondérée des valeurs de précision et de rappel. Ce score tient donc compte à la fois des faux positifs et des faux négatifs. Il peut être calculé selon l'équation suivante :

$$F1 - score = 2 * \frac{precision * recall}{tprecisio + recall} \tag{40}$$

Accuracy: il s'agit d'un rapport entre le nombre d'instances prédites de manière appropriée et le nombre total d'instances. La précision d'un algorithme est sa capacité à différencier avec exactitude les cas de vote et de non-vote. La précision d'un système peut être mesurée par l'équation suivante :

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn} \tag{41}$$

RMSE : L'Erreur Quadratique Moyenne (Root Mean Square Error) constitue une métrique statistique essentielle pour évaluer la précision des prévisions comparées aux valeurs réelles. Elle mesure l'écart entre les valeurs estimées et les valeurs observées en calculant la racine carrée de la moyenne des carrés de ces écarts. La formule de la RMSE est la suivante :

$$RMSE = \sqrt{\frac{\sum_{c \in C} \sum_{i \in Itest_C} (reco(c,i) - r_{c,i})^2}{\sum_{c \in C} Itest_C}} \tag{42}$$

MSE : l'erreur quadratique moyenne est la moyenne quadratique entre les valeurs estimées et la valeur réelle, comme indiqué dans l'équation ci-dessous :

$$MSE = \frac{1}{N} \sum_{i=1} (Y - \hat{Y})^2 \tag{43}$$

MAE : l'erreur absolue moyenne est la moyenne, sur l'échantillon de vérification, des valeurs absolues des différences entre la prévision et la direction correspondante, comme indiqué dans l'équation (44) :

$$MAE = \frac{\sum_{c \in C} \sum_{i \in Itest_c} |reco(c,i) - r_{c,i}|}{\sum_{c \in C} Itest_C} \tag{44}$$

AUC : spécifie le degré d'efficacité d'un modèle à différencier les classes données. Plus la valeur de l'AUC est élevée, plus le modèle est capable de prévoir les 0 comme des 0 et les 1 comme des 1, c'est-à-dire que le modèle peut distinguer avec précision les cas de vote et d'absence de vote.

IV.6. Résultats et discussion

Au cours de notre travail, nous avons utilisé : Python, Apache Spark dans Jupyter, Colab, et Cloud Databricks dans un ordinateur Acer Predator avec les performances suivantes : système d'exploitation (window 10 Home), Processeur Intel R core i7, vitesse 2,60 GHz, partie affichage & graphique (Contrôleur graphique fabricant NVIDIA R, Modèle GForce R GTX 1660Ti, Capacité de mémoire jusqu'à 6 Go) et stockage 1 TB, Capacité totale du disque dur 1 TB, 16 Go RAM.

Dans ce chapitre, nous considérons l'approche hybride comme le système de recommandation le plus puissant pour recommander un candidat à un électeur. Nous utilisons les données de ce lien (https://www.kaggle.com/unanimad/us-election-020?select=governors_county_candidate.csv:shows tous les fichiers utilisés au format CSV : comma separated values) comme données de test. Notre système se distingue par sa capacité à prendre en compte l'éventualité qu'un candidat ait délibérément choisi de ne pas participer au vote. Dès lors, il est impératif de déterminer le nombre maximal de votants pour chaque candidat en recourant à des méthodologies analogues. Plus précisément, nous allons résoudre notre problème en différentes étapes dont : importation de nos données, prétraitement de nos données, modélisation, évaluations (comme le rappel, la précision, la mesure f1-score, RMSE, MSE, MAE, AUC), et comparaison de nos résultats d'algorithmes pour faire la recommandation la plus forte.

Les Tableaux 5 et 6 montrent que K-Nearest Neighbors et SGDC avec GridSearchCV ont obtenu de très bons résultats, avec une précision de 96 % sur la validation croisée 7 fois. Le SVM suit avec une précision de 92 %. En outre, l'algorithme de Rocchio a une précision minimale de 66 % par rapport aux autres algorithmes. Les figures 29 à 39 illustrent et comparent les performances des algorithmes utilisés sur la base de la précision, du rappel, du f1- score, de la RMSE, de la MSE, de la MAE et de l'AUC (aire sous la courbe ROC évaluée) pour l'ensemble de données relatives aux élections.

Sur la base des résultats, nous pouvons observer que le K-NN et le SGDC GridSearchCV ont surpassé tous les autres classificateurs. À l'avenir, nous utiliserons ces classificateurs dans notre système de recommandation intelligent, où des recommandations personnalisées seront générées pour chaque utilisateur en fonction de sa condition de vote ou d'abstention.

Algorithm	Precision	Recall	F1-score	Accuracy
Rocchio's	0.77	0.56	0.66	0.66
KNN	0.96	0.96	0.96	0.96
SGDC GridSearchCV	0.96	0.96	0.96	0.96
DTC	0.88	0.89	0.88	0.89
SVM	0.85	0.92	0.89	0.92

Tableau 5 : Comparaison entre les algorithmes selon les mesures : précision, rappel, F1- score et exactitude

Algorithm	MSE	RMSE	MAE	AUC
Rocchio's	22.893	4.784	2.17	0.96
KNN	1871268587.5	43258.16	565.19	1.0
SGDC GridSearchCV	195255897.58	12460.17	429.92	1.0
Naïve Bayes	231533.567	481.17	16.638	0.99
DTC	231514.49	481.15	15.831	0.89
SVM	0.85	0.92	0.89	0.92
LR	1.0	1.0	1.0	1.0
ANN	0.991	0.9995	0.01	0.5

Tableau 6 : Comparaison des différents algorithmes par les métriques : MSE, RMSE, MAE et AUC.

D'après le tableau 7, l'algorithme slope one présente les erreurs les plus faibles en termes de MSE, RMSE et MAE par rapport à tous les autres algorithmes, suivi par l'algorithme SVDpp. On constate que l'algorithme SVD est celui qui admet des valeurs colossales en termes d'erreurs, ce

qui induit que slope one représente le meilleur algorithme qui explique mieux notre variable qui présente d'ailleurs une MSE de 0,59<1, RMSE de 0,768<1, et MAE=0,760<1.

Algorithm	**MSE**	**RMSE**	**MAE**
Slope One	0.59	0.768	0.768
NMF	40110374.3	2002.608221	2002.56
SVD	4010369.24	2002.608225	2002.56
SVDpp	231533.567	481.17	16.638

Tableau 7 : Evaluation et comparaison de la précision des modèles de prédiction

Les courbes da la figure ci dessous sont constituées des valeurs de précision et de rappel pour les valeurs de seuil qui sont comprises entre 0 et 1. Notre objectif est de rendre la courbe aussi proche que (1,1), ce qui signifie une bonne précision et un bon rappel ce qui est dans la première courbe de la Figure 29 dans la classe 10 de surface 87%.

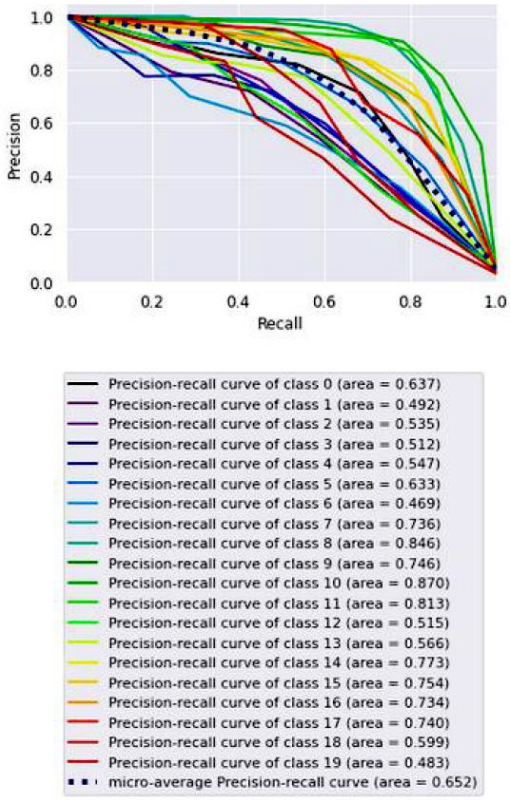

Figure 29 : ROC (prédiction vs test) de la méthode de Rocchio

Dans la deuxième courbe de la Figure 30, lorsque 0,5 < AUC <1, il y a de bonnes chances que le classificateur puisse distinguer les valeurs de classe positives des valeurs de classe négatives. En effet, le classificateur peut détecter un plus grand nombre de vrais positifs et de vrais négatifs que de faux négatifs et de faux positifs. Dans notre cas, nous avons un problème de classification binaire multi-classes utilisant la technique Un contre tous. Pour cela, la meilleure classe de ROC est la classe 8 de la zone 96%.

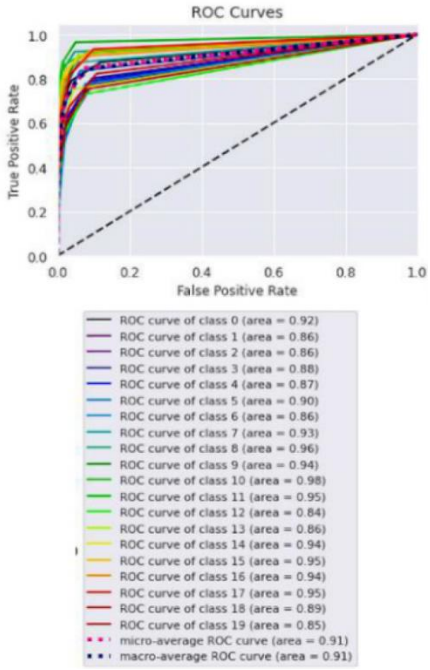

Figure 30 : ROC (probabilité en fonction du test) de la méthode de Rocchio

Dans la Figure 31, si l'AUC = 1, le classificateur peut parfaitement distinguer correctement tous les points de classe positifs et négatifs. Cependant, si l'AUC avait été de 0, le classificateur aurait prédit tous les points négatifs comme positifs et tous les points positifs comme négatifs.

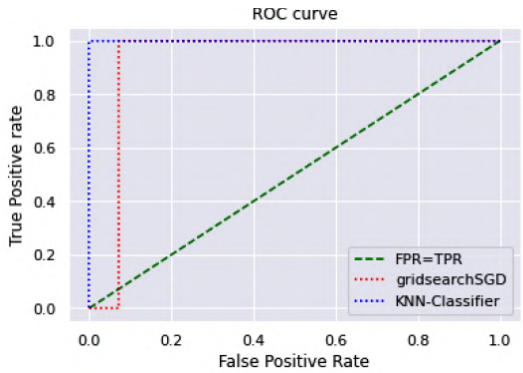

Figure 31 : ROC de SGDC GridSearchCV et KNN

Dans la Figure 32, si l'AUC = 1, le classificateur peut distinguer parfaitement tous les points positifs et négatifs de la classe. En revanche, si l'AUC avait été de 0, le classificateur aurait prédit tous les négatifs comme positifs et tous les positifs comme négatifs.

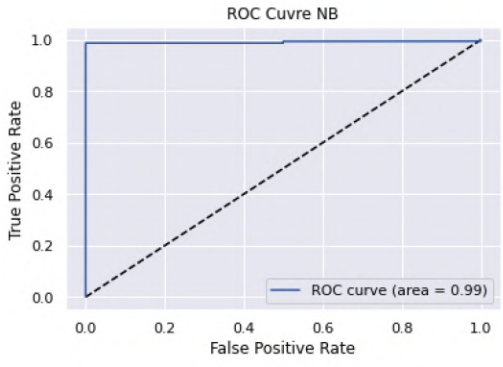

Figure 32 : ROC de NB

Dans la Figure 33, si l'AUC = 1, le classificateur est capable de distinguer parfaitement tous les points de classe positifs et négatifs. Cependant, si l'AUC avait pu être de 0, le classificateur aurait prédit tous les négatifs comme positifs et tous les positifs comme négatifs.

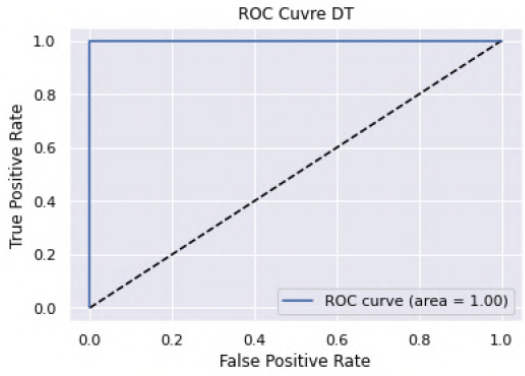

Figure 33 : ROC de DT

La Figure 34 montre que le score de formation est toujours proche du maximum de 1,0 et que le score de validation pourrait être augmenté avec davantage d'échantillons de formation de 0,88.

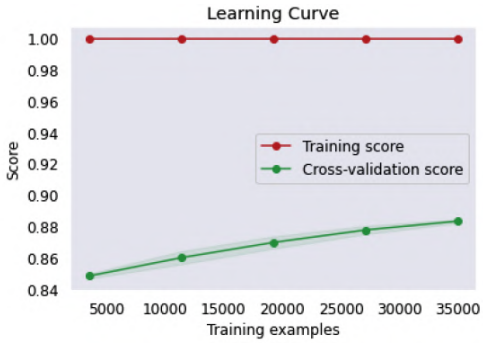

Figure 34 : Courbe d'apprentissage de DT

La Figure 35 montre clairement que le score d'entraînement reste constamment proche de la valeur maximale de 0,92. Il est également évident que le score de validation pourrait être amélioré en augmentant le nombre d'échantillons d'entraînement, ce qui pourrait potentiellement atteindre le score maximum de 0,92.

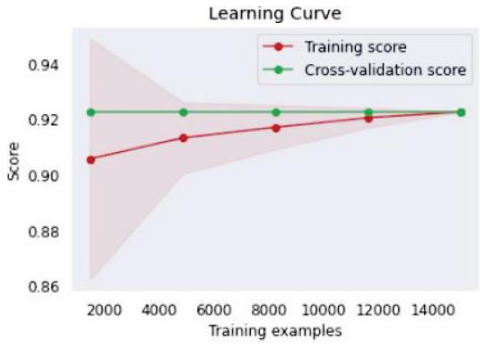

Figure 35 : Courbe d'apprentissage du SVM

La Figure 36 montre clairement que le score d'apprentissage est toujours proche du maximum de 0,0 et que le score de validation pourrait être réduit avec davantage d'échantillons d'apprentissage.

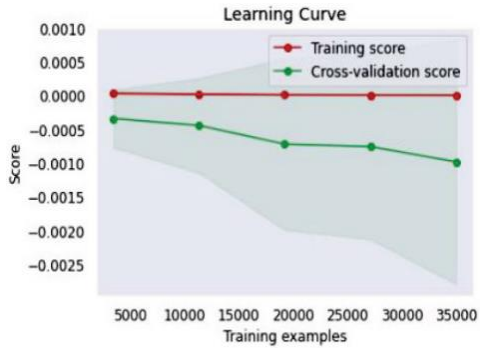

Figure 36 : Courbe d'apprentissage de LR

Dans la Figure 37, nous utilisons cette courbe pour sélectionner le nombre optimal de grappes en ajustant le modèle avec une gamme de valeurs pour K dans l'algorithme K-means. Nous avons représenté une ligne tracée entre la SSE (Somme des Erreurs Quadratiques) et le nombre de clusters pour trouver le point représentant le "point de coude" (c'est un point après lequel la SSE ou l'inertie commence à diminuer linéairement).

Le point du graphique SES / Inertie où le SES ou l'inertie commence à diminuer linéairement est le point de coude. Dans la figure 12, on peut noter que c'est à partir du nombre de grappes = 6 que le SSE commence à diminuer linéairement.

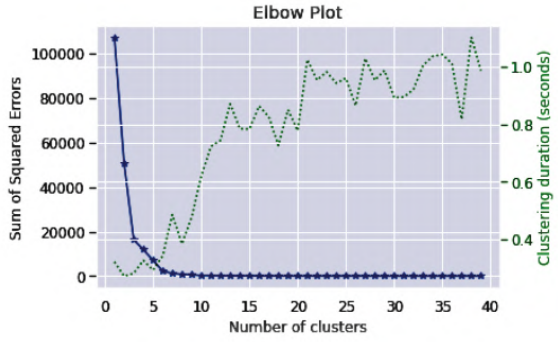

Figure 37 : Somme des carrés des erreurs en fonction du nombre de clusters

Dans la Figure 38, nous avons une SSC = 0,5, ce qui signifie que le classificateur n'est pas en mesure de faire la distinction entre les points de classe positifs et négatifs. Cela signifie que le classificateur prédit une classe aléatoire ou une classe constante pour tous les points de données.

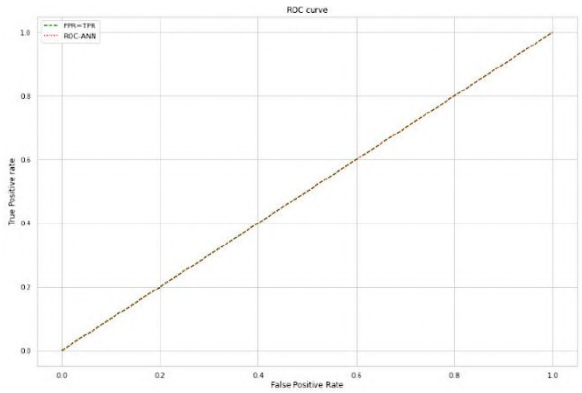

Figure 38 : 13. ROC de l'ANN

La Figure 39 montre que le score d'entraînement qui est toujours proche du score maximum de 1,0 et que le score de validation pourrait être augmenté avec davantage d'échantillons d'entraînement du score maximum de 1,0.

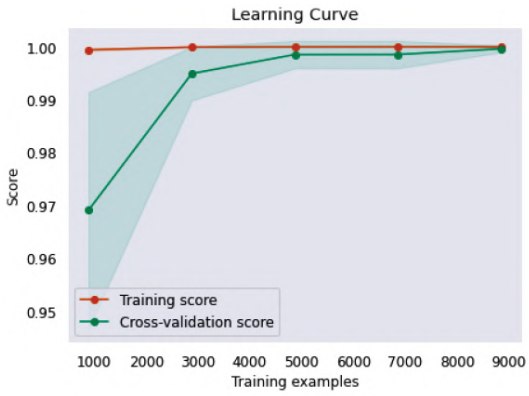

Figure 39 : Courbe d'apprentissage de Pearson

Dans la Figure 40, nous obtenons l'indice des cinq premiers produits en corrélation avec l'identifiant du produit choisi par l'utilisateur, puis nous voyons les notes que l'utilisateur peut attribuer à ces cinq produits les plus corrélés.

	index	productId	estimated_rating
0	0	CST	1
1	1	IND	1
2	2	MNP	1
3	3	REP	1
4	4	UNA	1

	index	productId	estimated_rating
0	0	CST	1
1	1	DEM	1
2	2	GRN	1
3	3	IAP	1
4	4	IND	1

Figure 40 : Résultats de l'approche hybride

IV.7. Conclusion

La croissance des appareils IoT générant une grande quantité de données a fait du cloud computing une technologie indispensable pour faciliter le stockage, le traitement, l'analyse et l'accès facile aux données. L'utilisation étant diversifiée, elle doit toujours se faire de manière optimale. Plusieurs méthodes, telles que les SR, peuvent être utilisées pour améliorer son utilisation. Les SR sont connus pour fournir des suggestions pertinentes à un utilisateur en fonction de ses préférences ou de ses besoins. Par conséquent, les SR actuels s'appuient sur l'internet des objets (IoT) pour atteindre certains objectifs. En outre, il y a trop d'informations disponibles sur l'internet qui doivent

être analysées ou calculées à des fins de recommandation. Cependant, avec l'augmentation du volume de données, les performances du système se dégradent progressivement. Par conséquent, un environnement distribué tel que le cloud est nécessaire pour effectuer et stocker tous les calculs sur un seul système. Sur la base de cette étude, nous observons que tous les domaines susmentionnés sont interdépendants.

Dans ce chapitre, nous avons réalisé une étude comparative des performances de quatorze modèles appliqués à l'ensemble de données électorales, parmi d'autres : Rocchio, analyse sémantique latente, K-voisins les plus proches, Naïve Bayes, classificateur d'arbre décisionnel, machine à vecteur de support, régression linéaire, K-moyens, Slope One, factorisation non matricielle, vecteur de décomposition singulière, co-regroupement, corrélation de Pearson et réseau de neurones artificiels.

L'évaluation des algorithmes est basée sur la précision, le rappel, le f1-score, l'exactitude, le RMSE, le MSE, le MAE et l'aire sous la courbe. Les résultats montrent que K-NN et SGDC GridSearchCV sont les plus performants pour tous les paramètres d'évaluation.

CHAPITRE V : GREY WOLF OPTIMISATION POUR LE PLACEMENT DESMACHINES VIRTUELLES DANS LE CLOUD

V.1. Résumé

Dans un environnement de cloud computing, une bonne gestion des ressources reste un défi majeur pour son bon fonctionnement. La mise en œuvre du VMP sur des machines physiques permet d'atteindre divers objectifs, tels que l'allocation des ressources, l'équilibrage de la charge, la consommation d'énergie et la qualité de service. Le VMP dans le cloud est critique, il est donc important d'auditer sa mise en œuvre. Il doit prendre en compte les ressources du serveur physique, notamment le processeur, la mémoire vive et le stockage. Dans ce chapitre, un algorithme métaheuristique fondé sur la méthode d'optimisation des loups gris (GWO) est employé afin de parfaire le placement des machines virtuelles (VMP) au sein d'un environnement infonuagique. Cette approche vise à réduire de manière efficiente le nombre de machines virtuelles actives nécessaires à l'hébergement des serveurs virtuels. Les résultats expérimentaux démontrent l'efficacité de la méthode proposée, appelée Grey Wolf Optimization for Virtual Machine Placement (GWOVMP). La méthode réduit la consommation d'énergie de 20,99 et le gaspillage de ressources de 1,80 par rapport aux algorithmes existants.

Les mots clés : Cloud Computing, GWO Algorithm, Metaheuristics Algorithm, Optimization, Virtual Machine Placement, Data Center, Power Consumption.

V.2. Introduction

De nos jours, le cloud computing s'est développée rapidement et est devenue une technologie indispensable dans le domaine de l'IT. Cela a conduit à l'émergence de centres de données modernes de plus en plus grands avec un grand nombre d'appareils physiques, ce qui entraîne une montée remarquable de la consommation d'énergie et d'énormes dépenses de refroidissement. Dans ces centres de données, trois types d'équipements physiques consomment de l'électricité : les serveurs, les systèmes de refroidissement et les équipements de réseau. Pour avoir une meilleure idée des avantages qu'un utilisateur peut obtenir, CC offre quatre modes de déploiement principaux : cloud public, cloud privé, cloud hybride et cloud communautaire. Les différentes ressources du CC sont fournies sous la forme de trois services principaux : Infrastructure as a Service (IaaS), Platform as a Service (PaaS) et Software as a Service (SaaS).

Dans le modèle du cloud public, les différentes ressources sont gérées par le fournisseur de services qui possède le cloud, et ses ressources sont vendues au public en fonction de ses propres besoins. Dans ce type de cloud, une partie des ressources peut être louée et gérée par les utilisateurs finaux. Parmi les fournisseurs du cloud les plus connus, on peut citer Amazon, Google et Microsoft[212]. En ce qui concerne le modèle de cloud privé, le propriétaire de ce type du cloud serait une entité ou une organisation qui souhaite utiliser certaines applications contenant des informations très sensibles. S'il s'agit d'un cloud communautaire, le fournisseur est un ensemble d'organisations qui se réunissent dans un but commun[213]. Dans ce cas, plusieurs entreprises peuvent par exemple décider de fédérer leurs efforts en construisant et en gérant ensemble un cloud. Grâce à la virtualisation, ces serveurs sont proposés aux clients sous forme de VM, ce qui permet une grande flexibilité pour la bonne gestion des ressources. Ces centres de données facilitent surtout l'exécution des charges de travail de manière élastique ; certaines techniques telles que la consolidation du matériel sont donc combinées pour maximiser l'efficacité énergétique. Dans un DC (data center), la technique de virtualisation permet également d'exécuter différentes applications sur des VM ou des conteneurs. Cette technologie permet également une mise en commun efficace des ressources physiques, telles que l'unité centrale, la mémoire vive et les interfaces E/S, tout en permettant aux serveurs physiques d'héberger plusieurs machines virtuelles.

Bien que le CC présente plusieurs avantages, les coûts associés à la consommation d'énergie dans les centres de données ainsi que les émissions de dioxyde de carbone constituent les plus grands défis du CC. Pour surmonter ces difficultés, les avantages de la virtualisation sont utilisés pour minimiser le nombre de machines physiques actives et pour éteindre les machines physiques inactives. Cela nécessite la mise en œuvre d'une stratégie pour mieux gérer le placement des machines virtuelles (VMP). Pour minimiser la puissance totale d'un DC, il est donc nécessaire de restreindre la consommation des PMs. En outre, il existe d'autres défis tels que la sécurité, sur laquelle certaines études se sont concentrées[214]. Les chercheurs se sont intéressés au fait qu'ils peuvent exploiter les grandes capacités de stockage et de traitement qu'offre le cloud computing[215]. Le VMP dans l'environnement cloud reste un sujet qui attire l'attention de plusieurs chercheurs. L'optimisation du VMP dans les centres de données peut améliorer l'efficacité de la quantité d'énergie et l'utilisation optimale des ressources tout en maximisant la qualité des services (QoS) offerts.

L'optimisation des problèmes VMP dans un environnement cloud est considérée comme un problème NP-hard pour lequel il est très rare de converger vers une solution optimale. La résolution d'un tel problème peut être coûteuse en termes de temps de calcul lorsque des méthodes courantes

telles que la théorie des graphes sont utilisées. Il est donc préférable d'utiliser des techniques métaheuristiques et heuristiques. Les métaheuristiques souffrent d'un temps d'exécution élevé mais fournissent des solutions optimales. Les algorithmes heuristiques ont un temps d'exécution modéré, mais ne fournissent pas de solutions de bonne qualité[216]. Plusieurs algorithmes et objectifs ont été mis en œuvre pour résoudre le problème VMP, notamment l'algorithme de logique floue[217].

Dans ce chapitre, un algorithme appelé GWOVMP est adapté de la méthode GWO pour optimiser le VMP dans un DC hétérogène. Pour comparer GWOVMP aux algorithmes existants, nous effectuons des similitudes à l'aide de l'outil CloudSIM. Les principales contributions de ce chapitre sont les suivantes :

- Formulation de modèles mathématiques du problème de placement de VM dans le cloud ;
- Un algorithme métaheuristique basé sur la hiérarchie du loup gris appelé GWOVMP est proposé pour résoudre ce problème VMP dans un environnement cloud ;
- Nous effectuons une simulation à l'aide de CloudSIM et évaluons l'algorithme proposé par rapport à ceux de l'état de l'art.

Le reste de ce chapitre est organisé comme suit. La section 2 présente une revue de la littérature et les différents travaux connexes. La section 3 présente la formulation du problème et le GWO pour le placement de machines virtuelles. L'évaluation expérimentale de l'algorithme GWOVMP et l'analyse des résultats de notre travail se trouvent dans la section 4. Enfin, la section 5 présente la conclusion et les travaux futurs.

V.3. Revue de la littérature

V.3.1. Placement de machines virtuelles

D'une manière générale, le problème du placement est un problème ancien et classique. Il consiste à remplir des boîtes avec différents objets tout en utilisant le nombre minimal de boîtes. Dans le cas des DC, le placement d'une machine virtuelle dans l'affectation optimale des VM dans le CC consiste à affecter les VM dans un nombre minimum de PM d'un DC. Les VM sont donc considérées comme des objets et les PM comme des boîtes.

L'objectif du placement optimal dans le DC est de répartir les charges sur tous les serveurs physiques tout en minimisant des facteurs tels que la consommation d'énergie, l'augmentation de la

qualité de service, le respect de l'accord de niveau de service (SLA) et la réduction du gaspillage des ressources. Dans un centre de données cloud, la virtualisation est l'une des technologies clés sur lesquelles il faut s'appuyer. Les avantages de la virtualisation, tels que la consolidation, la migration et l'équilibrage de la charge des différentes VM entre les PM, augmentent l'efficacité du DC et réduisent les coûts d'exploitation. La Figure 41 illustre l'architecture simplifiée de cette technique de virtualisation.

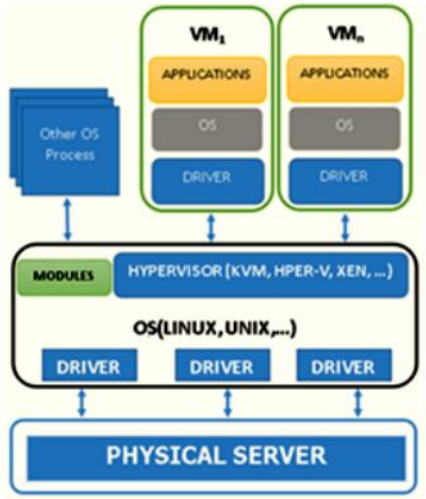

Figure 41 : Architecture de virtualisation

V.3.2. Optimisation par le Grey Worf Optimazation

Au cours des deux dernières décennies, l'optimisation métaheuristique est devenue très populaire. Certains algorithmes métaheuristiques sont bien connus et utilisés dans différents domaines d'application[218][219]. Des algorithmes tels que l'optimisation par colonies de fourmis (ACO), l'algorithme génétique (GA) et l'optimisation par essaims de particules (PSO) ont été utilisés dans diverses recherches, y compris pour le problème VMP. Le Grey Wolf Optimizer est l'un des algorithmes métaheuristiques basés sur la vie qui rappelle les techniques de chasse du loup gris et s'inspire de la chaîne de commandement du loup gris. Ces loups gris vivent en meutes de 5 à 12 individus en moyenne et il existe quatre types de loups gris : α, β, δ et ω. Le premier niveau est composé des alphas qui sont des loups gris mâles ou femelles qui sont plus forts et qui guident les autres dans les domaines appropriés.

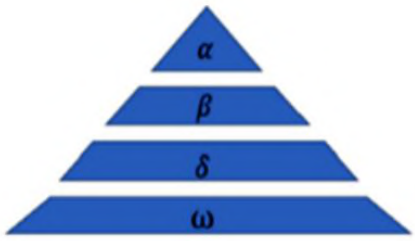

Figure 42 : Hiérarchie sociale des loups gris[219]

La Figure 42[219] illustre la hiérarchie sociale des loups gris. Les équations (45) et (46) ci-dessous présentent la modélisation mathématique de cet algorithme qui est basé sur l'encerclement de la proie par ces loups gris[220]:

$$\vec{D} = \vec{C}.\overrightarrow{X_p}(t) \quad (45)$$

$$\vec{X}(t+1) = \overrightarrow{X_p}(t) - \vec{A}.\vec{D} \quad (46)$$

Dans les équations (44) et (48), l'itération en cours est représentée par t, les coefficients sont représentés dans les équations (47) et (48) par $\vec{A}$ et $\vec{C}$ et $\vec{X}(t)$ est le vecteur qui indique la position du loup gris tandis que le vecteur $\overrightarrow{X_p}$ indique la position de la proie. Les équations suivantes permettent de calculer les vecteurs $\vec{A}$, $\vec{C}$ et $\vec{a}$[221]:

$$\vec{A} = 2\vec{a}.\overrightarrow{r_1} - \vec{a} \quad (47)$$

$$\vec{C} = 2.\overrightarrow{r_2} \quad (48)$$

$$\vec{a} = 2 - t\frac{2}{NumOfIt} \quad (49)$$

Les vecteurs $\overrightarrow{r_1}$ et $\overrightarrow{r_2}$ sont aléatoires dans l'intervalle [0, 1], et pendant la phase d'exploration et d'exploitation, le vecteur $\vec{a}$ diminue linéairement de 2 à 0 au cours de ces itérations. La meilleure solution est représentée d'abord par les alphas, puis les bêtas et enfin les deltas.

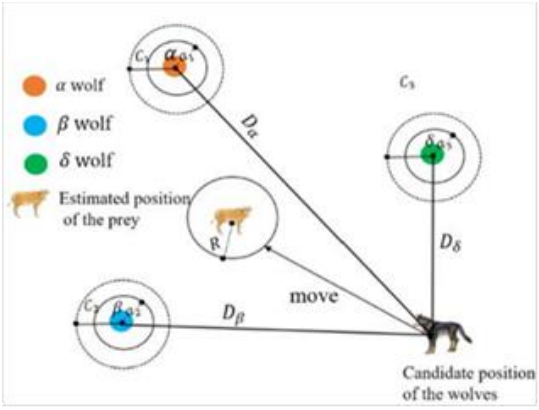

Figure 43 : Mise à jour de la position des loups gris pendant la chasse[219]

Les omégas sont considérés comme des solutions indésirables et ne sont donc pas pris en compte. La Figure 43[219] montre les différentes positions des loups pendant la chasse et peut être définie mathématiquement par les équations ci-dessou[222] :

$$\vec{X}(t+1) = \frac{\overrightarrow{X_1} + \overrightarrow{X_2} + \overrightarrow{X_3}}{3} \tag{50}$$

Où $\vec{X}_1 + \vec{X}_2 + \vec{X}_3$ sont définis dans les équations (51)-(53) :

$$\overrightarrow{X_1} = \overrightarrow{X_\alpha} - \overrightarrow{A_\alpha}.\overrightarrow{D_1} \tag{51}$$

$$\overrightarrow{X_2} = \overrightarrow{X_\beta} - \overrightarrow{A_\beta}.\overrightarrow{D_2} \tag{52}$$

$$\overrightarrow{X_3} = \overrightarrow{X_\delta} - \overrightarrow{A_\delta}.\overrightarrow{D_3} \tag{53}$$

Dans ce cas, $\vec{X}_1 + \vec{X}_2 + \vec{X}_3$ sont les meilleures solutions. $\overrightarrow{D_\alpha}$, $\overrightarrow{D_\beta}$ et $\overrightarrow{D_\delta}$ qui sont les voisinages de la position des loups α, β et δ sont définies dans les équations (54) - (56) comme suit[223]:

$$\overrightarrow{D_\alpha} = |\overrightarrow{C_1}.\overrightarrow{X_\alpha} - \vec{X}| \tag{54}$$

$$\overrightarrow{D_\beta} = |\overrightarrow{C_2}.\overrightarrow{X_\beta} - \vec{X}| \tag{55}$$

$$\overrightarrow{D_\delta} = |\overrightarrow{C_3}.\overrightarrow{X_\delta} - \vec{X}| \tag{56}$$

V.3.3. Travaux connexes

Dans[224], un algorithme a été proposé pour exécuter de manière aléatoire le VMP dans un environnement hétérogène et multidimensionnel de type "cloud". Pour optimiser l'utilisation de l'énergie et des ressources, ils ont utilisé GRVMP (Greedy Randomized Virtual Machine Placement) qui s'inspire du modèle de puissance à deux choix et place les machines virtuelles sur les PM les plus efficaces sur le plan énergétique. Ils ont évalué les performances de leurs algorithmes en utilisant des scénarios de production synthétiques et réels avec des matrices de performance. Leurs résultats montrent une réduction significative des machines virtuelles actives ainsi que du gaspillage total des ressources de leur algorithme par rapport aux méthodes de l'étude précédente.

Les auteurs ont mis en œuvre des techniques intelligentes pour traiter le problème d'optimisation du VMP[225]. Leur objectif est de pouvoir diminuer le nombre de serveurs physiques actifs ainsi que la consommation d'énergie. Ils ont adapté deux algorithmes du modèle d'optimisation Grey Wolf à un problème VMP. Les algorithmes proposés ont été évalués dans un environnement CloudSIM composé de serveurs homogènes et hétérogènes. Par rapport aux techniques existantes, leurs algorithmes minimisent le nombre de PM actifs utilisés pour héberger des VM.

Pour le placement optimal dans un centre de données cloud(CDC), les auteurs ont proposé dans[226] un algorithme basé sur la baleine discrète hybride et l'algorithme d'optimisation multi-objectif. L'optimiseur multivers avec des fonctions chaotiques est également utilisé pour optimiser le placement dans un environnement en cloud. L'objectif principal est de minimiser la consommation d'énergie dans les CDC en réduisant le nombre de machines physiques actives, ainsi que de réduire le gaspillage des ressources en utilisant le placement optimal des VMs sur les PMs. La méthode appliquée évite l'augmentation du taux de migration des machines virtuelles sur les machines physiques. Les résultats obtenus montrent la performance de l'algorithme proposé par rapport à d'autres algorithmes de pointe.

Dans[227], les auteurs ont fourni une technique basée sur un algorithme d'essaim particulaire pour améliorer l'ordonnancement des tâches dans un système cloud. Dans la technique proposée, le choix d'une fonction objective appropriée leur a permis d'effectuer un VMP dynamique sur le PM, d'équilibrer la charge de travail des VM, de réduire le temps de toutes les tâches tout en maximisant l'utilisation de toutes les ressources, et d'augmenter la productivité. La solution proposée fournit des résultats optimaux pour l'ordonnancement des tâches, l'allocation correcte des tâches dans le placement des VM sur le PM, et sur le processus d'externalisation des VM, le temps a été amélioré de 0,02.

Dans[228], les auteurs ont développé deux algorithmes pour la gestion optimale des ressources dans le cloud. Le premier algorithme utilise la planification non supervisée avec l'algorithme K-means et le second est l'algorithme KNN pour l'apprentissage supervisé. Cela s'applique également à l'analyse systématique des modèles VM et PM, qui leur a permis d'établir une règle pour le déploiement hybride et dynamique des ressources du centre de données en cloud.

Les auteurs de[229] ont proposé une technique pour résoudre le problème lié à l'optimisation de la VMP dans le cloud en tenant compte de la puissance et du trafic. La méthode proposée minimise la consommation d'énergie du CDC, le gaspillage des ressources et la consommation du réseau. La méthode a été comparée à des algorithmes de pointe. Les résultats obtenus montrent une réduction de 29 % de la consommation du réseau, une réduction de 18 % de la consommation d'énergie globale et une réduction de 68 % du gaspillage des ressources.

L'approche proposée minimise conjointement la consommation d'énergie, l'utilisation optimale du réseau et le gaspillage des ressources dans un système cloud hétérogène et multidimensionnel. Les résultats ont été obtenus après comparaison avec l'état de l'art de la technologie de l'arbre à graisse.

Dans[230], un algorithme GWO a été proposé sur la base de la capacité de fiabilité des ressources pour l'équilibrage adaptatif de la charge. La méthode utilisée consiste d'abord à trouver les nœuds inactifs ou occupés, puis à calculer le seuil et la fonction d'adéquation pour chaque nœud. Ils ont utilisé CloudSim pour la simulation et leurs résultats ont amélioré le coût et le temps de réponse de la méthode proposée par rapport à d'autres techniques.

Les auteurs de[231] ont proposé un algorithme génétique amélioré (l-GA) basé sur le modèle d'architecture hiérarchique virtuelle (VHAM) pour résoudre le problème du placement des machines virtuelles. Ils ont utilisé l'application d'analyse par éléments finis (FEA) pour mener les expériences. Leurs résultats montrent une réduction efficace du nombre de PMs utilisés pour héberger les VMs et donc une réduction de la consommation d'énergie dans les centres de données tout en assurant la disponibilité du placement. I-GA est plus performant que les autres algorithmes, même dans les petites instances.

Dans[232], les auteurs ont formalisé le problème VMP en une optimisation intelligente tout en s'appuyant sur les différentes spécifications d'un centre de données. L'objectif de leur travail est d'effectuer un bon placement des VM sur les PM sans affecter les performances. Avec les expériences qu'ils ont menées, la méthode utilisée est un algorithme métaheuristique d'optimisation par essaims de particules qui a donné des résultats sur une diminution significative de l'énergie du centre de données et un équilibrage optimal entre les différentes ressources.

Le tableau 8 présente une étude comparative de certains des travaux réalisés sur le VMP. Il indique les différents paramètres utilisés, l'environnement de simulation (SE) et les différents noms d'algorithmes mis en œuvre.

	Paramètres									
Travaux	**Utilisation des ressources**				**Gaspillage deressources**	**SLA**	**Energie**	**Autres**	**Algorithmes**	**SE**
	CPU	**RAM**	**Stockage**	**Bande passante**						
[224]	×	×			×		×		GRVMP	Amazon EC2
[225]	×	×					×		GWO-VMP	CloudSIM
[226]	×	×	×	×	×	×	×			Amazon EC2
[227]	×	×			×		×			MATLAB
[228]	×				×		×		EHML	CloudSIM Amazon Cloud datacenter
[229]	×				×		×	×		Java
[230]					×		×	×		CloudSIM
[231]	×	×			×		×		I-GA	CloudSIM
[232]	×	×					×		RPMOTA	
Notre travail	×	×	×		×		×		GWOVMP	CloudSIM

Tableau 8 : Étude comparative de l'état de l'art

V.4. GWO adapté(GWOVMP)

V.4.1. Formulation du problème

Dans ce chapitre, l'objectif de base est de mettre en œuvre un schéma de placement optimal pour les machines virtuelles. Le but principal du VMP est de pouvoir modifier l'allocation originale des machines virtuelles aux nœuds physiques afin de minimiser les coûts énergétiques. Dans ce travail, l'objectif est de minimiser cette consommation d'énergie dans le cloud computing en fonction des ressources disponibles. Notons la fonction d'appariement f. Ce problème de placement de machines virtuelles peut être noté f : V→P[216]. Le nombre minimum de serveurs actifs pour une solution optimale de VMP peut être formulé comme suit :

$$f(s) = min \sum_{i=1}^{n} y_i \tag{57}$$

Dans le cloud computing, les centres de données sont composés de machines physiques hétérogènes. Dans ce chapitre, un certain nombre m de PM et un certain nombre n de VM sont considérés. P désigne l'ensemble des PM où i est le $i^{ième}$ PM, i appartient à $[1, \dots, m]$, et $P_i \in P$. De même, V désigne l'ensemble des VM hétérogènes, V_j où désigne la $j^{ième}$ VM, j appartient à $[1, \dots, n$], et $V_j \in V$.

La capacité de la $j^{ième}$ VM est représentée par un vecteur à r dimension $V_j = \{R_i^1, R_i^2, \dots, R_i^r\}$ où R_i^k représente la capacité de la ressource k^{th} de la $j^{ième}$ VM, $\forall k = \{1,2, \dots, r\}$.

Pour exprimer la relation entre VM et PM, cette fonction est représentée par la matrice binaire $X_{m \times n}$ comme indiqué dans l'équation (58):

$$x_{ij} = \begin{cases} 1 & if\ V_j\ is\ assigned\ to\ P_i \\ 0 & Otherwise, \forall i, \in P\ et\ \forall j \in V \end{cases} \tag{58}$$

Pour désigner l'état de la PM, $y_{m \times n}$les matrices y_{ij} et P_i sont utilisées et $y_{ij=1}$ est active et si $y_{ij} = 0$ inactive. C'est ce que montre l'équation (59) :

$$y_{ij} = \begin{cases} 1 & if\ \sum_{i=1}^{n} P_i \geq 1\ \ \forall i \in P_i \\ 0 & otherwise \end{cases} \tag{59}$$

Dans ce travail, les ressources de processeur, de mémoire et de stockage sont prises en compte. Les capacités de la puissance de calcul de la PM sont représentées par P^{cpu} et celles de la mémoire

par P^{ram}. De même, le processeur et la mémoire de la VM sont représentés par $\mathrm{V}^{\mathrm{cpu}}$ et $\mathrm{V}^{\mathrm{ram}}$ pour les ressources virtuelles, ce qui nous amène aux contraintes suivantes :

$$\sum_{i}^{M} V^{cpu}.x_{ij} \leq P^{cpu}.y_i \quad \forall i \in P \tag{60}$$

$$\sum_{i}^{N} V^{ram}.x_{ij} \leq P^{ram}.y_i \quad \forall i \in P \tag{61}$$

Dans les équations (60) et (61), nous spécifions que le CPU et la mémoire ne doivent pas dépasser, ce qui spécifie les contraintes de capacité. L'équation (62) stipule que l'espace de stockage total requis pour les VM sur les PM ne doit pas dépasser l'espace de stockage du PM qu'il héberge.

$$\sum_{i}^{N} V^{sto}.x_{ij} \leq P^{sto}.y_i \quad \forall i \in P \tag{62}$$

Plusieurs travaux montrent que la consommation d'énergie des CC est linéaire avec une charge linéaire[233], [234], [235]. L'arrêt des serveurs inactifs réduira le coût de la consommation d'énergie.

$$P^{cpu} = \begin{cases} P_{iddle} + \left(P_{full} - P_{iddle}\right) \times U_i^{cpu} & if\ U_i^{cpu} > 0 \\ 0, & otherwise \end{cases} \tag{63}$$

Où P_{iddle} est la puissance de la consommation d'énergie par le PM lorsqu'il est en état d'inactivité, P_i est la puissance qui est consommée dans un CPU pleinement utilisé, et ∈ [0, 1]. La formule ci-dessous montre que U_i^{cpu}est le taux de processeur (CPU) du PM Pi en fonction de la demande de VM V_i.

$$U_i^{cpu} = \frac{\sum_{j=1}^{n} x_{ij} \times R_i^{cpu}}{P_i^{cpu}} \tag{64}$$

Pour calculer le gaspillage des différentes ressources d'un PM de dimension r, nous généralisons la formule ci-dessous de manière à l'appliquer aux ressources :

$$W_i = \frac{|R_i^{cpu} - R_i^{ram}|}{|P_i^{cpu} + P_i^{ram}|} + \varepsilon \tag{65}$$

Où Wi représente le gaspillage des différentes ressources PM. R^{cpu} et R^{ram} représentent respectivement les ressources processeur et mémoire normalisées de la machine physique ; les valeurs de et décrivent l'utilisation normalisée de la machine. ϵ est un petit nombre réel égal à 0,0001[236]. Ce modèle vise à inclure au mieux les ressources de tous les PM et permet d'atteindre un équilibre entre plusieurs ressources. La formule mathématique ci-dessous indique le gaspillage total de ressources W tot de tous les MP dans un CDC :

$$W_i^{tot} = \sum_{i=1}^{m} y_i \times \frac{\sum_{k=1}^{r} |R_i^{cpu} - R_i^{ram}|}{\sum_{k=1}^{r} P_k^r} + \varepsilon \quad (66)$$

V.4.2. GWO pour le placement de machines virtuelles

La réduction effective de la consommation d'énergie dans un centre de données cloud est généralement obtenue en réduisant considérablement le nombre de machines physiques actives. Dans ce cas, un algorithme adapté de l'algorithme GWO est utilisé pour résoudre le problème VMP afin d'obtenir une solution optimale qui alloue les machines virtuelles à un nombre minimum de serveurs physiques actifs. La meilleure solution qui a un nombre minimum de serveurs physiques actifs est choisie comme S. Dans l'état initial, le placement optimal des VM dans les serveurs physiques n'est pas connu et la meilleure solution qui a été générée, désignée comme SI, est considérée en premier. En effet, pour faire correspondre les m VM aux n PM, une distribution aléatoire des m VM aux n PM et chaque VM est soumise à un seul PM. Il y a donc m^n solutions possibles.

Pour rechercher une proie, les loups mettent continuellement à jour leurs positions. Après l'étape de solution qui est distribuée aléatoirement pour chaque loup qui rejoint la meute. GWOVMP génère une nouvelle solution tout en mettant à jour la solution existante de chaque loup pour trouver la distribution la plus optimale des VM sur le PM. Pour chaque itération t, les positions des loups sont mises à jour sur la base des meilleures solutions générées par α, β et δ.

Pour minimiser le nombre de machines physiques actives et la consommation d'énergie dans un système basé sur le cloud, la solution doit être améliorée pour réduire le nombre de PMs actives en cours d'exécution dans toutes les conditions. En général, l'algorithme GWO met à jour le chemin du meilleur ensemble de solutions sur la base de l'équation (49), où un par un certain emplacement de VM sont mis à jour. On peut constater que certains emplacements sont mis à jour en dehors de la limite [1, ..., n]. Dans ce cas, l'algorithme s'appuie sur le modèle binaire pour

recalculer l'allocation de ces machines virtuelles tout en veillant à ce que les serveurs physiques soient dans la limite :

$$W_{ij}(t+1) = \begin{cases} 1 & if\ sigmoid\ (\frac{\overrightarrow{x_1} + \overrightarrow{x_2} + \overrightarrow{x_3}}{3}) \\ 0 & otherwise \end{cases} \quad (67)$$

En général, sans tenir compte du cas traité dans ce document, nous avons dans l'équation (66) le rand qui représente un nombre aléatoire extrait de la distribution uniforme qui ∈ [0, 1]. xij représente la position binaire qui est mise à jour à l'itération t et dans la dimension déterminée, la fonction sigmoïde est déterminée par l'équation (67) :

$$sigmoid\ (a) = \frac{1}{1 + e^{-10(a-0,5)}} \quad (68)$$

Input: VM, PM
Output: Placement of VMs on PMs

Step 1: Initialization of a population of n grey wolves positions randomly. Set parameters a via Equation (47) and (48). n is defined as the number of wolves considered as a search factor. Determine the total number of iterations. Set it= 1 as the initial iteration.
Step 2: Let n Wolves build and save the top tree solutions δ.
Step 3: Update the solutions of α, β and δ with Equations (59) and (60).
Step 4: Calculate the different fitness values for all solutions and identify the tree best solution for the current iteration
Step 5: Terminate the algorithm. Check the current number of iterations; the algorithm terminates if the number reached exceeds the maximum number of iterations. Otherwise, increment and return to step (3).
Step 6: Return α

Cet algorithme décrit le GWOVMP sur la base d'un ajustement des différentes opérations de base de l'algorithme d'optimisation de Grey Wolf, et l'organigramme correspondant est présenté à la Figure 44.

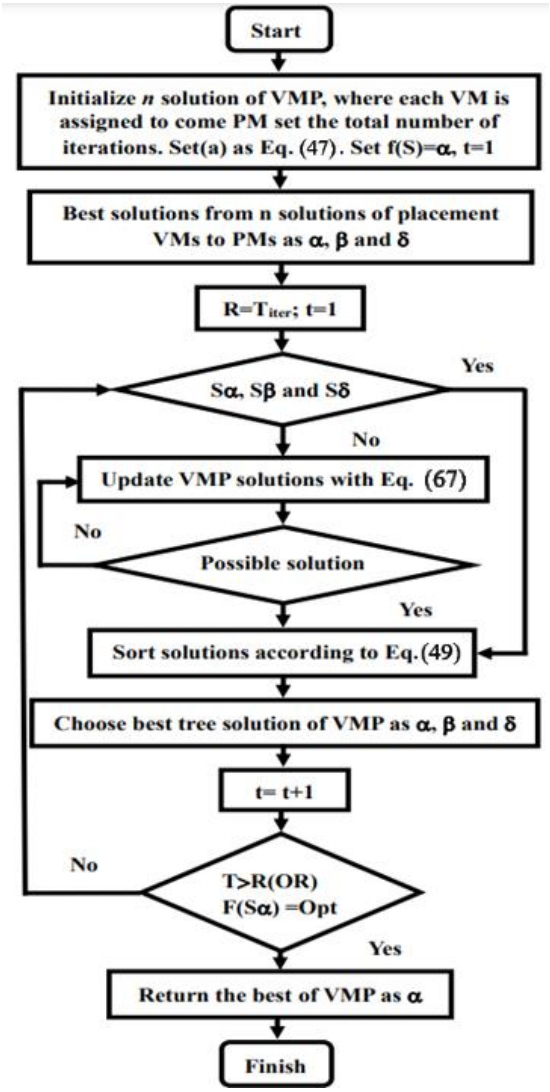

Figure 44 : Flowchart du GWOVMP

Si une VM n'a pas été allouée lors d'une mise à jour, d'une suppression ou d'une surcharge de doublons, une opération est utilisée pour la réallouer. Les contraintes des VM sont évaluées par les équations (60), (61) et (62) après une mise à jour des positions des loups. Si la valeur de xij correspondant à Vj n'a été attribuée à aucun PM lors de la mise à jour, la VM est réattribuée à un PM non surchargé disposant de ressources suffisantes pour répondre à la demande.

Comme nous l'avons mentionné, le GWO est basé sur une stratégie maîtresse dans laquelle les loups mettent à jour leurs positions à chaque fois pour attaquer la proie en se basant sur les positions des trois premiers loups. En outre, la consommation d'énergie du CDC doit être considérablement réduite en limitant le nombre de serveurs physiques actifs. Cet objectif peut être atteint en améliorant la solution sur les serveurs actifs tout en réduisant leur nombre.

Dans ce cas, on constate que la phase d'exploration effectuée par les loups peut parfois se faire en sens inverse en faisant intervenir l'espace de recherche binaire. C'est sur les généralités de cet algorithme que si chaque bit des valeurs α, β et δ passe à un, la valeur qui correspond à xij à soumettre à Pi est mise à jour.

Pour équilibrer et maximiser l'utilisation des ressources mémoire et CPU, les VM sont réaffectées aux PM qui donnent la plus petite dissimilarité absolue estimée entre l'utilisation des ressources CPU et mémoire après l'utilisation de la machine virtuelle. La différence entre l'utilisation du CPU et de la mémoire après l'ajout d'une machine virtuelle au PM est calculée après avoir ajouté les différents besoins en ressources de la machine virtuelle aux ressources utilisées par le serveur physique.

Le tableau 9 présente les différents symboles utilisés dans ce chapitre. Chaque symbole a une signification particulière et a été soigneusement choisi pour contribuer à la clarté et à la compréhension des différentes équations. Le lecteur dispose ainsi d'une référence visuelle pratique pour interpréter rapidement les informations contenues dans le travail.

Symbol	Descriptions
P	Set of PMs
V	Set of VMs
P^{cpu}	CPU capacities of the PM
P^{ram}	CPU capacities of the PM
V^{cpu}	CPU capacities of the VM
V^{ram}	CPU capacities of the VM
P^{sto}	Storage capacities of the PM
V^{sto}	Storage capacities of the VM
P_{iddle}	Power that is consumed in an idle state
P_{full}	Power that is consumed in an idle state

U_i^{cpu}	Normalized CPU utilization
R_i^{cpu}	Normalized CPU resources of the PM
R_i^{ram}	Normalized RAM resources of the PM
W_{tot}	Total waste of resources of all PMs in a CDC
x_{ij}	Assigning V to P or not
y_{ij}	Status of the machine (Active if y=1, Otherwise)

Tableau 9 : Différents symboles utilisés dans ce chapitre

V.5. Évaluation des performances et discussion

Dans cette partie du travail, des tests expérimentaux ont été effectués pour déterminer les performances de l'algorithme proposé, appelé GWOVMP. Cet algorithme a été mis en œuvre en Java comme tous les autres algorithmes comparés. Le simulateur utilisé dans ce travail est CloudSim et la version 5.0 de la boîte à outils du simulateur CloudSIM a été utilisée[237], [238]. Il prend en charge de nombreuses fonctionnalités IaaS telles que l'approvisionnement en ressources à la demande et les solutions de consommation d'énergie. Il nous permet également d'évaluer les performances des nouvelles applications et stratégies de cloud avant qu'elles ne soient développées dans l'environnement réel. Les différents tests et expériences ont été réalisés sur un ordinateur équipé d'un processeur AMD Ryzen 5800U cadencé à 4,4 GHz et de 16 Go de RAM. Le système d'exploitation utilisé était Windows 11.

Pour évaluer les performances de cette méthode, elle est comparée à des algorithmes tels que First fit decreasing FFD[239], Modified best fit decreasing (MBFD)[240], Random first fit (RFF)[241] et Best Fit Decreasing (BFD)[242]. Pour tester l'efficacité de l'algorithme GWOVMP, des instances de différentes tailles allant de 100 à 4000 ont été créées. Chaque VM est équipée d'un processeur à 16 cœurs et de 32 Go de RAM. Un CPU de 1 à 4 cœurs est nécessaire pour chaque VM, et une RAM de 1 à 8 Go est générée aléatoirement. Dans les cas où le CPU est l'une des ressources les plus importantes, le rapport entre les différents besoins en mémoire et en CPU est presque de 10:9.

Les paramètres associés à l'algorithme GWOVMP sont a = 2, qui diminue linéairement à chaque itération, et les vecteurs aléatoires r1 et r2 sont compris dans [0,1]. Ceci donne à cet algorithme l'avantage de définir peu de paramètres. Dans le cas des autres algorithmes, les paramètres sont fixés en fonction de la littérature de base. Il est tenu compte de la possibilité d'atteindre 100 % de l'utilisation des ressources et sa mise en œuvre utilise 100 itérations qui s'arrêtent prématurément à la cinquième itération.

V.5.1. Performances en matière d'utilisation des ressources

Le Tableau 10 montre les résultats obtenus après avoir comparé certains algorithmes existants avec le modèle proposé. Dans ce cas, 100 à 2000 machines virtuelles sont utilisées pour voir l'évolution des résultats et pour pouvoir conclure si un petit nombre d'évolutions est appliqué à toutes les méthodes utilisées pour la comparaison. Dans l'état actuel des connaissances, FFD produit le plus mauvais résultat de tous les algorithmes pour toutes les instances. GWOVMP donne de bons résultats dans de nombreux cas, en particulier lorsque le nombre de machines virtuelles actives augmente. L'algorithme BFD est le deuxième plus performant. L'algorithme RFF donne les pires résultats dans tous les cas, ce qui montre également que cet algorithme n'est pas adapté à un environnement hétérogène.

No	VM	BFD	RFF	MBFD	FFD	GWOVMP
1	100	16	16	16	18	16
2	200	32	33	33	37	32
3	300	48	46	46.03	53	48
4	400	63	64	64.4	74	63
5	500	78	77	77.63	91	78
6	600	94	95	95.77	111	99
7	1000	158	159	161.23	184	153.26
8	2000	316	317	319.64	366	278.08

Tableau 10 : Nombre de machines physiques actives.

L'algorithme GWOVMP, qui est la solution proposée et mise en œuvre, génère de meilleurs résultats que toutes les autres solutions, sauf dans le cas de l'utilisation d'un petit nombre de machines virtuelles. Lorsque la taille du problème augmente, l'algorithme GWO donne un résultat satisfaisant.

La Figure 45 montre le nombre actif de serveurs physiques utilisés pour prendre en charge les différents ensembles de machines virtuelles pour les différents algorithmes utilisés. La solution PM optimale utilisée pour le cas GWOVMP est observée lorsqu'un nombre important de machines virtuelles est utilisé.

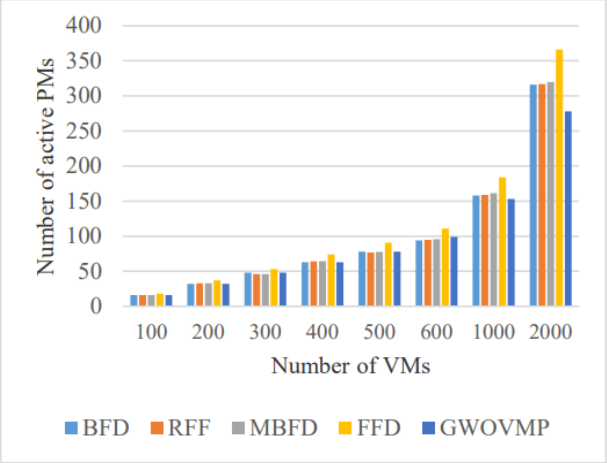

Figure 45 : Nombre de PM actifs

Si l'on se réfère à la Figure 46, le modèle GWOVMP proposé est le plus efficace en termes d'utilisation de l'unité centrale. Ce modèle fournit une utilisation optimale de l'unité centrale pour les différents algorithmes, ce qui devient très visible lors de l'utilisation d'un grand nombre de machines virtuelles. Bien que l'algorithme FFD soit le moins efficace, il convient de noter qu'il donne une solution optimale lors de l'allocation d'un très petit nombre de machines virtuelles sur les machines virtuelles, par rapport aux algorithmes BFD, RFF et MBFD.

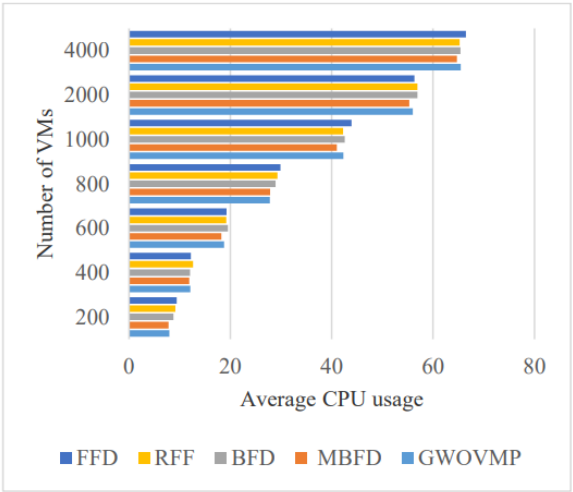

Figure 46 : Utilisation moyenne du CPU

La Figure 47 montre l'utilisation moyenne de la mémoire RAM des serveurs actifs dans l'allocation obtenue par BFD, FFD, RFF, MBFD et GWOVMP pour les différents tests. FFD et RFF ont obtenu l'utilisation de mémoire la plus élevée. Pour FFD, une faible utilisation de la mémoire est obtenue en utilisant un très petit nombre de VM jusqu'à 1000 VM. MBFD donne de meilleures solutions optimales pour des nombres petits à moyens de machines virtuelles. La différence dans cet algorithme est observée lorsque le nombre de machines virtuelles augmente considérablement. Dans ce cas, il devient le deuxième algorithme à offrir une solution plus optimale. Dans le cas de l'algorithme GWOVMP proposé, l'utilisation de la mémoire reste dans des limites raisonnables dans tous les cas possibles.

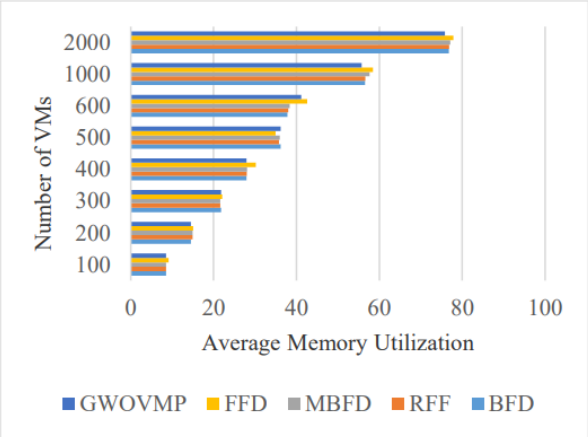

Figure 47 : Utilisation moyenne de la mémoire RAM

La Figure 48 montre que la méthode FFD atteint systématiquement une utilisation exceptionnellement élevée du stockage dans de multiples scénarios, suivie de près par l'algorithme MBFD. Cependant, c'est la solution proposée, GWOVMP, qui se distingue, en particulier lorsqu'elle est confrontée à une augmentation substantielle du nombre de machines virtuelles actives.

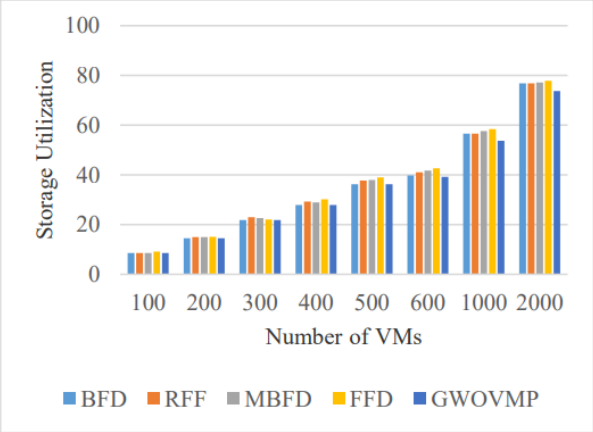

Figure 48 : Utilisation du stockage

De manière remarquable, GWOVMP affiche un taux d'amélioration moyen de 1,8 % par rapport à toutes les méthodes étudiées. Cela souligne son efficacité et sa supériorité dans l'optimisation de l'utilisation du stockage, même face à des demandes opérationnelles accrues. Ces résultats mettent non seulement en évidence les lacunes des méthodes de pointe telles que le FFD, mais soulignent également le potentiel de GWOVMP en tant que solution transformatrice aux défis liés à l'efficacité du stockage dans le cloud.

V.5.2. Gaspillage des ressources

La Figure 49 présente les résultats des taux d'utilisation des ressources gaspillées dans un centre de données. Nous pouvons constater que l'algorithme FFD présente systématiquement un taux élevé de gaspillage des ressources dans plusieurs scénarios, suivi de près par l'algorithme MBFD. Cependant, c'est l'algorithme GWOVMP qui se distingue, en particulier lorsqu'il est confronté à une augmentation substantielle du nombre de machines virtuelles actives. GWOVMP atteint un taux d'amélioration moyen de 1,6 par rapport à toutes les méthodes étudiées. Ces résultats soulignent son efficacité à limiter le gaspillage des ressources, ce qui en fait une meilleure solution que les méthodes comparées.

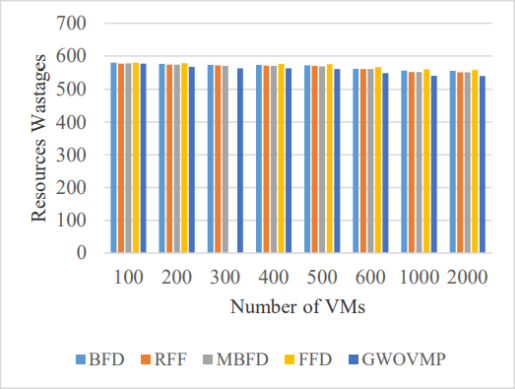

Figure 49 : Ressources gaspillées

V.5.3. Performance en matière de consommation d'énergie

Cette partie du chapitre révèle des résultats clés sur la consommation d'énergie de divers algorithmes qui ont été utilisés pour optimiser l'environnement des centres de données cloud. La Figure 50 montre une tendance notable dans laquelle la consommation d'énergie des algorithmes BFD, RFF, MBFD, FFD et GWOVMP augmente en corrélation directe avec le nombre croissant de machines virtuelles actives.

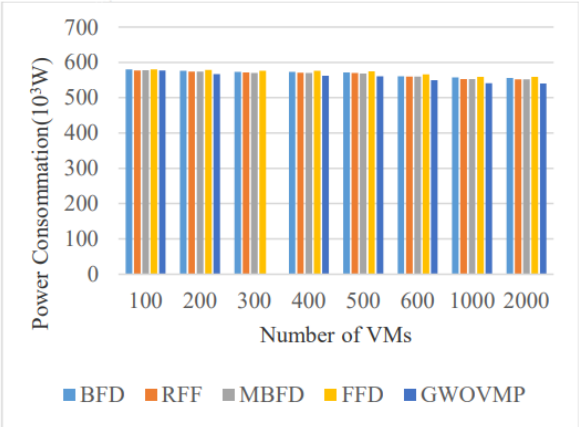

Figure 50 : Consommation d'énergie

Le nombre croissant de machines virtuelles actives. De manière intrigante, GWOVMP se démarque de ces méthodes de l'état de l'art en démontrant une réduction remarquable de 20,99 % de la consommation d'énergie au sein de l'infrastructure cloud.

Ils mettent en évidence la consommation d'énergie maximale, qui culmine à 6580 kWh pour BFD, et la consommation d'énergie minimale, notamment atteinte par RFF avec 6588 kWh. MBFD et FFD enregistrent des consommations d'énergie de 65983 kWh et 6899 kWh, respectivement, pour

un nombre équivalent d'utilisateurs actifs et de demandes de PM. Ces statistiques fournissent des preuves quantitatives précieuses de l'efficacité de GWOVMP par rapport à ses pairs.

Collectivement, ces résultats soulignent non seulement l'escalade de la demande d'énergie dans le cloud computing, mais aussi le potentiel de GWOVMP en tant que solution plus durable et plus économe en énergie pour les infrastructures cloud.

V.5.4. Discussion et limites

Le choix de CloudSim comme environnement de simulation a été dicté par ses performances, telles qu'observées dans les travaux de la littérature. Il nous a permis d'automatiser le cycle de vie de la simulation tout en réduisant le temps de traitement. Dans cet environnement de simulation, le modèle proposé a été configuré dans une bibliothèque selon les paramètres de configuration proposés.

Les résultats obtenus dans cette étude permettent d'affirmer que l'algorithme GWOVMP est meilleur que les autres algorithmes pour résoudre le problème d'optimisation VMP. Comme la plupart des travaux qui ont été utilisés pour comparer l'algorithme proposé, les travaux réalisés ne prennent pas en compte le réseau et certaines métriques d'évaluation (par exemple la migration, la QoS, l'allocation des ressources en fonction du risque). Pour y remédier, cette méthode doit être améliorée en impliquant beaucoup plus de paramètres d'évaluation. Ce travail s'est concentré sur le problème du placement des machines virtuelles afin de réduire l'utilisation de l'énergie, de l'unité centrale, de la mémoire vive et de l'espace de stockage. L'autre paramètre pris en compte a été le gaspillage des ressources.

Malgré les performances de GWOVMP, il doit être amélioré pour donner de bien meilleurs résultats sur l'utilisation du stockage et le gaspillage des ressources, ce qui donne un faible taux d'amélioration. Pour résoudre ce problème, il faudra améliorer cette méthode en ajoutant certains paramètres, comme la bande passante du réseau. La migration des machines virtuelles devra également être prise en compte pour renforcer le modèle.

Mais aussi, il serait préférable d'implémenter et de comparer la méthode qui a été proposée dans ce travail avec d'autres simulateurs utilisés dans certains travaux de recherche et de comparer les résultats.

V.6. Conclusion

Dans un centre de donnéescloud, les machines physiques ainsi que le système de refroidissement consomment beaucoup d'énergie. La réduction significative du nombre de machines physiques allumées aura un impact important sur la consommation d'énergie dans ces environnements basés sur le cloud computing. Le placement optimal des machines virtuelles dans un environnement cloud est considéré comme l'une des tâches les plus difficiles. De nombreuses recherches ont été menées pour résoudre ce type de problème d'optimisation.

Les métaheuristiques sont devenues une solution très importante pour ce type de problème d'optimisation difficile. L'algorithme inspiré du GWO qui a été proposé dans ce travail est l'une des méthodes inspirées de la nature. Dans ce chapitre, nous avons mis en œuvre une approche métaheuristique appelée GWOVMP pour fournir une solution optimale ou quasi-optimale au problème VMP dans un environnement cloud computing. L'objectif de GWOVMP est de pouvoir diminuer la consommation d'énergie dans un environnement cloud tout en minimisant les PM actifs et en minimisant le gaspillage des ressources CPU et mémoire. Les résultats de ce travail montrent que la technique proposée permet une utilisation optimale des ressources des PM en plaçant la VM sur un PM approprié tout en réduisant la consommation d'énergie par rapport à d'autres algorithmes.

CHAPITRE VI : PLACEMENT MULTI-OBJECTIF DE MACHINES VIRTUELLES PAR LE SEAGULL ALGORITHME OPTIMISATION

VI.1. Résumé

Le placement de machines virtuelles (VMP) consiste à sélectionner la machine physique (PM) la plus appropriée pour exécuter une machine virtuelle (VM) dans les centres de données cloud (CDC). Malheureusement, les méthodes VMP actuelles ne prennent en compte qu'un nombre limité de ressources, ce qui entraîne un déséquilibre de la charge et l'activation inutile de certaines PM dans le centre de données (CD). Dans ce chapitre, une nouvelle approche appelée Machine Virtuelle à Algorithme d'Optimisation Multi-Objectif Seagull (MOSOAVMP) est proposée pour résoudre ces problèmes et améliorer la gestion des ressources dans les CDC. L'objectif est d'optimiser l'utilisation des ressources, de minimiser la consommation d'énergie, de réduire les violations des accords de niveau de service et d'améliorer l'efficacité globale des CDC. Le but est d'obtenir un déploiement optimal qui répondra à ces différents objectifs tout en réduisant les coûts associés à l'exploitation des CDC. Les résultats obtenus montrent l'efficacité du MOSOAVMP proposé par rapport aux algorithmes existants pour les différentes mesures considérées. Ces résultats expérimentaux montrent que MOSOAVMP réduit les gaspillages de ressources et la consommation d'énergie de 5,44%, améliore l'utilisation moyenne du CPU de 14,84%, l'utilisation de la mémoire de 11,54%, l'utilisation moyenne du stockage de 5,37% et l'utilisation moyenne de la bande passante de 6,88%.

Mots clés : Cloud computing, algorithme d'optimisation Seagull, algorithme métaheuristique, SLA, placement de machines virtuelles, centre de données, consommation d'énergie.

VI.2. Introduction

Ces dernières années, le CC s'est imposée comme un paradigme informatique majeur pour l'hébergement et la fourniture de services via l'internet. Offrant une grande évolutivité et une grande variété de services, son adoption est désormais omniprésente et ne cesse de croître. L'impact du cloud sur la vie de tous les jours est significatif, notamment par le biais des réseaux sociaux et des réseaux de capteurs. La généralisation des appareils intelligents a accentué l'utilisation du modèle du CC. Par conséquent, le nombre et la taille des centres de données cloud augmentent rapidement. Le cloud réduit également les coûts d'infrastructure des réseaux. Diverses techniques et technologies sont utilisées pour répondre aux besoins des utilisateurs et optimiser les performances du cloud. Il s'agit notamment de la migration, qui consiste à transférer une

machine virtuelle (VM) d'un serveur physique (PM) disposant de ressources insuffisantes vers un autre disposant des ressources nécessaires. De plus, la virtualisation est au cœur du CC. La virtualisation a révolutionné les opérations informatiques dans les centres de données (CD), où des machines virtuelles (VM) sont déployées sur des machines physiques (PM) pour exécuter les applications des utilisateurs[243]. Le placement optimal de ces machines virtuelles sur des hôtes physiques est crucial pour garantir de bonnes performances. Ce problème VMP est un défi d'optimisation dont l'objectif est de déterminer la meilleure allocation de VM parmi les PM disponibles. Le placement peut être statique ou dynamique, selon que les MP peuvent être remplacés en raison de changements dans l'environnement du CC, tels que des variations de la charge de travail, de la disponibilité des ressources ou des contraintes matérielles. La migration des machines virtuelles peut être nécessaire pour équilibrer la charge, même si cela peut entraîner une violation des accords de niveau de service (SLA)[244]. Toutefois, il est important d'éviter une utilisation excessive des ressources, car cela peut entraîner une dégradation des performances. Le cloud facilite également le stockage et l'analyse de grands volumes de données[215], ce qui signifie que plusieurs défis, tels que la sécurité[245], doivent être pris en compte.

Il est donc essentiel de trouver une solution optimale pour le VMP qui réponde aux objectifs susmentionnés. Cela implique de trouver un équilibre entre les différents objectifs, car l'amélioration de l'un d'entre eux peut entraîner la détérioration d'un autre. Une approche d'optimisation multi-objectifs est donc nécessaire pour résoudre efficacement ce problème de PMV.

Ce chapitre présente un nouvel algorithme, conçu pour relever le défi de l'optimisation du placement des VM sur les PM afin d'améliorer la gestion des ressources informatiques. L'algorithme proposé a trois objectifs principaux. Le premier objectif consiste à réduire la consommation d'énergie dans les CDC en minimisant le nombre de serveurs physiques actifs. Le deuxième objectif est de réduire le gaspillage des différentes ressources du cloud computing tout en optimisant le VMP. Enfin, l'algorithme vise à minimiser et à réduire les accords de niveau de service (SLA) entre les PM actifs dans les CDC. En outre, l'algorithme prend en compte d'autres facteurs tels que l'utilisation de l'unité centrale, la mémoire, la bande passante, l'espace de stockage, ainsi que le nombre de machines actives et de migrations.

L'utilisation de cette solution a permis de réduire considérablement le nombre de migrations vers les CDC. L'algorithme MOSOAVMP proposé a été comparé à d'autres algorithmes couramment utilisés tels que DMOSCA-SSA[245], MOILP[246], PIAS[247], MGGAVP[248] et MBFD[249].

Les résultats de la simulation démontrent les performances élevées de l'algorithme proposé par rapport aux approches susmentionnées.

Le reste de ce chapitre est organisé comme suit : La deuxième section passe en revue l'état des lieux des différents travaux existants. La section 3 présente une introduction générale aux différents concepts mathématiques de l'algorithme d'optimisation de la mouette (SOA), suivie de la formulation mathématique de l'optimisation multi-objectifs appliquée à la VMP, et la description de l'algorithme proposé se trouve dans la section 4. La section 5 présente les résultats de l'évaluation des performances de la méthode proposée par rapport à d'autres algorithmes couramment utilisés. Enfin, la section 6 présente les conclusions et les perspectives pour les travaux futurs.

VI.3. Travaux connexes

Comme indiqué plus haut, le VMP vise à garantir de bonnes performances du cloud et une gestion optimale des ressources. Il existe de nombreux algorithmes pour résoudre ce problème, et dans cette section, nous passons en revue certains des travaux qui traitent du problème VMP.

Lu et al [233] ont proposé un algorithme génétique amélioré (I-GA) pour résoudre le problème VMP, afin d'optimiser la disponibilité et la consommation d'énergie des CDC. Une combinaison d'architecture de hiérarchie virtuelle et d'algorithme génétique a été utilisée pour parvenir à une solution quasi optimale, afin d'améliorer la consommation d'énergie et la disponibilité des ressources. Une étape clé de leur approche est la génération de la population I-GA initiale, qui est réalisée à l'aide d'une analyse par éléments finis en arrière-plan. CloudSim est utilisé pour simuler les expériences. Les résultats démontrent une amélioration significative de la gestion de l'énergie et de la haute disponibilité dans les DC. Sur la base des résultats de référence des différentes méthodes de pointe utilisées pour le problème d'optimisation VMP, leur modèle présente de meilleurs résultats. En utilisant l'approche I-GA, les auteurs optimisent efficacement le VMP, ce qui contribue à une meilleure gestion des ressources et à une réduction de la consommation d'énergie dans les CDC.

Une approche multi-objectifs a été proposée par Caviglione et al pour déterminer les meilleures techniques pour le VMP, en tenant compte de divers objectifs, tels que l'impact des défaillances matérielles, la consommation d'énergie en courant continu et la satisfaction de l'utilisateur en termes de performance. Un cadre de renforcement profond facilite la sélection de la meilleure heuristique pour le placement des machines virtuelles. Les résultats montrent que la méthode

surpasse les heuristiques utilisées dans l'état de l'art pour les charges de travail réelles et synthétiques[233].

Pour optimiser l'efficacité énergétique des centres de distribution d'entreprise, Alharbi et al ont introduit une méthode combinant l'affectation de différentes applications et la VMP d'un centre de distribution. Ils ont formulé le problème lié à l'optimisation énergétique des centres de distribution d'entreprise comme un problème Int2LBP (Integrated Two-Layer Bin Packing). Cette approche a été considérée comme une solution initiale et a été développée ultérieurement pour optimiser l'efficacité énergétique des centres de distribution d'entreprise. En termes d'efficacité énergétique, les diverses simulations effectuées sur les centres de distribution prouvent la performance de l'algorithme Int2LBP_FFD par rapport à l'algorithme Consec2LBP_FFD. En outre, l'algorithme Int2LBP_ACS est plus performant que l'algorithme Int2LBP_FFD en matière de gestion efficace de l'énergie. Les algorithmes Int2LBP_FFD et Int2LBP_ACS sont bien adaptés à la gestion de différentes applications et VM sur les DC des grandes entreprises[250].

Ghetas et al ont introduit une nouvelle méthode appelée MBO-VM basée sur l'algorithme Monarch Butterfly Optimization (MBO) pour VMP. Cette méthode vise à maximiser les performances de l'emballage et à réduire les MP. Le simulateur CloudSim a été utilisé pour mettre en œuvre et évaluer les performances de cette approche. Cet outil leur a également permis d'effectuer des tests sur des charges de travail réelles et synthétiques dans le cloud. Les résultats obtenus à partir des simulations démontrent que MBO-VM surpasse les techniques VMP connues en termes de performance. En utilisant MBO-VM, il est possible de réduire plus efficacement le nombre d'hôtes actifs tout en maximisant l'efficacité du conditionnement. Cette approche offre donc une solution optimale pour le VMP, améliorant l'efficacité globale des CDC[235].

Pour résoudre le problème d'optimisation VMP, un nouvel algorithme ILP multi-objectif a été présenté par Regaieg et al , en considérant que chez les fournisseurs de services en cloud (CSP), l'environnement est composé de VM de types homogènes et hétérogènes. Ce modèle vise à réduire le gaspillage des ressources tout en maximisant le nombre de machines virtuelles hébergées sur des serveurs physiques, réduisant ainsi le nombre de PMs utilisés. En raison de la diversité des différents types de machines virtuelles disponibles dans un environnement cloud hétérogène, les résultats des tests ont montré l'efficacité de la réalisation des objectifs susmentionnés. Ces résultats ont l'avantage de réduire les coûts associés à l'exploitation des DC tout en maintenant un niveau élevé de qualité de service (QoS)[246].

Rashida et al. ont présenté un algorithme VMP appelé MGGAVP, qui prend en compte la corrélation et traite les problèmes VMP. Cet algorithme est basé sur l'hybridation de l'algorithme d'escalade et le regroupement génétique et est étendu pour fonctionner dans un environnement multi-cloud. Après simulation, les résultats révèlent les performances nettes offertes par leur algorithme par rapport à celles des autres algorithmes de référence. Il permet de réaliser des économies d'énergie de 51,93 % et de réduire les coûts énergétiques de 70,41 %[251].

Rahimi Zadeh et al ont proposé un nouveau schéma de planification appelé PIAS (Profit-aware Interference-aware Scheduling) pour la consolidation efficace des machines virtuelles dans le modèle Infrastructure-as-a-Service (IaaS). Le schéma PIAS prend en compte plusieurs facteurs tels que le profit, les coûts énergétiques, les interférences opérationnelles, l'utilisation des ressources et les accords de niveau de service. Pour minimiser les coûts d'exécution de la charge de travail et augmenter le profit du fournisseur, le problème d'optimisation a été modélisé sous la forme d'une programmation dynamique stochastique, imitant le comportement opérationnel des machines virtuelles. Des simulations basées sur des charges de travail réelles montrent que PIAS surpasse en moyenne les approches concurrentes, avec des améliorations d'au moins 40 % pour le profit, 29 % pour l'efficacité énergétique et 35 % pour le temps d'arrêt du service[247].

Pour améliorer la qualité de service (QoS) et la gestion efficace du trafic dans les réseaux de l'internet des objets (IoT), Gharehpasha et al proposent une métaheuristique basée sur l'optimisation de la mouette améliorée. Cette approche permet une meilleure gestion de l'acheminement des paquets et une amélioration significative de la QoS en termes de délai. Les performances de cette méthode ont été évaluées et comparées à celles des méthodes précédentes, démontrant ainsi sa précision, son efficacité et sa supériorité[225].

Nabavi et al ont présenté une approche multi-objectifs du VMP dans les DCs edge-cloud. Pour optimiser le trafic réseau et la puissance du trafic, un modèle basé sur la SOA a été présenté. La stratégie se concentre sur la réduction du trafic réseau entre les PM tout en consolidant les communications des VM sur les mêmes PM. Cela permet de réduire le transfert de données sur le réseau et de diminuer la consommation d'énergie des PM. Les auteurs ont effectué des simulations à l'aide de CloudSim et ont testé l'approche proposée sur deux topologies de réseau, VL2 et à trois niveaux. Les résultats démontrent l'efficacité de la méthode proposée pour réduire de manière significative la consommation d'énergie et le trafic réseau dans les environnements edge-cloud. Les résultats des tests de l'algorithme proposé montrent une nette réduction de l'efficacité

énergétique de 5,5 %, une réduction remarquable de 70 % du trafic réseau et une réduction de 80 % de la consommation d'énergie des composants du réseau[252].

Le tableau 11 présente un résumé de ces différents travaux de l'état de l'art. Il présente la méthodologie utilisée, les objectifs, les faiblesses, le simulateur utilisé et la méthode proposée.

Le choix de l'algorithme MOSOAVMP pour optimiser le placement des machines virtuelles (VM) dans le cloud computing repose sur les qualités inhérentes de l'algorithme SOA. Il s'agit notamment de sa simplicité, de sa facilité de mise en œuvre et de sa capacité à converger rapidement vers des solutions optimales, même avec de nombreuses itérations. Cela témoigne de sa robustesse et de sa capacité à résoudre des problèmes complexes tels que le placement de machines virtuelles. L'analyse présentée dans le tableau 1 révèle une évaluation approfondie des mesures utilisées dans les travaux antérieurs, ce qui nous a permis de discerner les mesures qui sont primordiales et celles qui sont souvent négligées. Cette analyse nous a permis de sélectionner un large éventail de mesures pertinentes pour notre étude, garantissant ainsi une évaluation complète et rigoureuse des performances du cloud.

VI.4. Optimisation du VMP des par le SAO

VI.4.1. Paradigme bio-inspiré

L'optimisation en général est un domaine qui n'a cessé de se développer ces dernières années, avec de nouveaux algorithmes bio-inspirés[253], [254], [255] ainsi que des algorithmes hybrides[256], [257]. Inspiré par les comportements de migration et d'attaque des mouettes dans la nature, SOA est une métaheuristique proposée par Dhiman et Kumar en 2017[258]. Cet algorithme est basé sur la population et utilise les mouvements migratoires pour rechercher des sources de nourriture abondantes. Leurs positions sont mises à jour en fonction de la meilleure position trouvée, tandis que des stratégies de fascination et d'attaque sont utilisées pour attirer les proies. L'algorithme est divisé en deux phases principales, la diversification et l'intensification, et peut passer d'une phase à l'autre en fonction de la position de la proie. Les mouettes peuvent modifier continuellement leur angle d'attaque pendant la migration[259].

Le comportement migratoire des mouettes se caractérise par les éléments suivants[260]:

- Les mouettes se déplacent en groupes pendant la migration, adoptant une stratégie de mouvement collectif pour éviter les collisions entre elles ;
- Au sein du groupe, les mouettes peuvent ajuster leur trajectoire en se déplaçant dans la direction qui assure la meilleure survie à la mouette la plus âgée. En d'autres

termes, elles peuvent suivre la direction choisie par la mouette la plus expérimentée pour maximiser leurs chances de survie.

Cet oiseau bruyant vit, se nourrit et dort en colonies. Sa nourriture se compose d'insectes, de leurs larves, de vers de terre, de petits crustacés, de mollusques et de petits poissons capturés en vol. Comme le montre la Figure 51[258], lors de l'attaque, ils effectuent un mouvement en spirale.

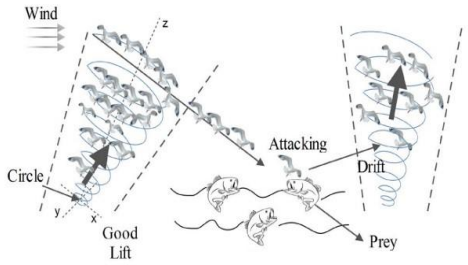

Figure 51 : Les phases de migration et d'attaque du comportement des mouettes [258]

VI.4.2. Formulation Mathématique du seagull algorithm optimization

Cet algorithme simule le déplacement d'un groupe de mouettes d'une position à une autre lors de la migration. Durant cette phase, trois conditions doivent être remplies par chacune des mouettes :

A. Phase de Migration (ou Phase d'Exploration)

Évitement des collisions: Une variable appelée est introduite pour éviter les collisions avec d'autres mouettes lors du calcul de la nouvelle position de l'agent de recherche[261].

$$\overrightarrow{C_s} = A \times \overrightarrow{P_s}(x) \tag{69}$$

Dans l'équation (69), $\overrightarrow{C_s}$ est une représentation de la position de l'agent de recherche qui évite les collisions entre voisins, P_s représente sa position actuelle, x désigne l'itération actuelle et A décrit le comportement des différents mouvements de l'agent dans l'espace de recherche donné. La Figure 52 montre comment les agents de recherche évitent les collisions entre eux.

Algorithme	Méthodologie appliquée	Objectifs	Vulnérabilité	Simulateur
I-GA[233]	Proposition d'un algorithme I-GA pour optimiser les problèmes VMP.	Assurer la disponibilité et optimiser la consommation d'énergie d'un CC.		CloudSim
ACS, Int2LBP[250]	Pour résoudre le problème de la consommation d'énergie dans un centre de données d'entreprise, les auteurs l'appellent Int2LBP.	Mise en œuvre de l'algorithme Int2LBP_FFD et de l'algorithme Int2LBP_ACS, sur la base des premiers résultats du premier algorithme, pour optimiser l'efficacité énergétique des centres de distribution des entreprises.	Exécution en temps réel, les applications sont allouées dynamiquement, VMP dynamique, consolidation de VM et migration de VM	Amazone EC2 T2
MBO-VM[235]	Consolidation des serveurs et utilisation des ressources	Réduire le nombre d'hôtes physiques actifs dans le DC pour assurer une consommation d'énergie optimale et réduire les coûts de maintenance du DC.	Stockage, bande passante et accords de niveau de service	CloudSim
MOILP[246]	Optimiser les machines virtuelles hébergées, limiter le gaspillage des différentes ressources et réduire le nombre de serveurs physiques actifs pour diminuer la consommation d'énergie dans les centres de données.	Minimiser les coûts d'exploitation du DC pour augmenter la satisfaction des clients.	L'effet de la charge de travail sur la consommation d'énergie efficace n'est pas pris en compte.	Amazone EC2, Julia Language (JL)
MGGAVP[13]	Optimisation du problème VMP dans un environnement hétérogène.	Hybridation des algorithmes d'escalade métaheuristiques et du regroupement génétique pour l'optimisation dans un environnement multi-cloud.	QoS, SLA	MATLAB R2015
PIAS[247]		Coût de l'efficacité énergétique, de l'interférence opérationnelle des VM, de l'utilisation réduite des ressources et de l'optimisation des accords de niveau de service.	Absence de prévision des périodes de pointe permettant d'augmenter ou de réduire la capacité des CC pendant les périodes de pointe ou d'inactivité. Pour éviter la saturation, l'utilisation des ressources de la VM doit être inférieure à 1.	CloudSim
SOAVMP[252]		Trafic, puissance et énergie	Faible utilisation des ressources de test, SLA, et gaspillage des ressources	CloudSim
Notre Méthode	MOSOAVMP optimise le placement des machines virtuelles	Consommation d'énergie SLA, utilisation des ressources (CPU, RAM, stockage et bande passante), migration et gaspillage des ressources,	Sécurité et planification des tâches	CloudSim

Tableau 11 : Résumé des différents travaux de l'état de l'art

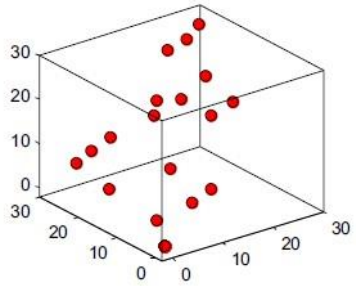

Figure 52 : Éviter les collisions entre agents de recherche[258]

Ces variables sont appliquées pour déterminer la position optimale de l'agent de recherche tout en évitant les collisions avec d'autres agent[261], [262]. Ceci est illustré par les équations (70) et (71).

$$A = f_c - \left(x \times \left(\frac{f_c}{Max_{it}} \right) \right) \tag{70}$$

$$\mathrm{x} = 0,1,2,\ldots,Max_{it} \tag{71}$$

Dans la formule (70), la variable A reproduit le comportement de mouvement de l'agent de recherche. Le paramètre f_c contrôle cette variable et diminue pratiquement de f_c à 0. Le nombre maximum d'itérations utilisées est représenté par Max_{it}. Ces éléments sont essentiels pour réguler le mouvement de l'agent et définir les limites du processus d'optimisation. La Figure 53 montre comment les différents agents se déplacent et recherchent le meilleur voisin.

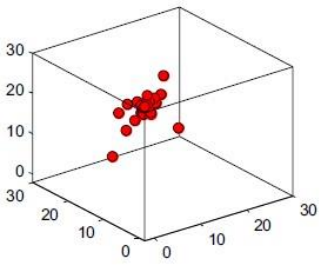

Figure 53 : Agents de déménagement et recherche du meilleur voisin[258]

Les mouettes se dirigent vers la mouette la plus performante de leur groupe, ce qui assure la cohésion. Ce phénomène est représenté par l'équation mathématique (72)[263], [264]:

$$\overrightarrow{M_s} = B \times \left(\overrightarrow{P_{bs}}(x) - \overrightarrow{P_s}(x) \right) \tag{72}$$

Où $\overrightarrow{M_s}$ représente la position $\overrightarrow{P_s}$ de la mouette par rapport au P_{bs} des mouettes les plus performantes.

Le paramètre de contrôle B, comme indiqué dans la référence[265], joue un rôle essentiel dans l'établissement de l'équilibre entre la diversification et l'intensification. Il est représenté mathématiquement dans l'équation (73)[264], [266] où Ran est un nombre aléatoire compris entre [0 ; 1][265]:

$$B = 2 \times A^2 \times Ran \tag{73}$$

S'approcher de la mouette optimale :Les mouettes recherchent activement des sources de nourriture pendant leur migration, en se déplaçant dans la direction qui offre les meilleures chances de survie. La Figure 54 illustre la mise à jour de la position de l'agent de recherche par rapport à la position du meilleur agent de recherche.

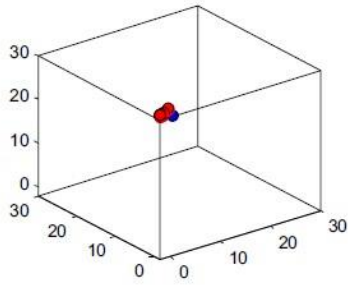

Figure 54 : Approche de l'agent de recherche optimal[258]

Dans ce cas, les mises à jour des positions des autres mouettes sont basées sur la position de la meilleure mouette. La formule ci-dessous (74) représente cette mise à jour[267]:

$$\overrightarrow{D_s} = \left|\overrightarrow{C_s} + \overrightarrow{M_s}\right| \tag{74}$$

B. Phase d'attaque (ou phase d'exploitation)

Dans la phase d'exploitation, les mouettes utilisent leurs connaissances et expériences antérieures. Elles peuvent constamment ajuster leur angle d'attaque et leur vitesse. Pour maintenir leur altitude, elles utilisent leurs ailes et le poids de leur corps.

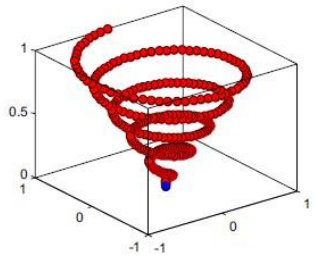

Figure 55 : Comportement naturel d'une attaque de mouettes[258]

Comme l'illustre la Figure 55, les mouettes s'engagent dans un mouvement en spirale lorsqu'elles attaquent leur proie dans les airs. Les équations ci-dessous calculent les trois dimensions (x, y, z) de ce comportement (75)-(78)[268], [269], [270] :

$$x = r \times \cos(\theta) \tag{75}$$

$$y = r \times \sin(\theta) \tag{76}$$

$$z = r \times \theta \tag{77}$$

$$r = v \times e^{\theta u} \tag{78}$$

Plusieurs paramètres sont utilisés pour mettre à jour la position de l'agent de recherche dans la spirale. Le rayon de la spirale est représenté par r , tandis que θ est un élément aléatoire entre $[0 < \theta < 2\pi]$ [271]. Ces paramètres influencent la forme de la spirale. En utilisant le logarithme naturel de la base e , l'équation suivante (79) calcule la position actualisée de l'agent de recherche[272], [273]:

$$\overrightarrow{P_s}(x) = \overrightarrow{(D_s} \times x \times y \times z) + \overrightarrow{P_{bs}}(x) \tag{79}$$

La variable $\overrightarrow{P_s}(x)$ de cette équation joue un rôle essentiel dans le maintien de la meilleure option disponible et dans la prise en compte de la situation actuelle des autres agents de recherche. Elle maintient une référence à la meilleure solution tout au long du processus.

VI.5. Formulation mathématique et le MOSOAVMP pour le placement de VMs

VI.5.1. Formulation Mathématique

Le placement optimal des machines virtuelles dans les CDC est une question clé dans le cloud computing. L'objectif principal est de minimiser la consommation d'énergie, de réduire le gaspillage des ressources, de réduire les violations des accords de niveau de service et de maximiser l'efficacité des centres de données.

Le cloud computing présente des défis complexes en matière de VMP. Le problème du VMP dans un CDC est imprévisible et ne suit aucun modèle cohérent. La complexité de ce problème peut être illustrée par le fait que le nombre maximum de mappages de VM sur les PM est égal à m^n , où n est le nombre de VMs et m le nombre de PM. Dans ce contexte, nous déterminons le premier objectif, qui est de minimiser la consommation d'énergie des PM dans le CDC. Les différents

symboles mathématiques utilisés dans la formulation mathématique sont décrits dans le Tableau 12.

Symbol	Descriptions
P	Groupe de Machines Physiques (PMs)
V	Groupe de Machines Virtuelles (VMs)
α	Coefficient d'équilibrage F1
β	Coefficient d'équilibrage F2
δ	Coefficient d'équilibrage F3
a, b	Variables Binaires
P_i^{iddle}	Consommation d'énergie en mode d'inactivité
P_i^{power}	Consommation d'énergie de PM i
P_i^{busy}	Consommation d'énergie des PMs pendant les périodes d'activité
U_i^{cpu}	L'utilisation normalisée de l'unité centrale des PM i
U_i^{memo}	L'utilisation normalisée du RAM de PM i
$R_i^{wastage}$	Utilisation inefficace des ressources de la PM
NR_i^{cpu}	CPU restante normalisée de la PM
NR_i^{memo}	Mémoire restante normalisée de la PM
TU^{CPU}	Utilisation globale de CPU par les PMs en fonctionnement
TU^{memo}	Utilisation globale de la mémoire des PMs en fonctionnement
TU^{sto}	Utilisation globale du stockage des PMs en fonctionnement
TU^{B}	Utilisation globale des bande passantes des PMs en fonctionnement

T_i^{cpu}	Utilisation maximale du CPU des PMs
T_i^{memo}	Utilisation maximale de la mémoire des PMs
T_i^{sto}	Utilisation maximale du stockage Peak des PMs
T_i^{band}	Utilisation maximale de la bande passante des PMs
C_j	Demande de CPU de la VM j en MIPS
M_j	Demande de la mémoire de la VM j en MB
H_j	Demande de stockage de la VM j en GB
B_j	Demande de la bande passante de la VM j en Mbps

Tableau 12 : Divers symboles mathématiques utilisés

Des recherches récentes ont montré une corrélation linéaire entre l'utilisation de l'unité centrale et la consommation d'énergie des serveurs dans les CDC. Une équation linéaire peut être utilisée pour calculer précisément la consommation d'énergie dans le contexte de le cloud. Cette relation montre que lorsque l'utilisation du CPU augmente, la demande d'énergie de l'hôte augmente proportionnellement[24], [216], [227]:

$$P_i^{power} = \begin{cases} \left(P_i^{busy} - P_i^{iddle}\right) \times U_i^{cpu} + P_i^{iddle} & if\ U_i^{cpu} > 0 \\ 0, & otherwise \end{cases} \quad (80)$$

L'équation (80) est utilisée pour déterminer la consommation d'énergie d'une MP dans un CC. Ces différentes variables P_i^{power}, P_i^{busy}, P_i^{iddle} et U_i^{cpu} et représentent respectivement la quantité d'énergie consommée lorsque la machine est entièrement chargée, au repos et inactive, et l'utilisation du CPU en MIPS. D'après cette équation, la consommation d'énergie est proportionnelle à l'utilisation du processeur. Ainsi, une hausse de l'activité du CPU se traduira par une augmentation de la consommation d'énergie cloud.

La consommation d'énergie dans les CDC est directement liée à la consommation d'énergie du CPU. L'augmentation de l'utilisation de l'unité centrale des PM influencera directement l'augmentation de la consommation d'énergie des CDC. Par conséquent, la consommation d'énergie globale des CDC peut être calculée sur la base de l'équation(81) :

$$\sum_{i=1}^{m} P_i^{power} = \sum_{i=1}^{m} b_i \times ((P_i^{busy} - P_i^{idle}) \times \sum_{v=1}^{m} a_{jp}.C_j + P_i^{idle} \quad (81)$$

La prévention du gaspillage des ressources est un aspect crucial du placement des machines virtuelles dans les CDC. Chaque serveur dispose de ressources matérielles qui doivent être utilisées de manière optimale pour héberger les machines virtuelles. Une gestion efficace des ressources inutilisées du serveur est essentielle. L'équation 82 quantifie le gaspillage des ressources[274].

L'équation (82) est utilisée pour quantifier les ressources inutilisées restantes sur un PM p dans un CDC. NR_i^{cpu} p représente la quantité de CPU inutilisée, tandis que NR_i^{memo} p représente la quantité de mémoire inutilisée par les VM sur ce même PM . Les variables U_i^{cpu} et U_i^{memo} représentent l'utilisation du CPU et de la mémoire, respectivement, par les VM sur le PM dans le CDC. L'équation (82) représente la quantité totale des différentes ressources consommées dans le (CDC).

$$\sum_{i=1}^{m} R_i^{wastage} = \sum_{p=1}^{m} \left[a_i \times \frac{\left| (T_i^{cpu} - \sum_{j=1}^{n}(b_{ji}.C_j)) - (T_i^{memo} - \sum_{j=1}^{n}(b_{ji}.B_j)) \right| + \varepsilon}{(\sum_{j=1}^{n}(b_{ji}.C_j) + (\sum_{j=1}^{n}(b_{ji}.M_j)} \right] \quad (82)$$

Lorsque les valeurs a_i et b_{vi} sont égales, cela indique que le PM est actif et qu'il héberge la VM v. Pour garantir un niveau de service minimum entre le fournisseur de services cloud et le client, un contrat SLA doit être respecté. L'équation (83) représente ce SLA:

$$SLA = \frac{1}{1 + e^{U_{cpu} - 0.9}} \quad (83)$$

Les objectifs du placement des VM sur les PM dans les CDC sont définis à partir des équations précédentes et comprennent les éléments suivants:

$$Min\ F(x) = \alpha F_1(x) + \beta F_2(x) + \delta F_3(x) \quad (84)$$

La fonction objective utilisée dans l'équation (84) prend en compte plusieurs objectifs dans le placement des VM. Les coefficients α, β et permettent d'équilibrer ces objectifs. Le premier objectif, F1, concerne la consommation d'énergie des PM. Le deuxième objectif, F2, mesure le gaspillage des ressources. Enfin, le troisième objectif, F3, prend en compte le respect des accords de niveau de service. Les valeurs de ces fonctions peuvent être calculées à l'aide des formules mathématiques suivantes (85)-(87) :

$$Min \sum_{i=1}^{m} P_i^{power} \tag{85}$$

$$Min \sum_{i=1}^{m} R_i^{wastage} \tag{86}$$

$$Min \sum_{i=1}^{m} SLA \tag{87}$$

Lors du placement optimal des VM sur les PM dans un CDC, plusieurs contraintes doivent être prises en compte, notamment:

$$\sum_{i=1}^{m} b_{ji} = 1 \tag{88}$$

$$\sum_{j=1}^{m} b_{ji} = b_{ij} \,.\, C_j \leq T_i^{cpu} . a_i \tag{89}$$

$$\sum_{j=1}^{m} b_{ji} = b_{ij} \,.\, M_j \leq T_i^{memo} . a_i \tag{90}$$

$$\sum_{j=1}^{m} b_{ji} = b_{ij} \,.\, H_j \leq T_i^{sto} . a_i \tag{91}$$

$$\sum_{j=1}^{m} b_{ji} = b_{ij} \,.\, B_j \leq T_i^{band} . a_i \tag{92}$$

Chaque VM est assignée à un seul PM, comme le montre l'équation (88). Les équations (89), (90), (91) et (92) garantissent que le total des ressources (CPU, mémoire, espace de stockage et bande passante) requises par les différentes VM sur un PM ne peut pas dépasser la capacité de ce PM.

VI.5.2. Algorithme proposé

Dans les CDC, il existe un ensemble de machines virtuelles homogènes ou non, qui utilisent les ressources matérielles des PM pour exécuter leurs différents services. L'hébergement de ces VM relève de la responsabilité des PM dans ces centres. Davantage de ressources matérielles sont nécessaires pour répondre à la demande croissante de services en cloud, ce qui entraîne une

augmentation des coûts. L'optimisation du VMP maximisera l'utilisation des ressources disponibles tout en permettant à chaque PM d'héberger un nombre important de VM. Cette approche évite de gaspiller les diverses ressources disponibles dans le cloud et exploite au mieux la puissance de traitement, la mémoire et les systèmes de stockage des serveurs virtuels. Cela garantit la performance d'un DC basé sur le cloud.

Cette étude présente une nouvelle solution basée sur MOSOAVMP, une approche multi-objectifs discrète pour le placement optimal des machines virtuelles. L'objectif principal est d'optimiser la gestion des ressources, de minimiser la consommation d'énergie et de réduire le SLA dans le CDC. L'algorithme 1 montre la solution proposée dans ce travail.

Input: Initializing the agent Population

Output: The best solution for optimizing VMP

Begin

1. Initialize SOA parameters
2. Set the population size of the Seagulls
3. Compute the cost function of each agent
4. Determine the maximum number of iterations
5. Initialize the seagull position randomly

while t < maxiter **do**

For each search agent **do**

Apply mutation and crossover on the solutions as migration and seagull behavior

Calculate objective function for all search agents

End for

Update seagull position

For each i=1 to number of seagulls **do**

Compute the costs involved with using all search agents.

Generate the most effective solutions using the latest search agents

End for

Consider the most efficient solution.

End while

Return the best solution

Algorithme MOSOAVMP

La première phase de tous les algorithmes métaheuristiques est la phase aléatoire appelée initialisation, au cours de laquelle des solutions aléatoires sont générées dans l'espace de recherche. Cette phase influence considérablement l'efficacité de l'algorithme et les résultats finaux. Pour trouver une solution optimale, un nombre spécifique d'agents est défini. Dans cette version de l'algorithme SOA proposé, les différentes étapes sont les suivantes :

Étape 1 : Il s'agit de l'étape cruciale d'initialisation des paramètres pris en compte dans cet algorithme. Elle consiste à faire correspondre le VMP du CDC à la mouette et à réinitialiser la

position du VM. Le nombre maximum d'itérations, le nombre d'agents et les dimensions de l'espace sont pris en compte lors de l'initialisation.

Étape 2 : Il s'agit de la première phase de la recherche de la solution optimale. Cette solution est obtenue en fonction des contraintes de ressources, à l'aide de l'équation représentant la fonction objective. La solution ayant la plus petite valeur est alors considérée comme la solution optimale;

Étape 3 : Cette étape consiste à mettre à jour la position de la mouette ;

Étape 4 : Si le nombre maximum d'itérations est atteint, le processus de recherche est interrompu. C'est la fin du processus et la position de l'agent supérieur est prise en compte. Dans le cas contraire, on revient à l'étape 2.

VI.6. Résultats et discussions

Cette section présente la configuration utilisée, les différentes mesures de performance, ainsi que les différents résultats de simulation du nouvel algorithme (MOSOAVMP), évalués et comparés à divers algorithmes existants traitant du problème VMP tels que DMOSCA-SSA MOILP, MBO-VM, PIAS, MGGAVP, et MBFD. L'objectif principal est d'effectuer le placement optimal des VM sur les PM dans un environnement Cloud afin de limiter le gaspillage des ressources, les coûts de consommation d'énergie, le SLA, et de maximiser la performance totale. Dans cette étude, nous avons pris en compte plusieurs mesures d'évaluation clés telles que le gaspillage des ressources, la consommation d'énergie et l'utilisation des différentes ressources (CPU, mémoire, stockage et bande passante). À cela s'ajoutent le nombre de machines actives et la migration.

L'étude expérimentale de l'algorithme MOSOAVMP a été réalisée à l'aide du simulateur CloudSim version 5.0[238]. Le simulateur CloudSim prend en charge la modélisation et la simulation d'un DC basé sur le cloud[237]. Il a été sélectionné en raison de sa grande flexibilité, permettant l'implémentation de diverses fonctionnalités IaaS comme la gestion de l'énergie et des ressources. De plus, il offre une excellente plateforme pour tester et évaluer de nouvelles applications avant leur déploiement dans un environnement réel. Les expériences ont été réalisées sur un ordinateur équipé d'un CPU AMD Ryzen 5800U cadencé à 4,4 GHz avec 8 cœurs et des processeurs logiques égaux à 16, plus 16 Go de RAM à 3200 MHz, fonctionnant sous le système d'exploitation Windows 11.

Pour effectuer des simulations de l'environnement hétérogène du cloud, un seul DC est établi, comprenant 1800 PMs avec deux configurations, comme indiqué dans le Tableau 13. Ceci est fait

pour un ensemble allant de 200 à 2 000 VMs, avec un intervalle de 200 VMs pour chaque itération. Les spécifications des VMs sont détaillées dans le Tableau 13.

		Host name	**Parameter and value**
Host Type	**PMs**	HP ProLiant ML110 G3	MIPS: 3000 RAM: 4GB Cores: 2 Storage: 1024 GB Bandwidth: 3Gbps
		HP ProLiant G4	MIPS: 3720 RAM: 4GB Cores: 2 Storage: 1024 GB Bandwidth: 3Gbps
	VMs	-	MIPS: 2000 RAM: 1024MB Cores: 1 Storage: 2.5GB Bandwidth: 512MB

Tableau 13 : Differentes configurations pour testerr le MOSOAVMP

La Figure 56 montre les résultats de la consommation d'énergie pour les algorithmes utilisés dans les tests. La consommation d'énergie augmente avec le nombre de machines virtuelles actives dans le CDC. Dans tous les cas, MOSOAVMP présente une nette amélioration de la consommation d'énergie par rapport aux autres algorithmes comparés. Si l'on considère la configuration de base de 400 VM, le modèle MOSOAVMP proposé affiche une faible consommation d'énergie de 5,28 % en moyenne, par rapport aux algorithmes : DMOSCA-SSA, MOILP, PIAS et MGGAVP. Pour la configuration maximale de 200 machines virtuelles, MOSOAVMP présente une amélioration de 6,18 % par rapport aux autres modèles comparés. Dans tous les cas, l'algorithme proposé permet

une réduction globale de la consommation d'énergie de 5,44 % par rapport aux autres algorithmes comparés.

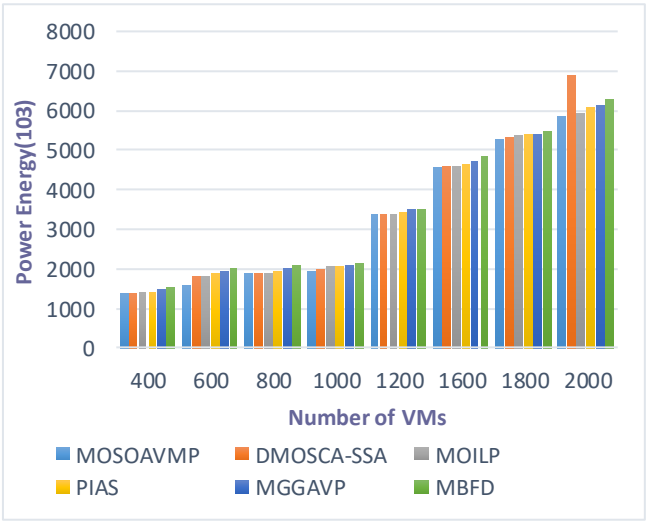

Figure 56 : Consommation d'énergie

La Figure 57 présente des comparaisons de migration de VM. La méthode proposée montre la performance et la stabilité des chiffres de migration. Pour migrer ces VM, nous prenons en compte les ressources disponibles sur les serveurs physiques. Dans le cas des algorithmes comparés, la corrélation des machines virtuelles et les changements soudains de ressources ne sont pas pris en compte, ce qui génère un grand nombre de machines virtuelles à migrer. L'algorithme proposé limite la migration des machines virtuelles tout en respectant les contraintes puisqu'il est basé sur la corrélation des ressources. Pour obtenir les meilleures performances, l'algorithme proposé minimise la concurrence qui peut survenir entre les machines virtuelles hébergées sur le même hôte en sélectionnant les machines virtuelles les plus appropriées.

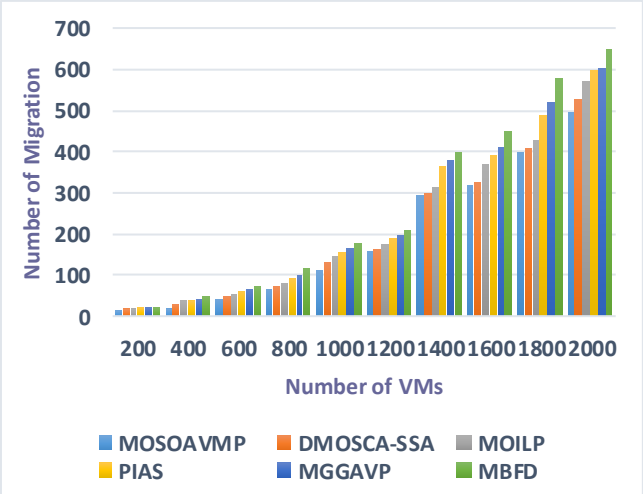

Figure 57 : Nombre de migrations

La Figure 57 montre l'efficacité de MOSOAVMP avec les coûts de migration les plus bas, suivi de DMOSCA-SSA, MOILP, PIAS, MGGAVP et MBFD. En général, la migration des machines virtuelles implique un volume considérable de transfert de données. Pour obtenir un VMP optimal, nous réduisons le nombre de migrations tout en plaçant les VM sur le minimum de PM possibles. De cette manière, l'algorithme proposé réduit le nombre de PM et de liens réseau actifs, tout en donnant la priorité au placement des VM. De cette manière, l'algorithme réduira le nombre de serveurs actifs tout en donnant la priorité au placement des machines virtuelles corrélées sur le même serveur ou sur des serveurs voisins.

La Fgure 58 montre les résultats pour le nombre de serveurs actifs en fonction du nombre de machines virtuelles hébergées. Les résultats des tests montrent que l'algorithme MOSOAVMP proposé nécessite moins d'hôtes physiques pour héberger les VM que les algorithmes comparés. Cette amélioration devient de plus en plus significative au fur et à mesure que le nombre de machines virtuelles qui lui sont assignées augmente. En moyenne, MOSOAVMP s'améliore de 7,92 % par rapport aux algorithmes comparés. Par exemple, comparons MOSOAVMP à DMOSCA-SSA, le deuxième algorithme le plus performant. Nous constatons une amélioration de 5,22 % lorsque nous utilisons une configuration de 200VM pour les deux algorithmes et une amélioration de 1,22 % dans le cas de 2000VM. Dans le cas de l'algorithme MBFD, qui présente les résultats les plus faibles, MOSOAVMP montre une nette amélioration de 19,92% pour la configuration de 200VMs et de 8,10% pour la configuration de 200%. Pour toutes les configurations de test, MOSOAVMP a obtenu de bons résultats par rapport à DMOSCA-SSA, MOILP, PIAS, MGGAVP et MBFD.

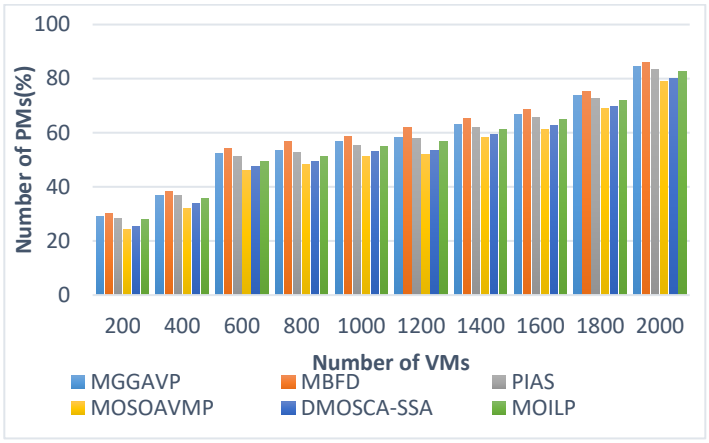

Figure 58 : Number of Active PMs

Les Figures 59,60,61 et 62 représentent respectivement l'utilisation moyenne des ressources PM telles que le CPU, la mémoire, le stockage et la bande passante en fonction des VM actives dans le DC. Les résultats démontrent l'efficacité de la méthode proposée dans toutes les situations. Pour l'utilisation du CPU, MOSOAVMP montre une amélioration moyenne de 14,84% par rapport aux algorithmes de pointe, suivi par DMOSCA-SSA. Pour la mémoire, MOSOAVMP montre une amélioration de 11,54%, une amélioration de 5,37% pour l'espace de stockage et une amélioration de 6,88% pour la bande passante.

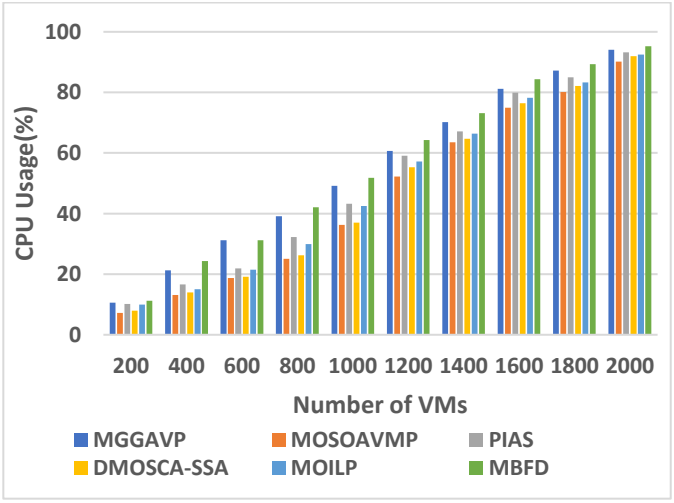

Figure 59 : Utilisation moyenne de la CPU

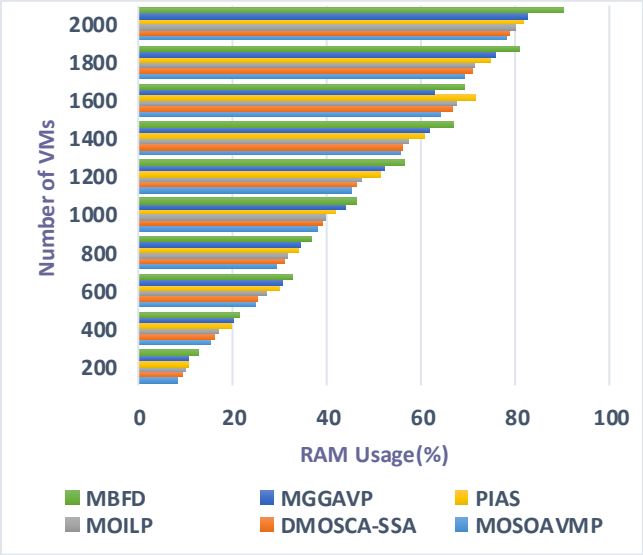

Figure 60 : Utilisation moyenne du RAM

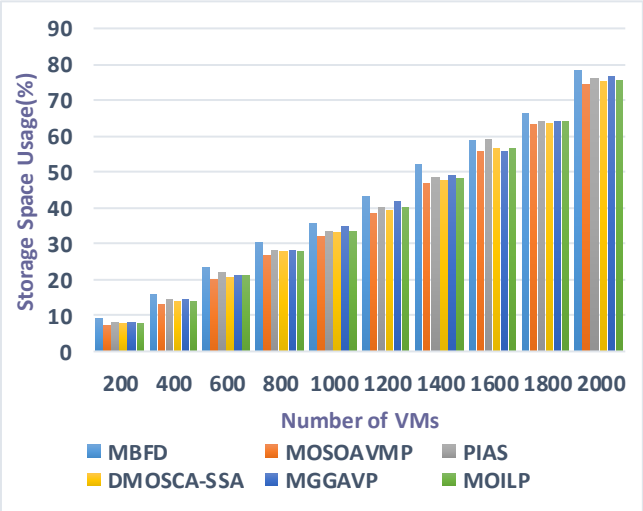

Figure 61 : Utilisation moyenne de stockage

Assurer une bonne gestion des ressources dans un cloud est l'une des principales tâches. Comme les autres mesures présentées ci-dessus, le gaspillage des ressources est très important pour les performances du cloud. D'après les résultats de la Figure 63, l'algorithme proposé minimise le gaspillage des ressources dans cloud computing par rapport aux algorithmes existants.

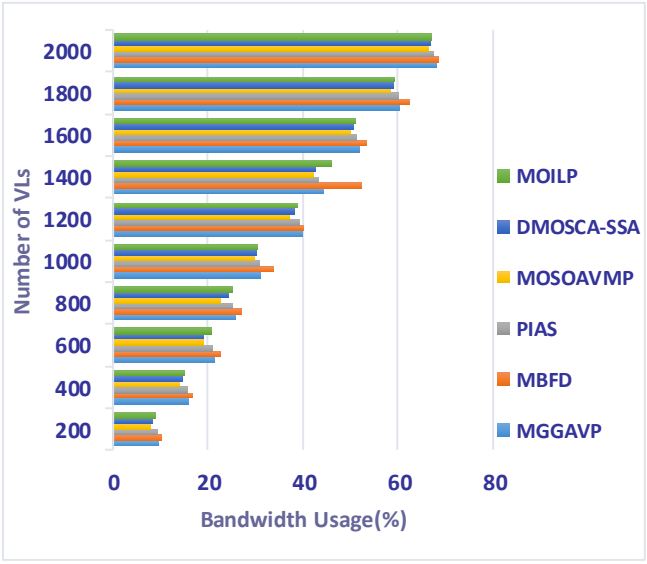

Figure 62 : Bande passante moyenne

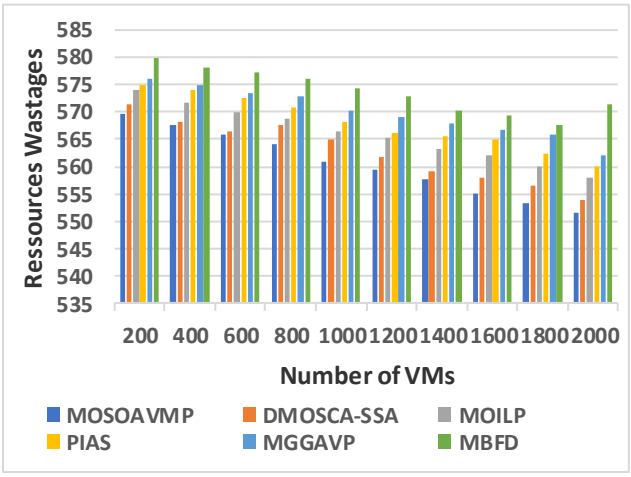

Figure 63 : Gaspillage de Ressources

La réduction du SLA se traduit par une expérience utilisateur plus fiable et la satisfaction du client. La Figure 64 montre l'efficacité de l'algorithme MOSOAVMP proposé pour la gestion des accords de niveau de service dans le cloud, par rapport à DMOSCA-SSA, MOILP, PIAS, MGGAVP et MBFD. Les diverses fluctuations observées précédemment avec les algorithmes existants ont été réduites. Comme le montre la Figure 64, les résultats obtenus démontrent la performance de MOSOAVMP par rapport aux algorithmes de l'état de l'arat comparés.

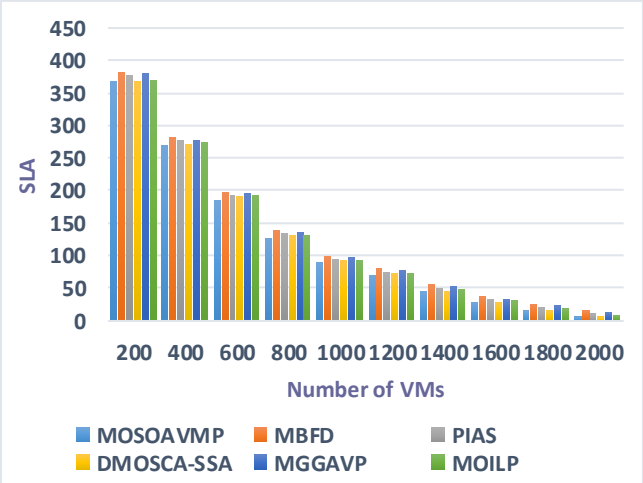

Figure 64 : SLA

Il convient de noter que pour les métaheuristiques, le temps de convergence pour trouver une solution optimale à différents problèmes est très important, c'est-à-dire le temps de convergence qui s'écoule pendant la phase d'exploration et d'exploitation. Pour examiner les performances de MOSOAVMP par rapport aux algorithmes de l'état de l'art, le nombre d'itérations est de 100. La Figure 65 montre les performances de la méthode proposée par rapport aux autres algorithmes. En tenant compte de l'augmentation du nombre d'itérations, l'aptitude des algorithmes DMOSCA-SSA, MOILP, PIAS, MGGAVP et MBFD diminue. Dans la même situation, l'algorithme MOSOAVMP devient très efficace. L'augmentation du nombre d'itérations permet à la méthode proposée de converger plus rapidement vers la solution optimale que les autres algorithmes. Comme toute métaheuristique, elle doit trouver un compromis entre la valeur de l'optimum local et la valeur de l'optimum global tout au long de la recherche de la meilleure solution. Cela détermine la capacité de MOSOAVMP à résoudre des problèmes complexes, en particulier les problèmes VMP.

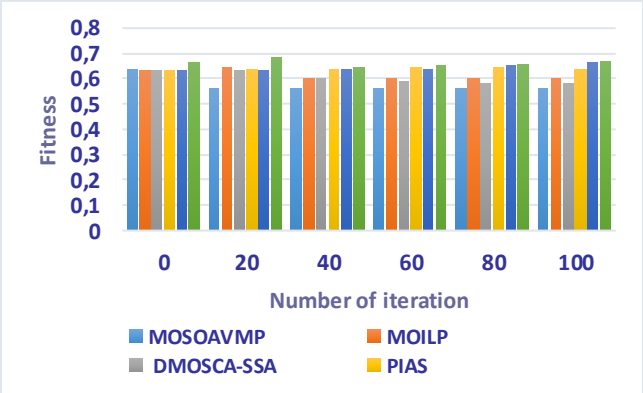

Figure 65 : Temps de convergence

La Figure 66 montre la comparaison entre le temps d'exécution de la méthode proposée et les méthodes existantes pour trouver une solution optimale pour le PMV. Pour effectuer cette comparaison, une plage de 100VMs à 500VMs est utilisée pour voir l'évolution du temps d'exécution. Si l'on se réfère à la figure 66, MOSOAVMP est plus performant que les autres algorithmes pour trouver la solution optimale du Placement des VMs à différentes tailles du problème.

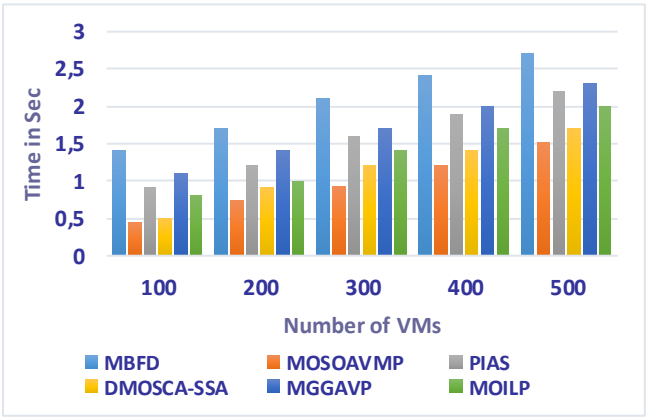

Figure 66 : Temps d'exécution

VI.7. Discussion et limitation

Les résultats indiquent que l'algorithme MOSOAVMP surpasse les autres principaux algorithmes en matière d'efficacité énergétique et de gestion des ressources dans les centres de données cloud. Comme le montre la Figure 56, la consommation d'énergie augmente proportionnellement au nombre de machines virtuelles (VM) actives. Cependant, MOSOAVMP réduit la consommation d'énergie de 5,44 % en moyenne par rapport à DMOSCA-SSA, MOILP, PIAS et MGGAVP, notamment avec une amélioration de 6,18 % pour 200 VM.

En termes de migrations de VMs (Figure 57), l'algorithme proposé démontre une stabilité supérieure en minimisant le nombre de migrations nécessaires. En prenant en compte la corrélation des ressources, MOSOAVMP optimise le placement des VM, réduisant le nombre de serveurs physiques actifs et les coûts associés. Par exemple, pour 2000 VM, MOSOAVMP améliore l'efficacité de 1,22% par rapport à DMOSCA-SSA et de 8,10% par rapport à MBFD.

Les figures 59 à 62 montrent que MOSOAVMP optimise l'utilisation des ressources PM telles que le CPU, la mémoire, le stockage et la bande passante, avec des améliorations respectives de 14,84 %, 11,54 %, 5,37 % et 6,88 % par rapport aux autres algorithmes. Cette gestion efficace des

ressources est essentielle pour réduire le gaspillage et améliorer les performances globales du cloud.

En ce qui concerne le respect des accords de niveau de service, MOSOAVMP fait preuve d'une efficacité accrue (Figure 64), réduisant les fluctuations observées avec les autres algorithmes et garantissant une expérience utilisateur plus fiable. Enfin, la figure 65 montre que MOSOAVMP converge plus rapidement vers une solution optimale en augmentant le nombre d'itérations.

VI.8. Conclusion

Les centres de données cloud sont constitués de diverses technologies gourmandes en énergie, notamment des serveurs physiques, des dispositifs de commutation et des dispositifs de refroidissement. Cependant, les dépenses des centres de données augmentent en raison de la consommation d'énergie de ces dispositifs qui peuvent éventuellement causer un problème majeur de chaleur. Dans ce chapitre, nous proposons un algorithme qui peut aider à optimiser le problème du VMP, dans le but de réduire la consommation d'énergie, d'améliorer l'utilisation des ressources et de minimiser les violations du SLA. Nous formulons ce problème comme un problème d'optimisation polyvalent et employons une approche basée sur les mouettes pour le résoudre. Les résultats des tests montrent que la solution proposée permet d'atteindre tous les objectifs fixés dans ce travail, par rapport aux algorithmes des travaux existants. MOSOAVMP a montré de bonnes performances d'exploitation et d'exploration. En termes de précision et de rapidité de recherche de la meilleure solution, MOSOAVMP a une meilleure vitesse de convergence.

Dans de futures recherches, nous prévoyons de comparer cette méthode avec de nouveaux algorithmes métaheuristiques tels que[275], [276], [277], [278], tout en prenant en compte des métriques supplémentaires telles que la sécurité du cloud et l'ordonnancement des tâches. En outre, nos objectifs comprennent l'exploration du placement des conteneurs dans le cloud à l'aide de métaheuristiques et de l'apprentissage profond. En outre, nous étudierons l'intégration de la théorie des jeux pour améliorer le processus d'optimisation.

CONCLUSION GENERALE

Conclusion

La problématique de l'optimisation des ressources virtuelles des datacenters basés cloud spécialement l'efficacité énergétique représente un domaine de recherche d'une importance capitale, compte tenu des enjeux fondamentaux qui y sont associés. Tout d'abord, l'essor sans précédent des datacenters à l'échelle mondiale, répondant à une demande croissante de services pour une population d'utilisateurs en croissance constante, met en lumière une réalité incontournable. En effet, cette expansion incessante se traduit par une augmentation continue des capacités de traitement des serveurs et des volumes de stockage, engendrant ainsi une préoccupation majeure quant à la consommation colossale d'énergie électrique requise pour les alimenter de manière adéquate.

De surcroît, étant donné que l'énergie électrique représente une ressource vitale pour le domaine de l'informatique, il est opportun d'explorer des moyens d'optimisation des réserves actuelles de l'humanité, face aux études prospectives alarmantes quant à leur épuisement à moyen ou long terme. Ainsi, de nombreuses recherches se sont penchées sur cette thématique, conduisant à l'élaboration de plusieurs techniques. D'une part, on trouve les techniques réactives, qui englobent toutes les stratégies visant à surveiller les taux d'utilisation des ressources des serveurs et à intervenir dès lors qu'ils dépassent les seuils d'efficacité énergétique préétablis. D'autre part, les méthodes proactives, qui scrutent les tendances de consommation électrique des serveurs et ont la capacité d'anticiper les éventuels débordements, prenant des mesures préventives avant qu'ils ne surviennent.

Dans ces deux cas, outre les considérations relatives à la facture énergétique, une attention particulière est portée au respect des "Service Level Agreements (SLA)". Ces techniques débouchent généralement sur des décisions telles que la migration des serveurs virtuels, visant à garantir une disponibilité optimale des ressources, ou encore l'extinction des serveurs sous-utilisés afin de réduire le gaspillage d'énergie qu'ils génèrent au sein du Datacenter. L'objectif primordial de ces actions est de parvenir à préserver chaque watt d'énergie fourni aux serveurs, qui ne contribue pas réellement au traitement effectif d'une tâche donnée. Cette démarche, à l'échelle des datacenters de grande envergure, pourrait entraîner une diminution significative de leur consommation énergétique globale.

Dans cette thèse, nous avons exploré l'importance cruciale de la gestion efficace des ressources dans les environnements de cloud computing. En particulier, nous nous sommes concentrés sur le placement optimal des machines virtuelles (VMP) dans les centres de données cloud, une tâche complexe qui influe directement sur des aspects essentiels tels que l'allocation des ressources, l'équilibrage des charges, la consommation d'énergie et la qualité de service. Notre contribution majeure réside dans la proposition et la mise en œuvre d'un algorithme métaheuristique novateur, le Grey Wolf Optimization for Virtual Machine Placement (GWOVMP), qui vise à optimiser le placement des machines virtuelles dans un environnement cloud. Notre étude a démontré l'efficacité de notre approche GWOVMP par le biais d'expérimentations rigoureuses. Les résultats obtenus ont révélé une réduction significative de la consommation d'énergie de l'ordre de 20,99 %, ainsi qu'une diminution du gaspillage des ressources de 1,80 % par rapport aux algorithmes existants. Ces résultats confirment l'utilité et l'efficacité de notre méthode pour répondre aux défis de l'optimisation du VMP dans les environnements cloud.

Il est important de souligner que notre recherche s'inscrit dans un contexte où la réduction de la consommation d'énergie et l'optimisation des ressources sont des préoccupations majeures dans les centres de données cloud. En proposant une solution optimale ou quasi-optimale pour le problème de placement des machines virtuelles, notre travail contribue à la promotion d'une utilisation plus efficace et durable des ressources dans ces environnements informatiques essentiels à notre société. Cette étude apporte une contribution à la recherche sur l'optimisation du cloud computing, en particulier en ce qui concerne le placement des machines virtuelles. En adoptant une approche métaheuristique innovante, notre méthode GWOVMP offre une solution prometteuse pour relever les défis complexes et évolutifs de la gestion des ressources dans les environnements de cloud computing.

Dans cette contribution de thèse, nous avons encore une fois abordé le défi crucial du placement optimal des machines virtuelles (VMP) dans les centres de données cloud (CDC), en reconnaissant les limites des méthodes actuelles qui ne prennent en compte qu'un nombre restreint de ressources. Notre contribution principale réside dans la proposition d'une nouvelle approche appelée Machine Virtuelle à Algorithme d'Optimisation Multi-Objectif Seagull (MOSOAVMP), qui vise à résoudre ces lacunes et à améliorer la gestion des ressources dans les CDC tout en introduisant des métriques qui n’avaient pas étatient abordées dans la pécédente contribution.

Notre étude a démontré l'efficacité de l'approche MOSOAVMP par le biais d'expérimentations approfondies. Les résultats obtenus ont révélé une réduction significative des gaspillages de ressources et de la consommation d'énergie, ainsi qu'une amélioration notable de l'utilisation des ressources et une minimisation des violations des accords de niveau de service. En particulier, MOSOAVMP a permis de réduire la consommation d'énergie de 5,44 % tout en améliorant l'utilisation moyenne du CPU, de la mémoire, du stockage et de la bande passante par rapport aux algorithmes de l'état de l'art qui ont été considérés. Il est important de souligner que notre recherche s'inscrit dans un contexte où les coûts d'exploitation des CDC augmentent en raison de la consommation d'énergie croissante des technologies gourmandes en énergie. En proposant une solution efficace et polyvalente pour optimiser le placement des machines virtuelles, notre travail contribue à atténuer ces défis tout en répondant aux exigences croissantes de performance et de disponibilité des services cloud. Notre étude offre une contribution significative à la recherche sur l'optimisation du cloud computing, en particulier en ce qui concerne le placement des machines virtuelles.

Etant donné que l'intégration croissante des appareils IoT a engendré une explosion de données nécessitant des technologies avancées comme le cloud computing pour leur gestion efficace. Cette thèse a démontré l'importance cruciale des systèmes de recommandation pour traiter et analyser ces données volumineuses, offrant des suggestions pertinentes basées sur les préférences et les besoins des utilisateurs. Nous avons particulièrement exploré l'application des systèmes de recommandation dans le contexte des élections, où ils peuvent améliorer la rapidité et l'efficacité des procédures de vote.

Nos recherches ont inclus une étude comparative exhaustive des modèles de recommandation appliqués aux données électorales, parmi lesquels des algorithmes tels que K-NN, Rocchio, l'analyse sémantique latente, Naïve Bayes, les classificateurs d'arbres décisionnels, et les réseaux de neurones artificiels. Cette évaluation, basée sur des métriques de performance telles que la précision, le rappel, le F1-score, le RMSE, le MSE, le MAE et l'aire sous la courbe ROC, a révélé que le modèle K-NN et SGDC GridSearchCV se distinguent par leurs performances supérieures. En outre, nous avons proposé un système de recommandation hybride combinant le filtrage basé sur le contenu et le filtrage collaboratif, qui optimise les recommandations en exploitant les similitudes entre les utilisateurs et les éléments. Cette approche hybride, en tirant parti des forces de chaque technique, offre des recommandations plus précises et personnalisées. Notre travail souligne la synergie entre les systèmes de recommandation, le cloud computing et l'IoT, et

démontre comment leur interconnexion peut révolutionner des domaines critiques comme les élections. Nos résultats mettent en lumière non seulement l'efficacité des algorithmes spécifiques comme K-NN dans des environnements Big Data, mais aussi l'importance de combiner diverses méthodes pour des performances optimales. Cette thèse ouvre ainsi la voie à des applications futures où la technologie peut continuer à transformer et à sécuriser des processus décisionnels et opérationnels dans divers secteurs.

Dans ce contexte, nous avons constaté que la sécurité des données et la disponibilité permanente des informations demeurent des défis majeurs pour les systèmes d'information intégrant l'IoT et le cloud computing. La technologie blockchain se présente comme une solution prometteuse pour répondre aux problématiques de sécurité actuelles. Notre thèse propose une nouvelle architecture basée sur la blockchain pour garantir la sécurité des environnements IoT et cloud. Cette contribution inclut une revue exhaustive de la littérature concernant l'application de la blockchain à la sécurité de l'IoT, illustrant ainsi les avantages et les défis associés à cette approche.

Cette recherche apporte une contribution en offrant une solution pour la sécurisation des infrastructures IoT et cloud, mettant en évidence l'efficacité de la blockchain pour améliorer la résilience et la fiabilité des systèmes d'information modernes. La proposition d'une architecture intégrée exploitant la blockchain pour renforcer la sécurité des données et garantir leur disponibilité permanente représente une avancée majeure pour le développement de solutions technologiques durables et sécurisées.

Perspectives des Travaux à Venir

L'essor des datacenters à l'échelle mondiale, répondant à une demande croissante de services, met en lumière la nécessité de développer des solutions innovantes pour gérer l'augmentation continue des capacités de traitement et des volumes de stockage tout en minimisant la consommation énergétique. Les travaux futurs devraient se concentrer sur plusieurs axes principaux.

Premièrement, l'exploration de nouvelles méthodes d'optimisation proactive des ressources, intégrant des techniques avancées d'apprentissage machine et d'intelligence artificielle, pourrait permettre d'anticiper les besoins en ressources et d'ajuster les configurations des datacenters en temps réel pour maximiser l'efficacité énergétique.

Deuxièmement, l'intégration de sources d'énergie renouvelable et la gestion intelligente des flux énergétiques au sein des datacenters constituent une piste prometteuse. L'objectif serait de développer des algorithmes capables de coordonner l'utilisation des énergies renouvelables avec les besoins des serveurs, réduisant ainsi la dépendance aux sources d'énergie traditionnelles et leur impact environnemental.

Troisièmement, l'accent devrait être mis sur l'amélioration des techniques de migration des machines virtuelles (VM), en minimisant les interruptions de service et les coûts associés. Des solutions hybrides combinant des approches réactives et proactives pourraient offrir des performances optimales.

Enfin, la mise en œuvre de frameworks de simulation et de modélisation robustes permettant de tester et de valider les nouvelles approches dans des environnements contrôlés avant leur déploiement à grande échelle est essentielle. Cela permettrait de garantir la fiabilité et l'efficacité des solutions proposées, tout en réduisant les risques opérationnels. En somme, la poursuite de ces axes de recherche est essentielle pour répondre aux défis énergétiques et opérationnels des datacenters modernes, contribuant ainsi à un cloud plus durable et efficiente.

REFERENCES BIBLIOGRAPHIQUES

[1] D. Villegas *et al.*, “Cloud federation in a layered service model,” in *Journal of Computer and System Sciences*, Academic Press Inc., 2012, pp. 1330–1344. doi: 10.1016/j.jcss.2011.12.017.

[2] A. Taherkordi, F. Zahid, Y. Verginadis, and G. Horn, “Future Cloud Systems Design: Challenges and Research Directions,” *IEEE Access*, vol. 6, pp. 74120–74150, 2018, doi: 10.1109/ACCESS.2018.2883149.

[3] M. Amini and N. J. Javid, “A Multi-Perspective Framework Established on Diffusion of Innovation (DOI) Theory and Technology, Organization and Environment (TOE) Framework Toward Supply Chain Management System Based on Cloud Computing Technology for Small and Medium Enterprises.” [Online]. Available: https://ssrn.com/abstract=4340207

[4] P. J. Sun, “Security and privacy protection in cloud computing: Discussions and challenges,” *Journal of Network and Computer Applications*, vol. 160. Academic Press, Jun. 15, 2020. doi: 10.1016/j.jnca.2020.102642.

[5] A. Rashid and A. Chaturvedi, “Cloud Computing Characteristics and Services A Brief Review,” *International Journal of Computer Sciences and Engineering*, vol. 7, no. 2, pp. 421–426, Feb. 2019, doi: 10.26438/ijcse/v7i2.421426.

[6] “Option : Informatique Titre Une approche basée agent pour la sécurité dans le Cloud Computing.”

[7] C. Gong, J. Liu, Q. Zhang, H. Chen, and Z. Gong, “The characteristics of cloud computing,” in *Proceedings of the International Conference on Parallel Processing Workshops*, 2010, pp. 275–279. doi: 10.1109/ICPPW.2010.45.

[8] A. Harbaoui, “Vers une modélisation et un dimensionnement automatiques des applications répar- ties,” 2011.

[9] A. Jula, E. Sundararajan, and Z. Othman, “Cloud computing service composition: A systematic literature review,” *Expert Systems with Applications*, vol. 41, no. 8. Elsevier Ltd, pp. 3809–3824, Jun. 15, 2014. doi: 10.1016/j.eswa.2013.12.017.

[10] I. Foster, Y. Zhao, I. Raicu, and S. Lu, “Cloud Computing and Grid Computing 360-Degree Compared.” [Online]. Available: http://www.us-vo.org

[11] D. T. Gandecha, “A Review on Cloud Computing Technology, Cloud Deployment and Service Delivery Models,” *Int J Res Appl Sci Eng Technol*, vol. 10, no. 1, pp. 1238–1245, Jan. 2022, doi: 10.22214/ijraset.2022.40022.

[12] X. Xu, “From cloud computing to cloud manufacturing,” *Robot Comput Integr Manuf*, vol. 28, no. 1, pp. 75–86, Feb. 2012, doi: 10.1016/j.rcim.2011.07.002.

[13] M. M. Alani, “Securing the Cloud: Threats, Attacks and Mitigation Techniques,” *Journal of Advanced Computer Science & Technology*, vol. 3, no. 2, pp. 202–213, Oct. 2014, doi: 10.14419/jacst.v3i2.3588.

[14] N. J. Taniwall, S. Hassan, H. Sulaimanzai, M. S. Haider, and S. K. Salih, “Incentives in Adopting Cloud Computing for Shaikh Zayed University, Khost Afghanistan,” *Journal for Research in Applied Sciences and Biotechnology*, vol. 2, no. 3, pp. 7–11, Jun. 2023, doi: 10.55544/jrasb.2.3.2.

[15] Y. Rao Bhandayker, “A Study on the Research Challenges and Trends of Cloud Computing,” 2019. [Online]. Available: www.rrjournals.com

[16] J. Srinivas, K. Venkata, S. Reddy, and A. M. Qyser, “CLOUD COMPUTING BASICS,” 2012. [Online]. Available: www.ijarcce.com

[17] U. A. Butt *et al.*, “A review of machine learning algorithms for cloud computing security,” *Electronics (Switzerland)*, vol. 9, no. 9. MDPI AG, pp. 1–25, Sep. 01, 2020. doi: 10.3390/electronics9091379.

[18] B. Chandrakala, “A Review on Chronicle of Cloud Computing Security and Storage Environment Models.”

[19] S. Bhonsle, N. Deshpande, S. More, and R. Patil, “Microsoft Azure vs. Amazon Cloud Services: A Comparative Analysis,” 2023.

[20] M. Ravend and K. Upadhyay, “REASEARCH IN CLOUD COMPUTING SECURITY,” *www.irjmets.com @International Research Journal of Modernization in Engineering*, vol. 395, [Online]. Available: www.irjmets.com

[21] M. De Donno, A. Giaretta, N. Dragoni, A. Bucchiarone, and M. Mazzara, “Cyber-storms come from clouds: Security of cloud computing in the IoT era,” *Future Internet*, vol. 11, no. 6, Jun. 2019, doi: 10.3390/fi11060127.

[22] M. Sakib and B. Alam, “Cloud Computing-Architecture, Platform and Security Issues: A Survey”, [Online]. Available: www.worldscientificnews.com

[23] Institute of Electrical and Electronics Engineers, *2019 IEEE 4th International Conference on Computer and Communication Systems : ICCCS 2019 : February 23-25, 2019, Singapore*.

[24] A. Ndayikengurukiye, A. Ez-Zahout, and F. Omary, “An overview of the different methods for optimizing the virtual resources placement in the Cloud Computing,” in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Jan. 2021. doi: 10.1088/1742-6596/1743/1/012030.

[25] A. Rashid and A. Chaturvedi, “Virtualization and its Role in Cloud Computing Environment,” *International Journal of Computer Sciences and Engineering*, vol. 7, no. 4, pp. 1131–1136, Apr. 2019, doi: 10.26438/ijcse/v7i4.11311136.

[26] C. Pahl, A. Brogi, J. Soldani, and P. Jamshidi, “Cloud container technologies: A state-of-the-art review,” *IEEE Transactions on Cloud Computing*, vol. 7, no. 3, pp. 677–692, Jul. 2019, doi: 10.1109/TCC.2017.2702586.

[27] S. D. A. Shah, M. A. Gregory, and S. Li, “Cloud-Native Network Slicing Using Software Defined Networking Based Multi-Access Edge Computing: A Survey,” *IEEE Access*, vol. 9. Institute of Electrical and Electronics Engineers Inc., pp. 10903–10924, 2021. doi: 10.1109/ACCESS.2021.3050155.

[28] C. Pahl, “Containerization and the PaaS Cloud.”

[29] A. Bhardwaj and C. R. Krishna, “Virtualization in Cloud Computing: Moving from Hypervisor to Containerization—A Survey,” *Arab J Sci Eng*, vol. 46, no. 9, pp. 8585–8601, Sep. 2021, doi: 10.1007/s13369-021-05553-3.

[30] A. Brogi *et al.*, “SeaClouds: An Open Reference Architecture for Multi-Cloud Governance.” [Online]. Available: https://jclouds.apache.org

[31] O. Bentaleb, A. S. Z. Belloum, A. Sebaa, and A. El-Maouhab, “Containerization technologies: taxonomies, applications and challenges,” *Journal of Supercomputing*, vol. 78, no. 1, pp. 1144–1181, Jan. 2022, doi: 10.1007/s11227-021-03914-1.

[32] I. Wardhana, M. Ariawijaya, R. Hasnur, R. Syafitri, and A. Nasuha, “Design and analysis security architecture virtualization OpenVz,” in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Jun. 2021. doi: 10.1088/1742-6596/1940/1/012088.

[33] A. M. Potdar, D. G. Narayan, S. Kengond, and M. M. Mulla, “Performance Evaluation of Docker Container and Virtual Machine,” in *Procedia Computer Science*, Elsevier B.V., 2020, pp. 1419–1428. doi: 10.1016/j.procs.2020.04.152.

[34] D. Godlove, “Singularity,” in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Jul. 2019. doi: 10.1145/3332186.3332192.

[35] G. Cavallaro, V. Kozlov, M. Götz, and M. Riedel, “REMOTE SENSING DATA ANALYTICS WITH THE UDOCKER CONTAINER TOOL USING MULTI-GPU DEEP LEARNING SYSTEMS”, doi: 10.2760/848593.

[36] M. K. Abhishek, D. R. Rao, and K. Subrahmanyam, “Framework for Containers Orchestration to handle the Scientific Workloads using Kubernetes,” *Journal of Computer Science*, vol. 18, no. 9, pp. 860–867, 2022, doi: 10.3844/jcssp.2022.860.867.

[37] “CLOUD DATA SECURITY METHODS: KUBERNETES VS DOCKER SWARM,” *International Research Journal of Modernization in Engineering Technology and Science*, Dec. 2022, doi: 10.56726/irjmets32176.

[38] Q. Li and C. Zhang, “Continual learning on deployment pipelines for Machine Learning Systems,” Dec. 2022, [Online]. Available: http://arxiv.org/abs/2212.02659

[39] M. C. Huebscher and J. A. McCann, “A survey of Autonomic Computing - Degrees, models, and applications,” *ACM Comput Surv*, vol. 40, no. 3, Aug. 2008, doi: 10.1145/1380584.1380585.

[40] I. Elgendi, M. F. Hossain, A. Jamalipour, and K. S. Munasinghe, “Protecting Cyber Physical Systems Using a Learned MAPE-K Model,” *IEEE Access*, vol. 7, pp. 90954–90963, 2019, doi: 10.1109/ACCESS.2019.2927037.

[41] P. Bourret, “Modèle à Composant pour Plate-forme Autonomique.”

[42] S. H. Haji *et al.*, “Comparison of Software Defined Networking with Traditional Networking,” *Asian Journal of Research in Computer Science*, pp. 1–18, May 2021, doi: 10.9734/ajrcos/2021/v9i230216.

[43] M. Panahi, “A Survey: Enhancing the Security of (IOT) Internet of Things by utilizing (SDN) Software-Defined Networking”, doi: 10.13140/RG.2.2.16048.02564.

[44] X. Cao, Y. Li, X. Xiong, and J. Wang, “Dynamic Routings in Satellite Networks: An Overview,” *Sensors*, vol. 22, no. 12. MDPI, Jun. 01, 2022. doi: 10.3390/s22124552.

[45] J. Lorincz, Z. Klarin, and D. Begusic, “Advances in Improving Energy Efficiency of Fiber–Wireless Access Networks: A Comprehensive Overview,” *Sensors*, vol. 23, no. 4. MDPI, Feb. 01, 2023. doi: 10.3390/s23042239.

[46] M. H. Rehmani, A. Davy, B. Jennings, and C. Assi, “Software Defined Networks-Based Smart Grid Communication: A Comprehensive Survey,” *IEEE Communications Surveys and Tutorials*, vol. 21, no. 3, pp. 2637–2670, Jul. 2019, doi: 10.1109/COMST.2019.2908266.

[47] W. Hassan, T.-S. Chou, X. Li, P. Appiah-Kubi, and O. Tamer, “Latest trends, challenges and Solutions in Security in the era of Cloud Computing and Software Defined Networks,” *International Journal of Informatics and Communication Technology (IJ-ICT)*, vol. 8, no. 3, p. 162, 2019, doi: 10.11591/ijict.v8i3.pp162-183.

[48] V. A. Pandey, “Issue 10 IJSDR2210181 www,” *ijsdr.org International Journal of Scientific Development and Research*, vol. 7, 2022, [Online]. Available: www.ijsdr.org

[49] H. Xu, H. Huang, S. Chen, G. Zhao, and L. Huang, “Achieving High Scalability Through Hybrid Switching in Software-Defined Networking,” *IEEE/ACM Transactions on Networking*, vol. 26, no. 1, pp. 618–632, Feb. 2018, doi: 10.1109/TNET.2018.2789339.

[50] M. Karakus and A. Durresi, “Service cost in software defined networking (SDN),” in *Proceedings - International Conference on Advanced Information Networking and Applications, AINA*, Institute of Electrical and Electronics Engineers Inc., May 2017, pp. 468–475. doi: 10.1109/AINA.2017.111.

[51] B. Yi, X. Wang, K. Li, S. k. Das, and M. Huang, “A comprehensive survey of Network Function Virtualization,” *Computer Networks*, vol. 133. Elsevier B.V., pp. 212–262, Mar. 14, 2018. doi: 10.1016/j.comnet.2018.01.021.

[52] S. Yang, F. Li, S. Trajanovski, R. Yahyapour, and X. Fu, “Recent Advances of Resource Allocation in Network Function Virtualization,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 2, pp. 295–314, Feb. 2021, doi: 10.1109/TPDS.2020.3017001.

[53] A. Arulappan, G. Raja, K. Passi, and A. Mahanti, “Optimization of 5G/6G Telecommunication Infrastructure through an NFV-Based Element Management System,” *Symmetry (Basel)*, vol. 14, no. 5, May 2022, doi: 10.3390/sym14050978.

[54] M. Yoshida, “A Multiple Objective Resource Scheduling Algorithm for Network Services on the NFV Infrastructure,” *Journal of Information Processing*, vol. 31, pp. 196–204, 2023, doi: 10.2197/ipsjjip.31.196.

[55] R. Mijumbi, J. Serrat, J. L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, “Network function virtualization: State-of-the-art and research challenges,” *IEEE Communications Surveys and Tutorials*, vol. 18, no. 1, pp. 236–262, Jan. 2016, doi: 10.1109/COMST.2015.2477041.

[56] I. Farris, T. Taleb, Y. Khettab, and J. Song, “A survey on emerging SDN and NFV security mechanisms for IoT systems,” *IEEE Communications Surveys and Tutorials*, vol. 21, no. 1, pp. 812–837, Jan. 2019, doi: 10.1109/COMST.2018.2862350.

[57] K. Nassiri *et al.*, “L ’ utilisation de la blockchain pour la sécurité de l ’ internet des objets . To cite this version : HAL Id : hal-02296372,” 2019.

[58] S. Kirkman, “A data movement policy framework for improving trust in the cloud using smart contracts and blockchains,” *Proceedings - 2018 IEEE International Conference on Cloud Engineering, IC2E 2018*, pp. 270–273, 2018, doi: 10.1109/IC2E.2018.00054.

[59] N. N. Misra, Y. Dixit, A. Al-Mallahi, M. S. Bhullar, R. Upadhyay, and A. Martynenko, “IoT, Big Data, and Artificial Intelligence in Agriculture and Food Industry,” *IEEE Internet Things J*, vol. 9, no. 9, pp. 6305–6324, May 2022, doi: 10.1109/JIOT.2020.2998584.

[60] S. Li, L. Da Xu, and S. Zhao, “The internet of things: a survey,” *Information Systems Frontiers*, vol. 17, no. 2, pp. 243–259, 2015, doi: 10.1007/s10796-014-9492-7.

[61] M. S. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli, and M. H. Rehmani, “Applications of Blockchains in the Internet of Things: A Comprehensive Survey,” *IEEE Communications Surveys and Tutorials*, vol. 21, no. 2, pp. 1676–1717, 2019, doi: 10.1109/COMST.2018.2886932.

[62] H. F. Atlam, A. Alenezi, M. O. Alassafi, and G. B. Wills, “Blockchain with Internet of Things: Benefits, challenges, and future directions,” *International Journal of Intelligent Systems and Applications*, vol. 10, no. 6, pp. 40–48, 2018, doi: 10.5815/ijisa.2018.06.05.

[63] Y. Yuan and F. Y. Wang, “Blockchain and Cryptocurrencies: Model, Techniques, and Applications,” *IEEE Trans Syst Man Cybern Syst*, vol. 48, no. 9, pp. 1421–1428, 2018, doi: 10.1109/TSMC.2018.2854904.

[64] R. M. Frey, R. Xu, and A. Ilic, “A Novel Recommender System in IoT,” *The 5th International Conference on the Internet of Things*, no. December, pp. 4–6, 2015.

[65] J. Deogirikar and A. Vidhate, “Security attacks in IoT: A survey,” *Proceedings of the International Conference on IoT in Social, Mobile, Analytics and Cloud, I-SMAC 2017*, pp. 32–37, 2017, doi: 10.1109/I-SMAC.2017.8058363.

[66] A. Antwiwaa, “A VIEW OF A SMARTER WORLD WITH INTERNET OF THINGS: A SURVEY,” 2017. [Online]. Available: https://www.researchgate.net/publication/318318424

[67] C. Jaag and C. Bach, “Blockchain Technology and Cryptocurrencies: Opportunities for Postal Financial Services,” *The Changing Postal and Delivery Sector*, vol. 41, no. 0, pp. 205–221, 2017, doi: 10.1007/978-3-319-46046-8_13.

[68] C. Noyes, “Blockchain Multiparty Computation Markets at Scale,” *Ieee Transactions on Xxxxx*, vol. 0, no. 0, pp. 1–4, [Online]. Available: https://pdfs.semanticscholar.org/fb7b/e4b7fd591da13380b350a19b2a3ed20a0323.pdf

[69] H. Wang, Z. Zheng, S. Xie, H. N. Dai, and X. Chen, “Blockchain challenges and opportunities: a survey,” *International Journal of Web and Grid Services*, vol. 14, no. 4, p. 352, 2018, doi: 10.1504/ijwgs.2018.10016848.

[70] N. Gogerty and J. Zitoli, “DeKo: An Electricity-Backed Currency Proposal,” *SSRN Electronic Journal*, no. January 2011, 2012, doi: 10.2139/ssrn.1802166.

[71] D. Peter, “Appropriated To Enhance Online Learning ?,” *The Open University’s repository of research publications and other research outputs*, pp. 1–6, 2015, [Online]. Available: http://oro.open.ac.uk/44966/1/Devine2015-altc-blockchainlearning.pdf

[72] R. Hackathorn, “Adaptive and Agile,” *DM Review*, vol. 2, no. September 2016, p. 700, 2002, doi: 10.1007/978-3-319-45153-4.

[73] R. Dennis and G. Owen, “Rep on the block: A next generation reputation system based on the blockchain,” *2015 10th International Conference for Internet Technology and Secured Transactions, ICITST 2015*, pp. 131–138, 2016, doi: 10.1109/ICITST.2015.7412073.

[74] N. Komninos, E. Philippou, and A. Pitsillides, “Survey in smart grid and smart home security: Issues, challenges and countermeasures,” *IEEE Communications Surveys and Tutorials*, vol. 16, no. 4, pp. 1933–1954, 2014, doi: 10.1109/COMST.2014.2320093.

[75] J. Wurm, K. Hoang, O. Arias, A. R. Sadeghi, and Y. Jin, “Security analysis on consumer and industrial IoT devices,” *Proceedings of the Asia and South Pacific Design Automation Conference, ASP-DAC*, vol. 25-28-Janu, pp. 519–524, 2016, doi: 10.1109/ASPDAC.2016.7428064.

[76] C. Sillaber and B. Waltl, “Life Cycle of Smart Contracts in Blockchain Ecosystems,” *Datenschutz und Datensicherheit - DuD*, vol. 41, no. 8, pp. 497–500, 2017, doi: 10.1007/s11623-017-0819-7.

[77] Y. Yuan and F. Y. Wang, “Towards blockchain-based intelligent transportation systems,” *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, no. November, pp. 2663–2668, 2016, doi: 10.1109/ITSC.2016.7795984.

[78] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten, “SoK: Research perspectives and challenges for bitcoin and cryptocurrencies,” *Proc IEEE Symp Secur Priv*, vol. 2015-July, pp. 104–121, 2015, doi: 10.1109/SP.2015.14.

[79] S. Islam *et al.*, “Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000. A Survey on Consensus Algorithms in Blockchain-based Applications: Architecture, Taxonomy, and Operational Issues”, doi: 10.1109/ACCESS.2017.DOI.

[80] N. Chowdhury, M. Ramachandran, A. Third, A. Mikroyannidis, M. Bachler, and J. Domingue, “Open Research Online,” 2020.

[81] K. K. Vaigandla, M. Siluveru, M. Kesoju, and R. Karne, “Review on Blockchain Technology: Architecture, Characteristics, Benefits, Algorithms, Challenges and Applications,” *Mesopotamian Journal of CyberSecurity*, vol. 2023. Mesopotamian Academic Press, pp. 73–84, Dec. 10, 2023. doi: 10.58496/MJCS/2023/012.

[82] H. Lu, K. Huang, M. Azimi, and L. Guo, “Blockchain technology in the oil and gas industry: A review of applications, opportunities, challenges, and risks,” *IEEE Access*, vol. 7, pp. 41426–41444, 2019, doi: 10.1109/ACCESS.2019.2907695.

[83] Z. Zheng *et al.*, “An overview on smart contracts: Challenges, advances and platforms,” *Future Generation Computer Systems*, vol. 105, pp. 475–491, 2020, doi: 10.1016/j.future.2019.12.019.

[84] R. Koulu, “Contracts as an Alternative to Enforcement,” vol. 13, no. 1, p. 65, 2016, doi: 10.2966/scrip.130116.41.

[85] G. W. Peters and E. Panayi, “Understanding modern banking ledgers through blockchain technologies: Future of transaction processing and smart contracts on the internet of money,” *New Economic Windows*, pp. 239–278, 2016, doi: 10.1007/978-3-319-42448-4_13.

[86] X. Song *et al.*, “Big Data and Emergency Management: Concepts, Methodologies, and Applications,” *IEEE Trans Big Data*, vol. 8, no. 2, pp. 397–419, Apr. 2022, doi: 10.1109/TBDATA.2020.2972871.

[87] E. D. Zamani, C. Smyth, S. Gupta, and D. Dennehy, “Artificial intelligence and big data analytics for supply chain resilience: a systematic literature review,” *Ann Oper Res*, vol. 327, no. 2, pp. 605–632, Aug. 2023, doi: 10.1007/s10479-022-04983-y.

[88] S. Zafar, H. ur Rashid Kayani, H. Burhan ul Haq, I. Khalid, and A. Nasir, “Big Data: Challenges, Popular Tools Of Big Data-Benefits And Applications,” *LUME*, vol. 10, p. 5, 2021, [Online]. Available: www.ijstr.org

[89] I. Ahmad *et al.*, “Communications Security in Industry X: A Survey”, doi: 10.36227/techrxiv.22128380.v1.

[90] H. B. Abdalla, “A brief survey on big data: technologies, terminologies and data-intensive applications,” *Journal of Big Data*, vol. 9, no. 1. Springer Science and Business Media Deutschland GmbH, Dec. 01, 2022. doi: 10.1186/s40537-022-00659-3.

[91] B. Ramesh, “Big Data Architecture,” 2015, pp. 29–59. doi: 10.1007/978-81-322-2494-5_2.

[92] S. Kaisler, F. Armour, J. A. Espinosa, and W. Money, “Big data: Issues and challenges moving forward,” in *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2013, pp. 995–1004. doi: 10.1109/HICSS.2013.645.

[93] Tong ji da xue (China), M. IEEE Systems, Chinese Association of Automation, Taiwan Association of System Science and Engineering, and Institute of Electrical and Electronics Engineers, *2014 International Conference on System Science and Engineering (ICSSE) : conference proceedings : July 11-13, 2014, Shanghai, China*.

[94] M. M. Ahemd, M. A. Shah, and A. Wahid, “IoT security: A layered approach for attacks & defenses,” *International Conference on Communication Technologies, ComTech 2017*, pp. 104–110, 2017, doi: 10.1109/COMTECH.2017.8065757.

[95] I. Andrea, C. Chrysostomou, and G. Hadjichristofi, “Internet of Things: Security vulnerabilities and challenges,” *Proc IEEE Symp Comput Commun*, vol. 2016-Febru, pp. 180–187, 2016, doi: 10.1109/ISCC.2015.7405513.

[96] Z. Ling, K. Liu, Y. Xu, Y. Jin, and X. Fu, “An End-to-End View of IoT Security and Privacy,” *2017 IEEE Global Communications Conference, GLOBECOM 2017 - Proceedings*, vol. 2018-Janua, pp. 1–7, 2017, doi: 10.1109/GLOCOM.2017.8254011.

[97] K. Zhao and L. Ge, “A survey on the internet of things security,” *Proceedings - 9th International Conference on Computational Intelligence and Security, CIS 2013*, pp. 663–667, 2013, doi: 10.1109/CIS.2013.145.

[98] S. Hameed, F. I. Khan, and B. Hameed, “Understanding Security Requirements and Challenges in Internet of Things (IoT): A Review,” *Journal of Computer Networks and Communications*, vol. 2019, pp. 1–18, 2019, doi: 10.1155/2019/9629381.

[99] Y. Qin, Q. Z. Sheng, N. J. G. Falkner, S. Dustdar, H. Wang, and A. V. Vasilakos, “When things matter: A survey on data-centric internet of things,” *Journal of Network and Computer Applications*, vol. 64, pp. 137–153, 2016, doi: 10.1016/j.jnca.2015.12.016.

[100] D. G. Orifama and H. Okoro, “Security Challenges in IoT Platforms and Possible Solutions,” vol. 8, no. 1, pp. 1–7, 2020, doi: 10.11648/j.iotcc.20200801.11.

[101] P. Varga, S. Plosz, G. Soos, and C. Hegedus, “Security threats and issues in automation IoT,” *IEEE International Workshop on Factory Communication Systems - Proceedings, WFCS*, 2017, doi: 10.1109/WFCS.2017.7991968.

[102] S. A. V. Jatti and V. J. K. Kishor Sontif, “Intrusion detection systems,” *International Journal of Recent Technology and Engineering*, vol. 8, no. 2 Special Issue 11, pp. 3976–3983, 2019, doi: 10.35940/ijrte.B1540.0982S1119.

[103] M. A. Amanullah *et al.*, “Deep learning and big data technologies for IoT security,” *Comput Commun*, vol. 151, pp. 495–517, 2020, doi: 10.1016/j.comcom.2020.01.016.

[104] M. Alrakhami and A. Alamri, “Wireless Sensor Networks Security: State of the Art,” no. September, pp. 1–11, 2018.

[105] P. R. K. Varma, V. V. Kumari, and S. S. Kumar, “Progress in Computing, Analytics and Networking,” vol. 710, no. January, pp. 785–793, 2018, doi: 10.1007/978-981-10-7871-2.

[106] G. Farjamnia, Y. Gasimov, and C. Kazimov, “Review of the Techniques Against the Wormhole Attacks on Wireless Sensor Networks,” *Wirel Pers Commun*, vol. 105, no. 4, pp. 1561–1584, 2019, doi: 10.1007/s11277-019-06160-0.

[107] K. Sharma, “Wireless Sensor Networks : An Overview on its Security Threats Wireless Sensor Networks : An Overview on its Security Threats,” *Int J Comput Appl*, no. May 2014, 2014, doi: 10.5120/1008-44.

[108] Z. Ren, X. Liu, R. Ye, and T. Zhang, “Security and privacy on internet of things,” *Proceedings of 2017 IEEE 7th International Conference on Electronics Information and Emergency Communication, ICEIEC 2017*, pp. 140–144, 2017, doi: 10.1109/ICEIEC.2017.8076530.

[109] I. Ali, S. Sabir, and Z. Ullah, “Internet of Things Security, Device Authentication and Access Control: A Review,” vol. 14, no. 8, pp. 456–466, 2019, [Online]. Available: http://arxiv.org/abs/1901.07309

[110] J. Ali, A. S. Khalid, E. Yafi, S. Musa, and W. Ahmed, “Towards a secure behavior modeling for IoT networks using blockchain,” *CEUR Workshop Proc*, vol. 2486, pp. 244–258, 2019.

[111] A. Yazdinejad, R. M. Parizi, A. Dehghantanha, Q. Zhang, and K.-K. R. Choo, “An Energy-efficient SDN Controller Architecture for IoT Networks with Blockchain-based Security,” *IEEE Trans Serv Comput*, vol. 1374, no. c, pp. 1–1, 2020, doi: 10.1109/tsc.2020.2966970.

[112] M. Abraham, H. Aithal, and K. Mohan, “Real time Smart Contracts for IoT using Blockchain and Collaborative Intelligence based Dynamic Pricing for the next generation Smart Toll Application,” 2020, [Online]. Available: http://arxiv.org/abs/2002.12654

[113] M. Shen, X. Tang, L. Zhu, X. Du, and M. Guizani, “Privacy-Preserving Support Vector Machine Training over Blockchain-Based Encrypted IoT Data in Smart Cities,” *IEEE Internet Things J*, vol. 6, no. 5, pp. 7702–7712, 2019, doi: 10.1109/JIOT.2019.2901840.

[114] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, “LSB: A Lightweight Scalable Blockchain for IoT security and anonymity,” *J Parallel Distrib Comput*, vol. 134, pp. 180–197, 2019, doi: 10.1016/j.jpdc.2019.08.005.

[115] S. Cities and Y. Byun, “A Data Verification System for CCTV Surveillance Cameras Using Blockchain Technology in,” 2020, doi: 10.3390/electronics9030484.

[116] N. Amara, H. Zhiqui, and A. Ali, “Cloud Computing Security Threats and Attacks with Their Mitigation Techniques,” *Proceedings - 2017 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, CyberC 2017*, vol. 2018-Janua, no. September, pp. 244–251, 2017, doi: 10.1109/CyberC.2017.37.

[117] P. C. Wei, D. Wang, Y. Zhao, S. K. S. Tyagi, and N. Kumar, “Blockchain data-based cloud data integrity protection mechanism,” *Future Generation Computer Systems*, vol. 102, pp. 902–911, 2020, doi: 10.1016/j.future.2019.09.028.

[118] N. Eltayieb, R. Elhabob, A. Hassan, and F. Li, “A blockchain-based attribute-based signcryption scheme to secure data sharing in the cloud,” *Journal of Systems Architecture*, vol. 102, no. August 2019, 2020, doi: 10.1016/j.sysarc.2019.101653.

[119] H. Wang, H. Qin, M. Zhao, X. Wei, H. Shen, and W. Susilo, “Blockchain-based fair payment smart contract for public cloud storage auditing,” *Inf Sci (N Y)*, vol. 519, pp. 348–362, 2020, doi: 10.1016/j.ins.2020.01.051.

[120] Z. Li, L. Liu, A. V. Barenji, and W. Wang, “Cloud-based Manufacturing Blockchain: Secure Knowledge Sharing for Injection Mould Redesign,” *Procedia CIRP*, vol. 72, pp. 961–966, 2018, doi: 10.1016/j.procir.2018.03.004.

[121] K. Rajwar, K. Deep, and S. Das, “An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges,” *Artif Intell Rev*, vol. 56, no. 11, pp. 13187–13257, Nov. 2023, doi: 10.1007/s10462-023-10470-y.

[122] T. Flutre, T. Julou, and L. Riboli-Sasco, “La théorie de la sélection naturelle présentée par Darwin et Wallace,” *BibNum*, Dec. 2009, doi: 10.4000/bibnum.622.

[123] A. E. Ezugwu *et al.*, “Metaheuristics: a comprehensive overview and classification along with bibliometric analysis,” *Artif Intell Rev*, vol. 54, no. 6, pp. 4237–4316, Aug. 2021, doi: 10.1007/s10462-020-09952-0.

[124] O. Al Jadaan, L. Rajamani, and C. R. Rao, “IMPROVED SELECTION OPERATOR FOR GA,” 2005. [Online]. Available: www.jatit.org

[125] A. S. Hameed, H. M. B. Alrikabi, A. A. Abdul-Razaq, Z. H. Ahmed, H. K. Nasser, and M. L. Mutar, “Appling the Roulette Wheel Selection Approach to Address the Issues of Premature

Convergence and Stagnation in the Discrete Differential Evolution Algorithm," *Applied Computational Intelligence and Soft Computing*, vol. 2023, 2023, doi: 10.1155/2023/8892689.

[126] B. Meniz and F. Tiryaki, "Genetic Algorithm Optimization with Selection Operator Decider," *Arab J Sci Eng*, 2024, doi: 10.1007/s13369-024-09068-5.

[127] A. Dwi Irfianti, R. Wardoyo, and S. Hartati, "DETERMINATION OF SELECTION METHOD IN GENETIC ALGORITHM FOR LAND SUITABILITY", doi: 10.1051/conf/2016.

[128] J. Vilma Roseline and D. D. Saravanan, "Crossover and Mutation Strategies applied in Job Shop Scheduling Problems," in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Nov. 2019. doi: 10.1088/1742-6596/1377/1/012031.

[129] D. M. Chitty, W. B. Yates, and E. Keedwell, "An edge quality aware crossover operator for application to the capacitated vehicle routing problem," in *GECCO 2022 Companion - Proceedings of the 2022 Genetic and Evolutionary Computation Conference*, Association for Computing Machinery, Inc, Jul. 2022, pp. 419–422. doi: 10.1145/3520304.3529046.

[130] V. A. Cicirello, "Non-Wrapping Order Crossover: An Order Preserving Crossover Operator that Respects Absolute Position," 2006.

[131] S. Mohammed Almufti, R. P. Maribojoc, and A. V. Pahuriray, "Ant Based System: Overview, Modifications and Applications from 1992 to 2022," *Polaris Global Journal of Scholarly Research and Trends*, vol. 1, no. 1, pp. 29–37, Oct. 2022, doi: 10.58429/pgjsrt.v1n1a85.

[132] S. Goss, S. Aron, J. L. Deneubourg, and J. M. Pasteels, "Self-organized Shortcuts in the Argentine Ant," Springer-Verlag, 1959.

[133] K. Hussain, M. N. Mohd Salleh, S. Cheng, and Y. Shi, "Metaheuristic research: a comprehensive survey," *Artif Intell Rev*, vol. 52, no. 4, pp. 2191–2233, Dec. 2019, doi: 10.1007/s10462-017-9605-z.

[134] M. Dorigo and T. Stützle, "Ant colony optimization: Overview and recent advances," in *International Series in Operations Research and Management Science*, vol. 272, Springer New York LLC, 2019, pp. 311–351. doi: 10.1007/978-3-319-91086-4_10.

[135] R. A. Rutenbar, "Simula ted Annealing Algorithms: A n Overview."

[136] D. Delahaye, S. Chaimatanan, and M. Mongeau, "Simulated annealing: From basics to applications," in *International Series in Operations Research and Management Science*, vol. 272, Springer New York LLC, 2019, pp. 1–35. doi: 10.1007/978-3-319-91086-4_1.

[137] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *J Chem Phys*, vol. 21, no. 6, pp. 1087–1092, 1953, doi: 10.1063/1.1699114.

[138] F. Glover, "Tabu Search: A Tutorial."

[139] F. Glover, “FUTURE PATHS FOR INTEGER PROGRAMMING AND LINKS TO ARTIFICIAL INTELLIGENCE,” 1986.

[140] V. K. Prajapati, M. Jain, and L. Chouhan, “Tabu Search Algorithm (TSA): A Comprehensive Survey,” in *Proceedings of 3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things, ICETCE 2020*, Institute of Electrical and Electronics Engineers Inc., Feb. 2020, pp. 222–229. doi: 10.1109/ICETCE48199.2020.9091743.

[141] R. Battiti and G. Tecchiolli, “Training Neural Nets with the Reactive Tabu Search,” *IEEE Trans Neural Netw*, vol. 6, no. 5, pp. 1185–1200, 1995, doi: 10.1109/72.410361.

[142] J. Kennedy, R. Eberhart, and bls gov, “Particle Swarm Optimization.”

[143] M. Jahandideh-Tehrani, O. Bozorg-Haddad, and H. A. Loáiciga, “Application of particle swarm optimization to water management: an introduction and overview,” *Environmental Monitoring and Assessment*, vol. 192, no. 5. Springer, May 01, 2020. doi: 10.1007/s10661-020-8228-z.

[144] M. Jain, V. Saihjpal, N. Singh, and S. B. Singh, “An Overview of Variants and Advancements of PSO Algorithm,” *Applied Sciences (Switzerland)*, vol. 12, no. 17. MDPI, Sep. 01, 2022. doi: 10.3390/app12178392.

[145] M. Clerc and J. Kennedy, “The Particle Swarm-Explosion, Stability, and Convergence in a Multidimensional Complex Space,” 2002.

[146] Y. Shi and R. Eberhart, “A Modified Particle Swarm Optimizer.”

[147] R. Burke, “Hybrid recommender systems: Survey and experiments,” *User Modelling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331–370, 2002, doi: 10.1023/A:1021240730564.

[148] U. Javed, K. Shaukat, I. A. Hameed, F. Iqbal, T. M. Alam, and S. Luo, “A Review of Content-Based and Context-Based Recommendation Systems,” *International Journal of Emerging Technologies in Learning*, vol. 16, no. 3, pp. 274–306, 2021, doi: 10.3991/ijet.v16i03.18851.

[149] G. Adomavicius and ; A Tuzhilin, “Incorporating Contextual Information in Recommender Systems Using a Multidimensional Approach,” 2005.

[150] P. Melville, R. J. Mooney, and R. Nagarajan, “Content-Boosted Collaborative Filtering for Improved Recommendations,” 2002. [Online]. Available: http://research.compaq.com/SRC/eachmovie

[151] P. Lops, M. de Gemmis, and G. Semeraro, “Content-based Recommender Systems: State of the Art and Trends,” in *Recommender Systems Handbook*, Springer US, 2011, pp. 73–105. doi: 10.1007/978-0-387-85820-3_3.

[152] R. Baeza-Yates and G. Navarro, “Faster Approximate String Matching 1,” 1999.

[153] B. Towle and C. Quinn, “Knowledge Based Recommender Systems Using Explicit User Models,” 2000. [Online]. Available: http://www.alexlit.com/

[154] M. Zihayat, A. Ayanso, X. Zhao, H. Davoudi, and A. An, “A utility-based news recommendation system,” *Decis Support Syst*, vol. 117, pp. 14–27, Feb. 2019, doi: 10.1016/j.dss.2018.12.001.

[155] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Item-Based Collaborative Filtering Recommendation Algorithms.”

[156] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl, “Applying Collaborative Filtering to Usenet News,” *Commun ACM*, vol. 40, no. 3, pp. 77–87, 1997, doi: 10.1145/245108.245126.

[157] B. Schafer, “LNCS 4321 - Collaborative Filtering Recommender Systems,” 2007. [Online]. Available: https://www.researchgate.net/publication/200121027

[158] L. Hamers *et al.*, “SIMILARITY MEASURES IN SCIENTOMETRIC RESEARCH: THE JACCARD INDEX VERSUS SALTON’S COSINE FORMULA,” 1989.

[159] Z. Cui *et al.*, “Personalized Recommendation System Based on Collaborative Filtering for IoT Scenarios,” *IEEE Trans Serv Comput*, vol. 13, no. 4, pp. 685–695, Jul. 2020, doi: 10.1109/TSC.2020.2964552.

[160] S. T. Lanza, L. M. Collins, D. R. Lemmon, and J. L. Schafer, “PROC LCA: A SAS Procedure for Latent Class Analysis,” 2007. [Online]. Available: http://methodology.psu.edu.

[161] G. Takács, I. Pilászy, B. Németh, and D. Tikk, “Matrix Factorization and Neighbor Based Algorithms for the Netflix Prize Problem General Terms.” [Online]. Available: http://sifter.org/~simon/journal/20061211.html

[162] Y. Koren, “The BellKor Solution to the Netflix Grand Prize,” 2009. [Online]. Available: www.netflixprize.com/leaderboard

[163] S. Bandyopadhyay, S. S. Thakur, and J. K. Mandal, “Product recommendation for e-commerce business by applying principal component analysis (PCA) and K-means clustering: benefit for the society,” *Innov Syst Softw Eng*, vol. 17, no. 1, pp. 45–52, Mar. 2021, doi: 10.1007/s11334-020-00372-5.

[164] K. Przystupa *et al.*, “Distributed singular value decomposition method for fast data processing in recommendation systems,” *Energies (Basel)*, vol. 14, no. 8, Apr. 2021, doi: 10.3390/en14082284.

[165] M. Rahul, V. Kumar, V. Yadav, and Rishabh, “Movie recommender system using single value decomposition and K-means clustering,” in *IOP Conference Series: Materials Science and Engineering*, IOP Publishing Ltd, Jan. 2021. doi: 10.1088/1757-899X/1022/1/012100.

[166] C. Desrosiers and G. Karypis, “A Comprehensive Survey of Neighborhood-based Recommendation Methods,” in *Recommender Systems Handbook*, Springer US, 2011, pp. 107–144. doi: 10.1007/978-0-387-85820-3_4.

[167] N. J. Belkin and W. B. Croft, “Illllllllll I! I:=! I~ Ill I~ {= D I~ !1 ~ Ill Information and Information TWO Sides of the Same Coin?”

[168] B. Kitts, D. Freed, and M. Vrieze, “Cross-sell: A Fast Promotion-Tunable Customer-item Recommendation Method Based on Conditionally Independent Probabilities,” 2000.

[169] G. Ball, J. Breese, and S. Researcher, “Emotion and Personality in a Conversational Character.”

[170] L. M. De Campos, J. M. Fernández-Luna, J. F. Huete, and M. A. Rueda-Morales, “Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks,” *International Journal of Approximate Reasoning*, vol. 51, no. 7, pp. 785–799, Sep. 2010, doi: 10.1016/j.ijar.2010.04.001.

[171] X. He, K. Deng, X. Wang, Y. Li, Y. D. Zhang, and M. Wang, “LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation,” in *SIGIR 2020 - Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Association for Computing Machinery, Inc, Jul. 2020, pp. 639–648. doi: 10.1145/3397271.3401063.

[172] M. J. Pazzani, “A Framework for Collaborative, Content-Based and Demographic Filtering.” [Online]. Available: http://www.ics.uci.edu/~pazzani/

[173] B. Walek and V. Fojtik, “A hybrid recommender system for recommending relevant movies using an expert system,” *Expert Syst Appl*, vol. 158, Nov. 2020, doi: 10.1016/j.eswa.2020.113452.

[174] A. Gunawardana and C. Meek, “A unified approach to building hybrid recommender systems,” in *RecSys’09 - Proceedings of the 3rd ACM Conference on Recommender Systems*, 2009, pp. 117–124. doi: 10.1145/1639714.1639735.

[175] S. Sengupta *et al.*, “A review of deep learning with special emphasis on architectures, applications and recent trends,” *Knowl Based Syst*, vol. 194, Apr. 2020, doi: 10.1016/j.knosys.2020.105596.

[176] A. K. Dey and G. D. Abowd, “Towards a Better Understanding of Context and Context-Awareness.”

[177] A. Zimmermann, A. Lorenz, and R. Oppermann, “An Operational Definition of Context,” 2007.

[178] C. Palmisano, A. Tuzhilin, and M. Gorgoglione, “Using context to improve predictive modeling of customers in personalization applications,” *IEEE Trans Knowl Data Eng*, vol. 20, no. 11, pp. 1535–1549, Nov. 2008, doi: 10.1109/TKDE.2008.110.

[179] G. Adomavicius, A. Tuzhilin, and R. Zheng, “REQUEST: A query language for customizing recommendations,” *Information Systems Research*, vol. 22, no. 1, pp. 99–117, 2011, doi: 10.1287/isre.1100.0274.

[180] L. Mathiassen, “Collaborative practice research,” *Information Technology & People*, vol. 15, no. 4, pp. 321–345, 2002, doi: 10.1108/09593840210453115.

[181] R. P. Adhikary, “Effectiveness of Content–based Instruction in Teaching Reading,” *Theory and Practice in Language Studies*, vol. 10, no. 5, p. 510, 2020, doi: 10.17507/tpls.1005.04.

[182] S. Sharma, A. Sharma, Y. Sharma, and M. Bhatia, “Recommender system using hybrid approach,” *Proceeding - IEEE International Conference on Computing, Communication and Automation, ICCCA 2016*, no. April 2016, pp. 219–223, 2017, doi: 10.1109/CCAA.2016.7813722.

[183] H. J. Zimmermann, “Fuzzy set theory,” *Wiley Interdiscip Rev Comput Stat*, vol. 2, no. 3, pp. 317–332, 2010, doi: 10.1002/wics.82.

[184] A. Gosain and S. Dahiya, “Performance Analysis of Various Fuzzy Clustering Algorithms: A Review,” *Procedia Comput Sci*, vol. 79, pp. 100–111, 2016, doi: 10.1016/j.procs.2016.03.014.

[185] T. Di Noia, R. Mirizzi, V. C. Ostuni, and D. Romito, “Exploiting the web of data in model-based recommender systems,” *RecSys’12 - Proceedings of the 6th ACM Conference on Recommender Systems*, no. September, pp. 253–256, 2012, doi: 10.1145/2365952.2366007.

[186] A. A. Abdallat, A. I. Alahmad, D. A. A. amimi, and J. A. AlWidian, “Hadoop MapReduce Job Scheduling Algorithms Survey and Use Cases,” *Mod Appl Sci*, vol. 13, no. 7, p. 38, 2019, doi: 10.5539/mas.v13n7p38.

[187] M. Uta, A. Felfernig, V.-M. Le, A. Popescu, T. N. T. Tran, and D. Helic, “Evaluating recommender systems in feature model configuration,” no. 1, pp. 58–63, 2021, doi: 10.1145/3461001.3471144.

[188] N. Bhalse and R. Thakur, “Algorithm for movie recommendation system using collaborative filtering,” *Mater Today Proc*, no. xxxx, pp. 1–6, 2021, doi: 10.1016/j.matpr.2021.01.235.

[189] K. Inagaki *et al.*, “A case of hepatitis E virus infection in a patient with y biliary cholangitisprimar,” *Acta Hepatologica Japonica*, vol. 58, no. 3, pp. 183–190, 2017, doi: 10.2957/kanzo.58.183.

[190] M. Pazzani and D. Billsus, “Learning and Revising User Profiles: The Identification of Interesting Web Sites,” *Mach Learn*, vol. 27, no. 3, pp. 313–331, 1997, doi: 10.1023/A:1007369909943.

[191] H. Christian, M. P. Agus, and D. Suhartono, “Single Document Automatic Text Summarization using Term Frequency-Inverse Document Frequency (TF-IDF),” *ComTech:*

Computer, Mathematics and Engineering Applications, vol. 7, no. 4, p. 285, 2016, doi: 10.21512/comtech.v7i4.3746.

[192] R. Madani, A. Ez-Zahout, and A. Idrissi, "An Overview of Recommender Systems in the Context of Smart Cities," *Proceedings of 2020 5th International Conference on Cloud Computing and Artificial Intelligence: Technologies and Applications, CloudTech 2020*, pp. 1–9, 2020, doi: 10.1109/CloudTech49835.2020.9365877.

[193] V. Vaidhehi and R. Suchithra, "A Systematic Review of Recommender Systems in Education," *International Journal of Engineering & Technology*, vol. 7, no. 3.4, p. 188, 2018, doi: 10.14419/ijet.v7i3.4.16771.

[194] Y. Ge *et al.*, "Understanding Echo Chambers in E-commerce Recommender Systems," *SIGIR 2020 - Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2261–2270, 2020, doi: 10.1145/3397271.3401431.

[195] A. Felfernig, V. M. Le, A. Popescu, M. Uta, T. N. T. Tran, and M. Atas, "An Overview of Recommender Systems and Machine Learning in Feature Modeling and Configuration," *ACM International Conference Proceeding Series*, 2021, doi: 10.1145/3442391.3442408.

[196] X. Zheng, L. Da Xu, and S. Chai, "QoS Recommendation in Cloud Services," *IEEE Access*, vol. 5, pp. 5171–5177, 2017, doi: 10.1109/ACCESS.2017.2695657.

[197] Y. R. Shrestha and Y. Yang, "Fairness in algorithmic decision-making: Applications in multi-winner voting, machine learning, and recommender systems," *Algorithms*, vol. 12, no. 9, pp. 1–28, 2019, doi: 10.3390/a12090199.

[198] I. Palomares, C. Porcel, L. Pizzato, I. Guy, and E. Herrera-Viedma, "Reciprocal Recommender Systems: Analysis of state-of-art literature, challenges and opportunities towards social recommendation," *Information Fusion*, vol. 69, pp. 103–127, 2021, doi: 10.1016/j.inffus.2020.12.001.

[199] D. S. Tarragó, C. Cornelis, R. Bello, and F. Herrera, "A multi-instance learning wrapper based on the Rocchio classifier for web index recommendation," *Knowl Based Syst*, vol. 59, pp. 173–181, 2014, doi: 10.1016/j.knosys.2014.01.008.

[200] R. M. Suleman and I. Korkontzelos, "Extending latent semantic analysis to manage its syntactic blindness," *Expert Syst Appl*, vol. 165, no. January 2020, p. 114130, 2021, doi: 10.1016/j.eswa.2020.114130.

[201] R. J. Kuo, C. K. Chen, and S. H. Keng, "Application of hybrid metaheuristic with perturbation-based K-nearest neighbors algorithm and densest imputation to collaborative filtering in recommender systems," *Inf Sci (N Y)*, vol. 575, pp. 90–115, 2021, doi: 10.1016/j.ins.2021.06.026.

[202] S. Renjith, A. Sreekumar, and M. Jathavedan, "An extensive study on the evolution of context-aware personalized travel recommender systems," *Inf Process Manag*, vol. 57, no. 1, p. 102078, 2020, doi: 10.1016/j.ipm.2019.102078.

[203] R. Panigrahi and S. Borah, “Rank Allocation to J48 Group of Decision Tree Classifiers using Binary and Multiclass Intrusion Detection Datasets,” *Procedia Comput Sci*, vol. 132, pp. 323–332, 2018, doi: 10.1016/j.procs.2018.05.186.

[204] J. Hamidzadeh, E. Rezaeenik, and M. Moradi, “Predicting users’ preferences by Fuzzy Rough Set Quarter-Sphere Support Vector Machine,” *Appl Soft Comput*, vol. 112, p. 107740, 2021, doi: 10.1016/j.asoc.2021.107740.

[205] K. Luo, S. Sanner, G. Wu, H. Li, and H. Yang, “Latent Linear Critiquing for Conversational Recommender Systems,” *The Web Conference 2020 - Proceedings of the World Wide Web Conference, WWW 2020*, vol. 2, pp. 2535–2541, 2020, doi: 10.1145/3366423.3380003.

[206] A. Yassine, L. Mohamed, and M. Al Achhab, “Intelligent recommender system based on unsupervised machine learning and demographic attributes,” *Simul Model Pract Theory*, vol. 107, p. 102198, 2021, doi: 10.1016/j.simpat.2020.102198.

[207] Q. X. Wang, X. Luo, Y. Li, X. Y. Shi, L. Gu, and M. S. Shang, “Incremental Slope-one recommenders,” *Neurocomputing*, vol. 272, pp. 606–618, 2018, doi: 10.1016/j.neucom.2017.07.033.

[208] T. V. R. Himabindu, V. Padmanabhan, and A. K. Pujari, “Conformal matrix factorization based recommender system,” *Inf Sci (N Y)*, vol. 467, pp. 685–707, 2018, doi: 10.1016/j.ins.2018.04.004.

[209] A. L. Vizine Pereira and E. R. Hruschka, “Simultaneous co-clustering and learning to address the cold start problem in recommender systems,” *Knowl Based Syst*, vol. 82, pp. 11–19, 2015, doi: 10.1016/j.knosys.2015.02.016.

[210] F. Maazouzi, H. Zarzour, and Y. Jararweh, “An effective recommender system based on clustering technique for ted talks,” *International Journal of Information Technology and Web Engineering*, vol. 15, no. 1, pp. 35–51, 2020, doi: 10.4018/IJITWE.2020010103.

[211] C. Djellali and M. Adda, “A new hybrid deep learning model based-recommender system using artificial neural network and hidden Markov model,” *Procedia Comput Sci*, vol. 175, no. 2019, pp. 214–220, 2020, doi: 10.1016/j.procs.2020.07.032.

[212] A. S. Abohamama and E. Hamouda, “A hybrid energy–Aware virtual machine placement algorithm for cloud environments,” *Expert Syst Appl*, vol. 150, p. 113306, 2020, doi: 10.1016/j.eswa.2020.113306.

[213] S. P. M. Ziyath and S. Senthilkumar, “MHO: meta heuristic optimization applied task scheduling with load balancing technique for cloud infrastructure services,” *J Ambient Intell Humaniz Comput*, vol. 12, no. 6, pp. 6629–6638, Jun. 2021, doi: 10.1007/s12652-020-02282-7.

[214] A. Ndayikengurukiye, A. Ez-zahout, and F. Omary, “A Big Data driven model for secured systems: A Blockchain-based architecture for Cloud Computing and IoT security.”

[215] A. Ndayikengurukiye, A. Ez-Zahout, A. Aboubakr, Y. Charkaoui, and O. Fouzia, "Resource Optimisation in Cloud Computing: Comparative Study of Algorithms Applied to Recommendations in a Big Data Analysis Architecture," *Journal of Automation, Mobile Robotics and Intelligent Systems*, vol. 2021, no. 4, pp. 65–75, 2021, doi: 10.14313/JAMRIS/4-2021/28.

[216] S. Omer, S. Azizi, M. Shojafar, and R. Tafazolli, "A priority, power and traffic-aware virtual machine placement of IoT applications in cloud data centers," *Journal of Systems Architecture*, vol. 115, May 2021, doi: 10.1016/j.sysarc.2021.101996.

[217] U. Chourasia, R. Gandhi, P. Vishwavidyalaya, and D. Sanjay, "Optimal Load Balancing in Cloud using ANFIS and MGWO based Polynomial Neural Network", doi: 10.21203/rs.3.rs-529618/v1.

[218] M. S. Kumar and G. R. Karri, "EEOA: Cost and Energy Efficient Task Scheduling in a Cloud-Fog Framework," *Sensors*, vol. 23, no. 5, Mar. 2023, doi: 10.3390/s23052445.

[219] Y. Li, X. Lin, and J. Liu, "An improved gray wolf optimization algorithm to solve engineering problems," *Sustainability (Switzerland)*, vol. 13, no. 6, Mar. 2021, doi: 10.3390/su13063208.

[220] J. Too, A. R. Abdullah, N. M. Saad, N. M. Ali, and W. Tee, "A new competitive binary grey wolf optimizer to solve the feature selection problem in EMG signals classification," *Computers*, vol. 7, no. 4, Dec. 2018, doi: 10.3390/computers7040058.

[221] E. G. Dada, S. B. Joseph, D. O. Oyewola, A. A. Fadele, H. Chiroma, and S. M. Abdulhamid, "Application of Grey Wolf Optimization Algorithm: Recent Trends, Issues, and Possible Horizons," *Gazi University Journal of Science*, vol. 35, no. 2, pp. 485–504, Jun. 2022, doi: 10.35378/gujs.820885.

[222] G. Shial, S. Sahoo, and S. Panigrahi, "An Enhanced GWO Algorithm with Improved Explorative Search Capability for Global Optimization and Data Clustering," *Applied Artificial Intelligence*, vol. 37, no. 1, 2023, doi: 10.1080/08839514.2023.2166232.

[223] N. M. Sallam, A. I. Saleh, H. Arafat Ali, and M. M. Abdelsalam, "An Efficient Strategy for Blood Diseases Detection Based on Grey Wolf Optimization as Feature Selection and Machine Learning Techniques," *Applied Sciences (Switzerland)*, vol. 12, no. 21, Nov. 2022, doi: 10.3390/app122110760.

[224] Y. Hou, H. Gao, Z. Wang, and C. Du, "Improved Grey Wolf Optimization Algorithm and Application," *Sensors*, vol. 22, no. 10, May 2022, doi: 10.3390/s22103810.

[225] H. M. R. Al-Khafaji, "Improving Quality Indicators of the Cloud-Based IoT Networks Using an Improved Form of Seagull Optimization Algorithm," *Future Internet*, vol. 14, no. 10, Oct. 2022, doi: 10.3390/fi14100281.

[226] P. Hu, J. S. Pan, and S. C. Chu, "Improved Binary Grey Wolf Optimizer and Its application for feature selection," *Knowl Based Syst*, vol. 195, May 2020, doi: 10.1016/j.knosys.2020.105746.

[227] S. Azizi, M. Shojafar, J. Abawajy, and R. Buyya, “GRVMP: A Greedy Randomized Algorithm for Virtual Machine Placement in Cloud Data Centers,” *IEEE Syst J*, vol. 15, no. 2, pp. 2571–2582, Jun. 2021, doi: 10.1109/JSYST.2020.3002721.

[228] A. Al-Moalmi, J. Luo, A. Salah, and K. Li, “Optimal virtual machine placement based on grey wolf optimization,” *Electronics (Switzerland)*, vol. 8, no. 3, Mar. 2019, doi: 10.3390/electronics8030283.

[229] S. Gharehpasha, M. Masdari, and A. Jafarian, “Virtual machine placement in cloud data centers using a hybrid multi-verse optimization algorithm,” *Artif Intell Rev*, vol. 54, no. 3, pp. 2221–2257, Mar. 2021, doi: 10.1007/s10462-020-09903-9.

[230] A. Yousefipour, A. M. Rahmani, and M. Jahanshahi, “Improving the load balancing and dynamic placement of virtual machines in cloud computing using particle swarm optimization algorithm,” *International Journal of Engineering Transactions C: Aspects*, vol. 34, no. 6, pp. 1419–1429, Jun. 2021, doi: 10.5829/ije.2021.34.06c.05.

[231] B. Liang, D. Wu, P. Wu, and Y. Su, “An energy-aware resource deployment algorithm for cloud data centers based on dynamic hybrid machine learning,” *Knowl Based Syst*, vol. 222, Jun. 2021, doi: 10.1016/j.knosys.2021.107020.

[232] S. S. Sefati, M. Mousavinasab, and R. Zareh Farkhady, “Load balancing in cloud computing environment using the Grey wolf optimization algorithm based on the reliability: performance evaluation,” *Journal of Supercomputing*, vol. 78, no. 1, pp. 18–42, Jan. 2022, doi: 10.1007/s11227-021-03810-8.

[233] J. Lu, W. Zhao, H. Zhu, J. Li, Z. Cheng, and G. Xiao, “Optimal machine placement based on improved genetic algorithm in cloud computing,” *Journal of Supercomputing*, vol. 78, no. 3, pp. 3448–3476, Feb. 2022, doi: 10.1007/s11227-021-03953-8.

[234] H. Talebian *et al.*, “Optimizing virtual machine placement in IaaS data centers: taxonomy, review and open issues,” *Cluster Comput*, vol. 23, no. 2, pp. 837–878, Jun. 2020, doi: 10.1007/s10586-019-02954-w.

[235] M. Ghetas, “A multi-objective Monarch Butterfly Algorithm for virtual machine placement in cloud computing,” *Neural Comput Appl*, vol. 33, no. 17, pp. 11011–11025, Sep. 2021, doi: 10.1007/s00521-020-05559-2.

[236] J. Luo, W. Song, and L. Yin, “Reliable Virtual Machine Placement Based on Multi-Objective Optimization with Traffic-Aware Algorithm in Industrial Cloud,” *IEEE Access*, vol. 6, pp. 23043–23052, Mar. 2018, doi: 10.1109/ACCESS.2018.2816983.

[237] I. Behera and S. Sobhanayak, “Task scheduling optimization in heterogeneous cloud computing environments: A hybrid GA-GWO approach,” *J Parallel Distrib Comput*, vol. 183, Jan. 2024, doi: 10.1016/j.jpdc.2023.104766.

[238] T. B. Hewage, S. Ilager, M. A. Rodriguez, and R. Buyya, “CloudSim express: A novel framework for rapid low code simulation of cloud computing environments,” *Softw Pract Exp*, 2023, doi: 10.1002/spe.3290.

[239] M. A. Khan, A. Paplinski, A. M. Khan, M. Murshed, and R. Buyya, “Dynamic virtual machine consolidation algorithms for energy-efficient cloud resource management: A review,” in *Sustainable Cloud and Energy Services: Principles and Practice*, Springer International Publishing, 2017, pp. 135–165. doi: 10.1007/978-3-319-62238-5_6.

[240] H. Li, L. Wen, Y. Liu, and Y. Shen, “More than Meets One Core: An Energy-Aware Cost Optimization in Dynamic Multi-Core Processor Server Consolidation for Cloud Data Center,” *Electronics (Switzerland)*, vol. 11, no. 20, Oct. 2022, doi: 10.3390/electronics11203377.

[241] J. Wang, H. Gu, J. Yu, Y. Song, X. He, and Y. Song, “Research on virtual machine consolidation strategy based on combined prediction and energy-aware in cloud computing platform,” *Journal of Cloud Computing*, vol. 11, no. 1, Dec. 2022, doi: 10.1186/s13677-022-00309-2.

[242] D. Zhu, “Max–min Bin Packing Algorithm and its application in nano-particles filling,” *Chaos Solitons Fractals*, vol. 89, pp. 83–90, Aug. 2016, doi: 10.1016/j.chaos.2015.09.024.

[243] M. Masdari and M. Zangakani, “Green Cloud Computing Using Proactive Virtual Machine Placement: Challenges and Issues,” *J Grid Comput*, 2019, doi: 10.1007/s10723-019-09489-9.

[244] J. Singh and N. K. Walia, “A Comprehensive Review of Cloud Computing Virtual Machine Consolidation,” *IEEE Access*, vol. 11. Institute of Electrical and Electronics Engineers Inc., pp. 106190–106209, 2023. doi: 10.1109/ACCESS.2023.3314613.

[245] S. Gharehpasha, M. Masdari, and A. Jafarian, “Power efficient virtual machine placement in cloud data centers with a discrete and chaotic hybrid optimization algorithm,” *Cluster Comput*, vol. 24, no. 2, pp. 1293–1315, Jun. 2021, doi: 10.1007/s10586-020-03187-y.

[246] R. Regaieg, M. Koubàa, Z. Ales, and T. Aguili, “Multi-objective optimization for VM placement in homogeneous and heterogeneous cloud service provider data centers,” *Computing*, vol. 103, no. 6, pp. 1255–1279, Jun. 2021, doi: 10.1007/s00607-021-00915-z.

[247] K. RahimiZadeh and A. Dehghani, “Design and evaluation of a joint profit and interference-aware VMs consolidation in IaaS cloud datacenter,” *Cluster Comput*, vol. 24, no. 4, pp. 3249–3275, Dec. 2021, doi: 10.1007/s10586-021-03310-7.

[248] A. Beloglazov, J. Abawajy, and R. Buyya, “Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing,” *Future Generation Computer Systems*, vol. 28, no. 5, pp. 755–768, May 2012, doi: 10.1016/j.future.2011.04.017.

[249] Z. Xiao, J. Jiang, Y. Zhu, Z. Ming, S. Zhong, and S. Cai, “A solution of dynamic VMs placement problem for energy consumption optimization based on evolutionary game theory,” *Journal of Systems and Software*, vol. 101, pp. 260–272, Mar. 2015, doi: 10.1016/j.jss.2014.12.030.

[250] F. Alharbi, Y. C. Tian, M. Tang, M. H. Ferdaus, W. Z. Zhang, and Z. G. Yu, "Simultaneous application assignment and virtual machine placement via ant colony optimization for energy-efficient enterprise data centers," *Cluster Comput*, vol. 24, no. 2, pp. 1255–1275, Jun. 2021, doi: 10.1007/s10586-020-03186-z.

[251] S. Y. Rashida, M. Sabaei, M. M. Ebadzadeh, and A. M. Rahmani, "A memetic grouping genetic algorithm for cost efficient VM placement in multi-cloud environment," *Cluster Comput*, vol. 23, no. 2, pp. 797–836, Jun. 2020, doi: 10.1007/s10586-019-02956-8.

[252] S. Nabavi, L. Wen, S. S. Gill, and M. Xu, "Seagull optimization algorithm based multi-objective VM placement in edge-cloud data centers," *Internet of Things and Cyber-Physical Systems*, vol. 3, pp. 28–36, 2023, doi: 10.1016/j.iotcps.2023.01.002.

[253] G. Dhiman, M. Garg, A. Nagar, V. Kumar, and M. Dehghani, "A novel algorithm for global optimization: Rat Swarm Optimizer," *J Ambient Intell Humaniz Comput*, vol. 12, no. 8, pp. 8457–8482, Aug. 2021, doi: 10.1007/s12652-020-02580-0.

[254] S. Mohapatra and P. Mohapatra, "American zebra optimization algorithm for global optimization problems," *Sci Rep*, vol. 13, no. 1, Dec. 2023, doi: 10.1038/s41598-023-31876-2.

[255] Z. Ma, G. Wu, P. N. Suganthan, A. Song, and Q. Luo, "Performance assessment and exhaustive listing of 500+ nature-inspired metaheuristic algorithms," *Swarm Evol Comput*, vol. 77, Mar. 2023, doi: 10.1016/j.swevo.2023.101248.

[256] R. Boopathi and E. S. Samundeeswari, "Amended Hybrid Scheduling for Cloud Computing with Real-Time Reliability Forecasting," *International Journal of Computer Networks and Applications*, vol. 10, no. 3, pp. 310–324, May 2023, doi: 10.22247/ijcna/2023/221887.

[257] P. Jain and S. K. Sharma, "A Load Balancing Aware Task Scheduling using Hybrid Firefly Salp Swarm Algorithm in Cloud Computing," *International Journal of Computer Networks and Applications*, vol. 10, no. 6, pp. 914–933, Nov. 2023, doi: 10.22247/ijcna/2023/223686.

[258] G. Dhiman and V. Kumar, "Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems," *Knowl Based Syst*, vol. 165, pp. 169–196, Feb. 2019, doi: 10.1016/j.knosys.2018.11.024.

[259] V. Kumar, Di. Kumar, M. Kaur, Di. Singh, S. A. Idris, and H. Alshazly, "A Novel Binary Seagull Optimizer and its Application to Feature Selection Problem," *IEEE Access*, vol. 9, pp. 103481–103496, 2021, doi: 10.1109/ACCESS.2021.3098642.

[260] G. Dhiman *et al.*, "EMoSOA: a new evolutionary multi-objective seagull optimization algorithm for global optimization," *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 2, pp. 571–596, Feb. 2021, doi: 10.1007/s13042-020-01189-1.

[261] F. M. Alamgir and M. S. Alam, "An artificial intelligence driven facial emotion recognition system using hybrid deep belief rain optimization," *Multimed Tools Appl*, vol. 82, no. 2, pp. 2437–2464, Jan. 2023, doi: 10.1007/s11042-022-13378-x.

[262] Q. Wang, M. Zhang, and S. Abdolhosseinzadeh, “Application of modified seagull optimization algorithm with archives in urban water distribution networks: Dealing with the consequences of sudden pollution load,” *Heliyon*, vol. 10, no. 3, p. e24920, Feb. 2024, doi: 10.1016/j.heliyon.2024.e24920.

[263] P. Krishnadoss, V. K. Poornachary, P. Krishnamoorthy, and L. Shanmugam, “Improvised Seagull Optimization Algorithm for Scheduling Tasks in Heterogeneous Cloud Environment,” *Computers, Materials and Continua*, vol. 74, no. 2, pp. 2461–2478, 2023, doi: 10.32604/cmc.2023.031614.

[264] X. Liu, G. Li, and P. Shao, “A Multi-Mechanism Seagull Optimization Algorithm Incorporating Generalized Opposition-Based Nonlinear Boundary Processing,” *Mathematics*, vol. 10, no. 18, Sep. 2022, doi: 10.3390/math10183295.

[265] B. B. Al-Onazi, H. Alshamrani, F. O. Aldaajeh, A. S. A. Aziz, and M. Rizwanullah, “Modified Seagull Optimization with Deep Learning for Affect Classification in Arabic Tweets,” *IEEE Access*, vol. 11, pp. 98958–98968, 2023, doi: 10.1109/ACCESS.2023.3310873.

[266] I. Abdullaev, N. Prodanova, K. A. Bhaskar, E. L. Lydia, S. Kadry, and J. Kim, “Task Offloading and Resource Allocation in IoT Based Mobile Edge Computing Using Deep Learning,” *Computers, Materials and Continua*, vol. 76, no. 2, pp. 1463–1477, Aug. 2023, doi: 10.32604/cmc.2023.038417.

[267] J. Xue, X. Liu, H. Xu, and D. Zhang, “Research on the seagull optimization algorithm-based convolutional neural network rolling bearing fault diagnosis method,” *Engineering Research Express*, vol. 5, no. 3, Sep. 2023, doi: 10.1088/2631-8695/acf09a.

[268] Q. Xia, Y. Ding, R. Zhang, H. Zhang, S. Li, and X. Li, “Optimal Performance and Application for Seagull Optimization Algorithm Using a Hybrid Strategy,” *Entropy*, vol. 24, no. 7, Jul. 2022, doi: 10.3390/e24070973.

[269] E. S. Ghith and F. A. A. Tolba, “Tuning PID Controllers Based on Hybrid Arithmetic Optimization Algorithm and Artificial Gorilla Troop Optimization for Micro-Robotics Systems,” *IEEE Access*, vol. 11, pp. 27138–27154, 2023, doi: 10.1109/ACCESS.2023.3258187.

[270] M. Hanif, N. Mohammad, K. Biswas, and B. Harun, “Seagull Optimization Algorithm for Solving Economic Load Dispatch Problem,” in *3rd International Conference on Electrical, Computer and Communication Engineering, ECCE 2023*, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/ECCE57851.2023.10101516.

[271] D. G. Nair, K. P. Sanal Kumar, S. Anu, and H. Nair, “Automated Social Distance Recognition and Classification using Seagull Optimization Algorithm with Multilayer Perceptron,” 2023. [Online]. Available: https://ballisticsjournal.com

[272] A. Mohammadzadeh, D. Javaheri, and J. Artin, “Chaotic hybrid multi-objective optimization algorithm for scientific workflow scheduling in multisite clouds,” *Journal of the Operational Research Society*, 2023, doi: 10.1080/01605682.2023.2195426.

[273] S. Kaushaley, B. Shaw, and J. Ranjan Nayak, “Optimized Machine Learning based forecasting model for Solar Power Generation by using Crow Search Algorithm and Seagull Optimization Algorithm,” 2022, doi: 10.21203/rs.3.rs-1987438/v1.

[274] S. Farzai, M. H. Shirvani, and M. Rabbani, “Multi-objective communication-aware optimization for virtual machine placement in cloud datacenters,” *Sustainable Computing: Informatics and Systems*, vol. 28, Dec. 2020, doi: 10.1016/j.suscom.2020.100374.

[275] Y. Yuan *et al.*, “Coronavirus Mask Protection Algorithm: A New Bio-inspired Optimization Algorithm and Its Applications,” *J Bionic Eng*, vol. 20, no. 4, pp. 1747–1765, Jul. 2023, doi: 10.1007/s42235-023-00359-5.

[276] F. A. Zeidabadi, M. Dehghani, P. Trojovský, Š. Hubálovský, V. Leiva, and G. Dhiman, “Archery Algorithm: A Novel Stochastic Optimization Algorithm for Solving Optimization Problems,” *Computers, Materials and Continua*, vol. 72, no. 1, pp. 399–416, 2022, doi: 10.32604/cmc.2022.024736.

[277] Y. Du, H. Yuan, K. Jia, and F. Li, “Research on Threshold Segmentation Method of Two-Dimensional Otsu Image Based on Improved Sparrow Search Algorithm,” *IEEE Access*, vol. 11, pp. 70459–70469, 2023, doi: 10.1109/ACCESS.2023.3293191.

[278] A. Ndayikengurukiye, A. Ez-Zahout, and F. Omary, “Optimizing Virtual Machines Placement in a Heterogeneous Cloud Data Center System,” *International Journal of Computer Networks and Applications*, vol. 11, no. 1, p. 1, Feb. 2024, doi: 10.22247/ijcna/2024/224431.

ANNEXES

An overview of the different methods for optimizing the virtual resources placement in the Cloud Computing

Journal of Physics: Conference Series **1743** (2021) 012030 doi:10.1088/1742-6596/1743/1/012030

An overview of the different methods for optimizing the virtual resources placement in the Cloud Computing.

Aristide Ndayikengurukiye[1], Abderrahmane EZ-Zahout[1] and Fouzia Omary[1]

lemure.dede@gmail.com, abderrahmane.ezzahout@um5.ac.ma and omary@fsr.ac.ma

E-mail: lemure.dede@gmail.com

Abstract. The emergence of Cloud Computing has driven users of new technologies to adopt it. This is made possible by the used technologies, namely the virtualization, which is the key element in Cloud Computing. The management and orchestration of virtual resources (the virtual machine placement and VNFs) in the Cloud remains a complex task, which, recently, attracted the interest of many researchers.This paper provides an overview of the techniques adopted to optimize the virtual machines placement and the virtual networks functions. Focused on different resources (CPU, memory, bandwidth and storage)applied in a virtualized environment. These techniques always target several approaches including the improvement of the QoS defined in the service level agreements (SLA), the good management of the energy used by physical resources, the allocation of resources and the management tasks in the data center.

1. Introduction

In recent years, many technologies have succeeded each other to alter the exploitation of IT resources. This has given rise to the Cloud Computing technology, which consists of providing virtualized services like information storage and calculations, sometimes via the net or internal network. rather than acquisition high direct prices within the purchase of IT infrastructure and addressing computer code and hardware maintenance and upgrades, firms will source their IT has to the Cloud. This is often created potential by virtualization, which involves running multiple virtual machines connected to every different via virtual networks running on one physical machine. The implementation of those virtualized resources, as well as virtual machines, the square measure provided by computer code, referred to as hypervisors. In alternative words, virtualization could be a technique for sharing one physical instance of a resource or application across multiple purchasers. This technology not solely provides a virtual atmosphere for physical instances and applications, however additionally storage and networks. The diagram below summarizes the taxonomy of virtualization. We tend to show the various sorts of virtualization in keeping with every application domain and for every kind, we tend to assign the techniques that area unit applied thereto. On the far side of these techniques, it shows the requirement to contemplate the role of management and security in virtualization.

The proliferation of Cloud Computing has led to the creation of large-scale data centers containing thousands of virtualized computing nodes and consuming huge amounts of electrical

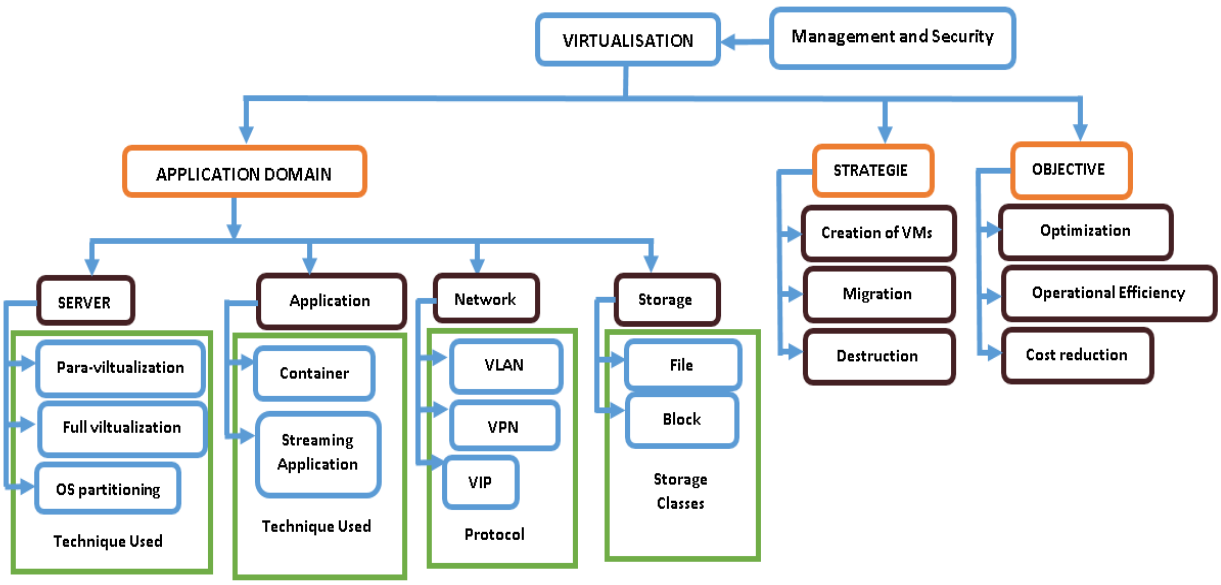

Figure 1. Virtualization taxonomy

energy to run and cool physical resources. Several techniques are implemented through virtualization to properly manage virtualized resources and address the problems associated with these energy consumptions while ensuring a good quality of service in the datacenters. The VMs placement is done while solving the problem of server consolidation. To achieve this, Virtual Machine placement is the most widely used technique. It is subdivided into four steps including detection of host overload, detection of the host under load, selection, and migration of the VM [1].

With the evolution of technology, it is important to also highlight the placement of virtual network functions to establish the placement of virtual machines and maximize QoS optimization and energy exploitation. In 2012, The Europian Telecommunications Standards Institute established this new concept of virtualization of network functions and defined its basic architecture composed of three main elements: Virtual Network Functions (VNFs), VNF Infrastructure and MANO (Management and Orchestration). The VNF adds new functionality to all communication networks and requires a new set of management and orchestration functions. In existing networks, implementations of the virtual network function are often associated with the infrastructure on which they run. The NFV separates software implementations of network functions from the computing, storage, and network resources they use. Virtualization isolates network functions from these resources through a virtualization layer as shown in the figure below.

In this article, we provide an overview of the different methods for optimizing the virtual resources placement in the Cloud. The remainder of the paper is organized as follows. In Section II, we first present work on the placement of virtual machines. In section III, we present the different algorithms used in the virtual machines placement and the metrics used to evaluate them. In section IV, we present the different machine learning techniques used to predict resource usage in Cloud Computing. In Section V, we classified various VNF placement work into two broad categories. In Section VI, we summarize the work..

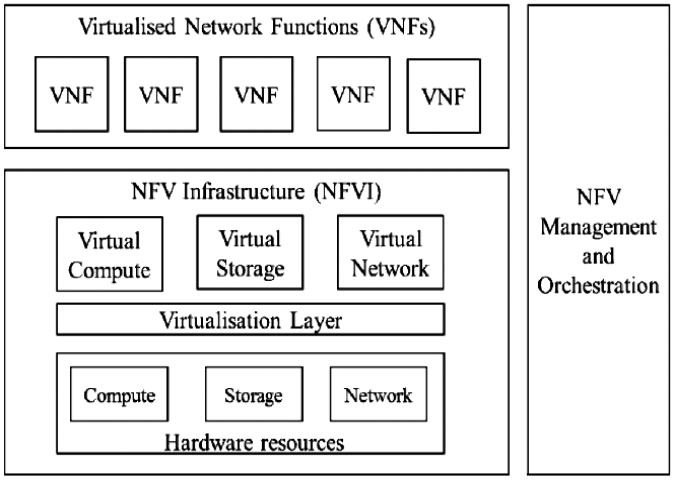

Figure 2. Architecture of the virtual function

2. Related work

In the literature, several works related to the placement of virtual machines in the Cloud have been proposed. For example, Weijia S. et al. have proposed an approach that uses migration to dynamically allocate data center resources according to application needs while optimizing the number of servers used. They developed a bin-packing resource allocation algorithm that can both avoid overloads and apply IT[2].

In order to optimise the allocation of resources, N . Janani , R . D . Shiva Jegan and P . Prakash have implemented the multidimensional backpack algorithm to minimize the migration of virtual machines, thus maximizing resource utilization. To improve the efficiency of the backpack algorithm, they also used genetic algorithms to optimize the solution of the first algorithm[3].

Mosa and R. Sakellariou proposed a parameter-based VM placement solution that works in the range of workload and full reservation to mitigate problems on resource reservation and demand. This solution was implemented while considering multiple resources (CPU, RAM, and bandwidth) [4].

C. Ghribi, M. Hadji, and D. Zeghlache have implemented algorithms acting together to reduce energy consumption and migration costs with acceptable convergence time compared to the other algorithms being compared. The optimal allocation algorithm is solved as the bin packing problem to reduce energy consumption and is compared to the fit algorithm [5].

A. Abdelsamea, A. A. El-moursy, E. E. Hemayed, and H. Eldeeb proposed the use of hybrid factors to improve VM consolidation. They developed a multiple regression algorithm taking into account CPU, memory, and bandwidth for the detection of host overload. The proposed algorithm reduces power consumption while ensuring SLA compliance compared to the other regression algorithms being compared [6].

M. Alaul, H. Monil, and R. M. Rahman proposed an algorithm for the detection of host overloads based on the mean, median, and standard deviation of virtual machine usage. To improve the performance of selecting VMs to be migrated from one host to another, they proposed a method that makes the decision in an intelligent way[7].

M. R. Chowdhury and R. M. Rahman presented an approach to generate multiple copies of virtual machines without sacrificing QoS. An algorithm based on dynamic programming and local search was provided to determine the number of VM copies and place them on servers to minimize the total power cost in a Cloud Computing system. The proposed solution offers a

flexible method to increase the energy efficiency of the Cloud Computing system or even increase the availability of resources in data centers compared to existing solutions[8].
C. Wei, Z. H. Hu, and Y. G. Wang have proposed a heuristic algorithm to reduce energy consumption and optimize the resource waste problem. This algorithm reduces the number of physical machines in operation by maximizing and balancing the load that are applied. They have also introduced a penalty and reward mechanism to manage VMP of virtual machines on physical machines[9].
The table below summarizes the various works already discussed on the methods used in the placement of virtual machines. It shows the objectives of these various works, the resources considered (CPU, RAM, Storage and bandwidth), the methods used as well as the algorithms used for the comparison that were the subject of the previous works on virtual machine placement. It shows that several works did not consider all the resources, which creates a hyndicap during the VMP.

3. Virtual machine placement algorithms and their evaluation metrics

3.1. Evaluation algorithms

3.1.1. Bin packing Algorithm: The main objective of this algorithm is to insert a set of elements or items with a different size in a minimum number of boxes. The latter considers VMs as the objects that must be inserted in the physical machines considered as bins. With its variants such as Fisrt Fit, Best Fit, Worst Fit Discreasing, this technique is used to optimize the placement of virtual machines. The main objective of this algorithm is to reduce energy consumption and the allocation of virtual resources in data centers. The authors in[15, 16, 17, 18] used the different categories of the bin packing technique to map virtual machines on physical machines.

3.1.2. Stochastic Integer programming: It is a modeling and optimization method involving the notions of probability and which can be analyzed statistically but cannot be predicted accurately. It is useful in cases where the claims are not known, but the distribution of claims is known or can be estimated. Algorithms that use this technique can allocate resources in three phases: reservation, utilization and on demand. This approach has been applied in [2, 19, 20] for VM placement.

3.1.3. Genetic Algorithm: Generally a genetic algorithm is an optimization technique based on natural and genetic selection. In [21, 22, 23] the authors also employed this technique in the placement of virtual machines by making natural selection of the solution close to all possible solutions. These algorithms can also be considered as a bin Packing algorithm when it takes additional constraints while optimizing costs.

3.1.4. Constraint programming Algorithm: The problem of virtual machine placement in the Cloud can be formulated as a constraint programming problem by taking into account several constraints. It is a technique that allows to solve multi-objective problems in virtual machine placement [24, 25].
For the virtual machines placement, several constraints can be taken into account such as constrained SLA, constrained QoS, constrained capacity. The CPU, memory bandwidth and storage are the different dimensions considered for each physical machine. P_i^r the set of resources of each physical machine i composed of $P_i^r = \{p_1^r, p_2^r, ..., p_n^r\}$ which represent the resources $p_i^{rcpu}, p_i^{rram}, p_i^{rbw}, p_i^{rsto}$. VM allocation and resource consumption of resources used by VMs must be less than that of the physical machine:

			Considered resources				
Authors	Objectives	Method used	CPU	Memory	Storage	Bandwidth	Best than
Weijia S. and al[2]	Resource allocation and green computing	Algorithm bin packing	✓	-	-	-	Black box, Offline bin packing and VectorDot
Gao, Liang Liu [10]	Waste of resources and energy consumption	Algorithm Ant colonies	✓	✓	-	-	MMAS Multi objectve genetic algorithm and binpacking
Janani and Shiva [3]	Resource use, energy consumption	Genetic algorithm	✓	✓	✓	✓	FF,FFD, Next made, Random Most fit and full
Mohammad A. and M. Rashedur [7]	Energy consumption and SLA	fuzzy search algorithm	✓	✓	-	-	IQR, RS, PBA MAD THR
Amany A. Ali A., E. and Hesham Elsayed E. [11]	Consumption of energy and SLA	multiple linear regression algorithm	✓	✓	-	✓	single LRR regression algorithm and LR
Abdelkhalik Mosa and Rizos Sakellariou [4]	Resource utilization, SLA	Sorting algorithm	✓	✓	-	✓	-
C. Ghribi, Mr. Hadji and D. Zeghlache [12]	Energy consumption	Constraint programming algorithm	✓	-	-	-	Fit algorithm

Table 1. Comparison of work on Virtual Machine Placement

$$\sum_{j}^{v} v_i^{rcpu} \; < \; p_i^{rcpu} \; \forall \; j \; \in \; v \tag{1}$$

$$\sum_{j}^{v} v_i^{rram} \; < \; p_i^{rram} \; \forall \; j \; \in \; v \tag{2}$$

$$\sum_{j}^{v} v_i^{rbw} \; < \; p_i^{rbw} \; \forall \; j \; \in \; v \tag{3}$$

$$\sum_{j}^{v} v_i^{rsto} \; < \; p_i^{rsto} \; \forall \; j \; \in \; v \tag{4}$$

3.2. Metrics for evaluating virtual machine placement algorithms

To evaluate the performance of the optimization algorithms used in the virtual machines placement, certain metrics are applied to them. In this part of the work we discuss some of these metrics.

Energy consumption: To evaluate the energy consumption in the data centers, the increase due to the use of the CPU must be taken into account. The relationship below shows that servers shut down when they are not in use, which reduces power consumption.

$$P_j = \begin{cases} U_j^c \times (P_j^{busy} - P_j^{iddle}) + P_j^{iddle}, U_j^c > 0 \\ 0, \qquad otherwise \end{cases} \tag{5}$$

Service Level Agreement Violation: A breach of the SLAV will occur when the cloud service provider fails to provide the service expected by customers. The SLAV metric can be calculated as follows:

$$SLAV = SLATH \times PDM \tag{6}$$

The SLATH: This is the violation time for each active host. It is calculated as follows:

$$SLATH = \frac{1}{N} \sum \frac{T_{oi}}{T_{ai}} \tag{7}$$

N the number of hosts, T_{oi} is the total time host i used 100% of the CPU, which led to the violation SLAV, T_{ai} is the total duration of activity of host i for the machine virtual service[26]. The logic of SLATH is that virtual machines cannot get the required processor capacity due to 100% CPU usage of the active host.

Performance degradation due to migration (PDM) : Live migration of virtual machines has a negative impact on the performance of the applications running on them. It is calculated as follows:

$$PDM = \frac{1}{M} \sum \frac{C_{dj}}{C_{rj}} \tag{8}$$

The M represents the number of VMs; In this relation C_{dj} is the is the estimate of the degradation of the virtual machine due to the migration of the VM_j; And C_{rj} corresponds to the total CPU capacity requested by the VM_j during its lifetime.

4. Artificial intelligence for the optimization of virtual resources in the Cloud

In this part of the work, we analyze the different methods of artificial intelligence applied to the virtual machines placement. These can be directly applied to various virtual resource management techniques such as CPU usage, power consumption and migration. In [27], the authors have carried out a study of predictive techniques for virtual machine placement while specifying the different methods used. Other authors have shown the importance of these classification techniques for improving the overall efficiency and performance of data centers [28].

4.1. Artificial Neural network

ANNs are made up of layers of artificial neurons that process information using a transfer function and are interconnected on the basis of a specific network topology, each connection being associated with a weight and a bias. In the case of virtual machine placement, the average and maximum CPU usage, memory usage and other VM resources are defined as the input vectors. Each neural network with weighted connections receives these values from the hidden layer. As indicated in relation (9), the network will then calculate the sum of the weighted signals z_k for each neural network :

$$z_k = \sum_{j=1}^{m} w_{kj} x_j \tag{9}$$

where $x_1, x_2, ..., x_n$ are the input signals $w_{k1}, w_{k2}, ..., w_{km}$ are the respective synaptic weights of the neuron k ; z_k is the output of the linear combiner due to the signal input ; In our case the activation function is applied to the output signal of each neuron which transforms the range of the signal into a value between 0 and 1. The most commonly used activation function is a sigmoid function which is defied in the equation (10) :

$$\gamma(z_k) = \frac{1}{1 + exp^{-az_k}} \tag{10}$$

where a is the slope parameter of the function, z_k is the activation value for the k neuron and $exp()$ is the exponential function. In [28], the authors show that for the case of classification problems, the network generates a probability whose input characteristics are between A and B when the signal is propagated in the network.

4.2. Support Vector Machines

SVM is a machine learning algorithm that is generally used for classification but it can also be used for regression. SVMs are the data points closest to the hyperplane. Data points in a data set which, if deleted, would change the position of the hyperplane in division and which in this case can be considered as preponderant elements of a data set. Like these other methods mentioned above, SVM improves the capacity for generalization by seeking to minimize the structural risk of the learning machine. The SVM is much more widely used when classifying small samples of data due to its fast training speed and high classification accuracy [29].
The SVM will separate the samples for a hyperplane for each binary problem. For the classifier to have a good classification accuracy, the SVM must obtain a suitable w and b. The mathematical formula (11) shows how to model the non-linear decision function:

$$h(x) = sgn\left(\sum_{i=1}^{i=1} \alpha_i y_i K(X_i, X) + b \right) \tag{11}$$

where k is a kernel function that is associated with the pair (X_i, X). The kernel that is most used is the Gaussian kernel:

$$K(X_i, X) = e^{-\frac{\|X_i, X\|^2}{2\sigma^2}} \tag{12}$$

4.3. K nearest neighbor (KNN)
K-NN is one of the methods used in machine learning. It is based exclusively on the choice of the classification metric, but it can also be used for regression metrics. To predict resource use, the closest neighbor k predicts the power used by each VM based on the similarity of characteristics and collects the training test mode [30]. To define the distance between the new value and the training value, the Euclidean distance is used, which is calculated as follows:

$$d(x, y) = \sqrt{\sum_{j=i}^{n} (X_j - y_j)^2} \tag{13}$$

4.4. Tree decisions algorithm
Tree decisions algorithm is a method used in supervised learning based on the use of the decision tree as a predictive model. It is an algorithm that uses both classification tasks and regression. If the sample is homogeneous, the samples belong to the same class. The impurity measures this homogeneity of the sample. Two measurements are often used to measure impurity namely[31]:
Entropy: This refers to the amount of information needed to accurately describe certain samples. If the entropy is equal to 0 the sample is homogeneous which means that the elements are similar. And if the entropy is at the maximum so if it is equal to 1, it shows that all the probabilities p_i are equal. Mathematically, it is presented by the formula below:

$$E(S) = -\sum_{i=1}^{n} p_i \times log(p_i) \tag{14}$$

where $E(S)$ is the entropy of a data set, n represents the number of classes in the system and p_i represents the proportion of the number of instances that belong to class i.
Gini Index: Is used instead of entropy to measure impurity. It measures inequality in the sample. If the Gini index is 0, it means that the sample is perfectly homogeneous, and if it is 1, it means that the inequality is maximum between the elements. Its mathematical formula is:

$$I_G = 1 - \sum_{i=1}^{m} p_i^2 \tag{15}$$

4.5. Linear regression
The linear regression algorithm is one of the algorithms of supervised learning. A variable, named X, is considered as the predictive variable and a variable Y is considered as the target variable or the variable to be explained. For a simple linear regression, it is given by the following relation[32]:

$$Y = aX + b + \epsilon \tag{16}$$

Where Y is the target variable, dependent random, a and b are coefficients to be estimated, X is the independent explanatory variable and ϵ is a random variable that represents the error. Although the linear relationship is indeed present, the measured data do not exactly verify this relationship. In the mathematical model, we must take into account the observed errors noted ϵ

which is the difference between the observed value Y_i and the value $a_i X_i + b$ given by the linear relation.

4.6. Logistic regression

Logistic regression is a prediction algorithm used in classification. It is a linear classification model which is the counterpart of linear regression. The logistic function is defined by the relation[33]:

$$Y_i = \frac{1}{1 + e^{-k}} \tag{17}$$

where y_i is the predicted classification for independent variables $x_1, x_2, \ldots, x_m$. The value e is a constant and y is defined as the summation of each independent variable multiplied by it is corresponding parameter[28].

5. Placement of Virtual Network Functions

In this part of the work, we first establish the link between Cloud Computing and NVFs before proposing various related works on the placement of the functions of virtual networks in a Cloud environment.

5.1. Relationship between VNF and Cloud Computing

VNF is the rapidly growing approach of performing network functions to servers precisely from virtual machines. Its main objective is to allow VNFs to be deployed and scaled up without putting new equipment in place. Its main objective is to allow VNFs to be deployed and scaled up without putting in place new equipment. In [34], the authors show that VNFs are not limited to telecommunications networks but rather extend to computer applications that can create large-scale communications services. Using Fig. 4, they compare Cloud Computing and VNF based on the different layers of service embedded in the Cloud and the architecture of virtual networks. It shows the correspondence of the IaaS layer of Cloud Computing to the NFVI part of the VNF architecture as well as the SaaS layer to the different VNFs.

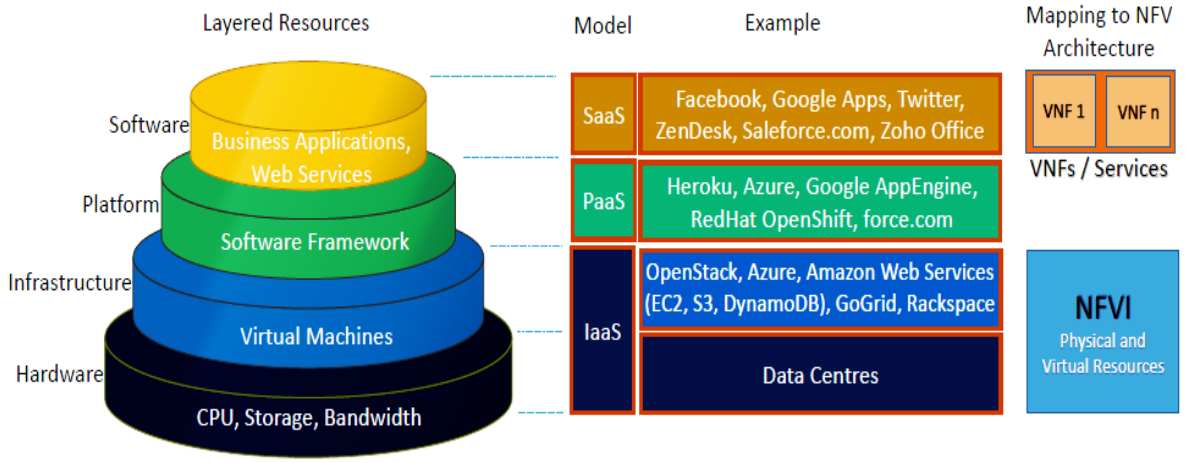

Figure 3. Comparison of the architecture of NVF with the Cloud Computing service layers

Undeniably, VNFs cannot be dissociated from SDNs (Soft Define Networking), which are also emerging as an alternative to existing networks, by splitting the data plane and the control plane. These two approaches, although distinct, often remain associated and complementary.

In SDN networks, routers are relegated to simple packet switching devices, and one or more controllers manage the control plane. This simple approach allows network administrators to gain better control over their network traffic. Indeed, SDN makes it possible to (i) collect metrology data of nodes and links; (ii) centrally manage these nodes, allowing and facilitating network optimization compared to distributed control. Research shows that these are areas that are strongly linked through virtualization.
The problem of optimal placement of VNF is widely addressed in several research studies. Several strategies are proposed to solve the various associated problems. Despite this, several questions related to this optimal investment persist. Recent efforts have been made to better introduce VNF and have resulted in the classification of VNF placement into two broad categories including general network functions placement (subdivided into VNF forwarding graph, VNF chains placement and VNF replication) and specific network functions placement (subdivided into transcoder and cache, S-GW and P-GW, vDPI and firewall and CDN).

5.2. Study on the VNFs placement

5.2.1. VNF Forwarding Graph

VNF-FG is a complex structure that providers the logical connectivity between virtual or physical network function and providers service chaining (the process of steering traffic flows across VNFs).
In [35], the authors presented a new approach for joint placement of virtalized network functions (VNFs) and their associated chains over Cloud environments. Our Eigen decomposition based solution scales to thousands of nodes and links and in insensitive to request size and the number of requested VNF-FGs. The algorithm has a fairly flat execution time penalty dominated by the physical infrastructure (NFVI) size.

5.2.2. VNF Chains Placement

The VNF Chain Placement Problem (VNF-CPP) is another VNF placement-related problem, which is known to be NP-Hard. It is important to find placement schemes that can scale with the size of the problem and find good quality solutions. In [36], they formalize the network function placement and chaining problem and propose an Integer Linear Programming (ILP) model to solve it. Additionally, in order to cope with large infrastructures, we propose a heuristic procedure for efficiently guiding the ILP solver towards feasible, near-optimal solutions.

5.2.3. VNF Repplication

In [37], they study the problem of VNF placement with replications, and especially the potential of VNFs replications to help load balance the network. In [38], the authors study the problem of VNF placement with replications, and especially the potential of VNFs replications to help load balance the network, while the server utilization is minimized. They present a Linear Programming (LP) model for the optimum placement of functions finding a trade-off between the minimization of two objectives: the link utilization and CPU resource usage. Carpio and Jukan study how to improve service reliability using jointly replications and migrations, considering the chaining problem inherent in NFV. While replications provide reliability, performing migrations to more reliable servers decreases the resource overhead. A Linear Programming (LP) model is presented to study the impact of active configurations on the network and server resources. Additionally, to provide a fast recovery from server failures, they consider N-to-N configurations in NFV networks and study its impact on server resources[38].

5.2.4. Transcoder and cachet

Nowadays, we find many streaming videos. The users challenge is to have a quick and non-interrupted access. In order to have a better QoS, the connection plays an important role.

Several elements can be taken into consideration to meet the challenge. The authors analyse some works studied on this subject. In [39] they laid out a network infrastructure that leveraged the storage and computing power of a cloud residing in the core for collecting network status and computing multi-path scalable video coding (SVC) streaming provisioning strategies.They show how OpenFlow is beneficial for optimizing the migration of transcoders. They show that not using this protocol would introduce an unacceptable interruption in the transmission of streaming[40].

5.2.5. S-GW and P-Gw Placement

Bagaa and al. have presented one of the important component of the carrier cloud vision, as to know Network Virtualization Function. They focused on the data anchor (PDN- GW) virtualization, and more specifically on how instantiating and assigning the virtual PDN-GW to UEs. Rather than using only geographical location, they proposed to consider applications/services type when selecting a PDN-GW to UEs[41]. In [42], they tackled the challenging problem of NF placement onto the cellular core. In this respect, we introduced a MILP and its relaxed variant for NF placement optimization, subject to capacity constraints, delay budgets between EPC components, and 3GPP-related restrictions. They further presented a greedy algorithm that strives to map NFs proximately to eNBs, inline with carriers common practice.

5.2.6. vDPI and Firewall

The authors formulate the vDPI placement problem as a cost minimization problem. They cast the problem as a multi- commodity flow problem. They then propose a centrality-based greedy algorithm and assess its validity by comparing it with the ILP optimal solution on random networks[43]. The authors tackle the VNFP problem with a focus on energy efficiency. Her work comprises developing an integer linear programming (ILP) formulation for the VNFP problem (subject to strict security and budgetary constraints) to minimize the energy consumption of servers, as well as implementation and evaluation of the model. They also showed that the security of the functions of virtual networks extends over several intrusion mechanisms[44].

5.2.7. CDN (Replication of Content Distribution Networks)

In [45], the study focused on virtual machine placement and flavors selection for different images in a CDNaaS platform. The platform manages a high number of videos deploying virtualized caches, transcoders, and streamers. A CDN slice owner can add videos specifying their resolutions and these videos are streamed to the end-users, consumers of the CDN slice. To create an efficient CDN slice, virtual machines hosting cache functions, transcoder functions, and streamer functions must be assigned adequate flavors and must instantiated at adequate locations. Similarly, the total cost to be paid by the CDN slice owner must be efficient and fair. They purpose, two solutions. The first one aims at minimizing the incurred total cost, where as the second one aims at maximizing QoE. In [46], the authors have devised mechanisms for allocating an appropriate set of VNFs for each CDN slice to meet its performance requirements and minimize as much as possible the incurred cost in terms of allocated virtual resources. A mathematical model is developed to evaluate the performance of the proposed mechanisms. They first formulate the VNF placement problem as two Linear Integer problem models, aiming at minimizing the cost and maximizing the quality of experience (QoE) of the virtual streaming service. By applying the bargaining game theory, they ensure an optimal trade off solution between the cost efficiency and QoE.

6. Conclusion

In this article, we present an overview of the different methods of optimizing the investment of resources in the Cloud. First, we introduce the different works on virtual machine placement. Next, we presented the different algorithms used in the placement of virtual machines and the metrics applied to evaluate them. Thereafter, we presented different machine learning (ML) techniques used to predict the use of resources in virtual resource placement. Then, we classified the various VNF works into two main categories. In future work, we will implement these different algorithms applied in optimization. This will allow us to carry out a detailed comparative study in order to get the best out of them. In the same work, we will then propose and implement a new algorithm and compare it with the best algorithm we have found.

References

[1] A. Beloglazov and R. Buyya, *Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers*, Concurr. Comput. Pract. Exp., vol. 24(2012), pp. 1397–1420.

[2] W. Song, Z. Xiao, S. Member, Q. Chen, and H. Luo, *Adaptive Resource Provisioning for the Cloud Using Online Bin Packing*, vol. 63(2014), no. 11, pp. 2647–2660.

[3] Janani, N.; Shiva Jegan, R.D.; Dr. Prakash P, *Optimization of Virtual Machine Placement in Cloud Environment Using Genetic*, Research Journal of Applied Sciences, Engineering and Technology , Amrita Vishwa," vol. 10(2015), pp. 274–287, 2015.

[4] A. Mosa and R. Sakellariou, *Virtual Machine Consolidation for Cloud Data Centers Using Parameter-Based Adaptive Allocation*,In Proceedings of the Fifth European Conference on the Engineering of Computer-Based Systems (ECBS '17). Association for Computing Machinery, Article 16(2017), 1–10.

[5] C. Ghribi, M. Hadji, and D. Zeghlache, *Energy efficient VM scheduling for cloud data centers: Exact allocation and migration algorithms*, Proc. - 13th IEEE/ACM Int. Symp. Clust. Cloud, Grid Comput. CCGrid 2013, vol.(2013) pp. 671–678.

[6] A. Abdelsamea, A. A. El-moursy, E. E. Hemayed, and H. Eldeeb, *Virtual machine consolidation enhancement using hybrid regression algorithms*,"Egypt. Informatics J., vol. 18(2017), pp. 161–170.

[7] M. Alaul, H. Monil, and R. M. Rahman, *VM consolidation approach based on heuristics , fuzzy logic , and migration control*," J. Cloud Comput., 2016.

[8] M. R. Chowdhury, M. R. Mahmud, and R. M. Rahman, *Implementation and performance analysis of various VM placement strategies in CloudSim*, J. Cloud Comput., vol. 4(2015), pp. 1–21.

[9] C. Wei, Z. H. Hu, and Y. G. Wang, *Exact algorithms for energy-efficient virtual machine placement in data centers*," Futur. Gener. Comput. Syst., vol. 106(2020), pp. 77–91.

[10] Y. Gao, H. Guan, Z. Qi, Y. Hou, and L. Liu, *Journal of Computer and System Sciences A multi-objective ant colony system algorithm for virtual machine placement in cloud computing*," J. Comput. Syst. Sci., vol. 79(2013), pp. 1230–1242.

[11] A. Abdelsamea, A. A. El-Moursy, E. E. Hemayed, and H. Eldeeb,*Virtual machine consolidation enhancement using hybrid regression algorithms Virtual machine consolidation enhancement*,Egypt. Informatics J., vol. 18(2017), pp. 161–170.

[12] C. Ghribi, M. Hadji, and D. Zeghlache, *Energy efficient VM scheduling for cloud data centers: Exact allocation and migration algorithms*,Proc. - 13th IEEE/ACM Int. Symp. Clust. Cloud, Grid Comput. CCGrid 2013, vol.(2013) pp. 671–678.

[13] H. Goudarzi and M. Pedram,*Energy-efficient virtual machine replication and placement in a cloud computing system*, Proc. - 2012 IEEE 5th Int. Conf. Cloud Comput. CLOUD 2012,vol.(2012) pp. 750–757.

[14] A. C. Adamuthe and T. R. Nitave, *Adaptive harmony search for optimizing constrained Resource Allocation problem*, Int. J. Comput., vol. 17(2018), pp. 260–269.

[15] S. Jangiti, V. Vijayakumar, and V. Subramaniyaswamy, *Hybrid best-fit heuristic for energy efficient virtual machine placement in cloud data centers*, EAI Endorsed Trans. Energy Web, vol. 7(2020), pp. 1–8.

[16] C. Lin, P. Liu and J. Wu, *Energy-efficient Virtual Machine Provision Algorithms for Cloud Systems*, 2011 Fourth IEEE International Conference on Utility and Cloud Computing, Victoria, NSW,vol.(2011), pp. 81-88.

[17] Z. Tang, Y. Mo, K. Li, and K. Li, *Dynamic forecast scheduling algorithm for virtual machine placement in cloud computing environment*, J. Supercomput., vol. 70(2014), pp. 1279–1296.

[18] S. Jangiti and S. Sriram. V.S., *Scalable and direct vector bin-packing heuristic based on residual resource ratios for virtual machine placement in cloud data centers*, Comput. Electr. Eng., vol. 68(2018), pp. 44–61.

[19] S. Chaisiri, B. Lee, and D. Niyato, *Optimal Virtual Machine Placement across Multiple Cloud Providers*, in Proceedings of IEEE Asia-Pacific services Computing Conference, vol.(2009), pp. 103-110.
[20] J. Zhou, Y. Zhang, L. Sun, S. Zhuang, C. Tang, and J. Sun, *Stochastic Virtual Machine Placement for Cloud Data Centers under Resource Requirement Variations*,IEEE Access, vol. 7(2019), pp. 174412–174424.
[21] A. S. Abohamama and E. Hamouda, *A hybrid energy–Aware virtual machine placement algorithm for cloud environments*, Expert Syst. Appl., vol. 150(2020), p. 113306.
[22] M. Tang and S. Pan,*A Hybrid Genetic Algorithm for the Energy-Efficient Virtual Machine Placement Problem in Data Centers*, Neural Process. Lett., vol. 41(2015), pp. 211–221.
[23] Mosa, A., Paton, N.W., *Optimizing virtual machine placement for energy and SLA in clouds using utility functions*. J Cloud Comp 5, 17 (2016). https://doi.org/10.1186/s13677-016-0067-7
[24] S. Kim and Y. ri Choi, *"Constraint-aware VM placement in heterogeneous computing clusters*, Cluster Comput., vol. 23(2020), pp. 71–85.
[25] A. E. A. Eltraify, S. H. Mohamed, and J. M. H. Elmirghani, *VM placement over WDM-TDM AWGR PON Based Data Centre Architecture.*, arXiv e-prints, pages = arXiv:2005.03590,2020.
[26] P. D. Bharathi, P. Prakash, and M. V. K. Kiran, *Virtual machine placement strategies in cloud computing*, 2017 Innov. Power Adv. Comput. Technol. i-PACT 2017, vol. (2017)l, pp. 1–7.
[27] Masdari, M., Zangakani, *M. Green Cloud Computing Using Proactive Virtual Machine Placement: Challenges and Issues*, J Grid Computing (2019). https://doi.org/10.1007/s10723-019-09489-9
[28] R. Shaw, E. Howley, and E. Barrett, *"An intelligent ensemble learning approach for energy efficient and interference aware dynamic virtual machine consolidation*, Simul. Model. Pract. Theory, vol.(2019), p. 101992.
[29] G. Liu et al., *Predicting cervical hyperextension injury: A covariance guided sine cosine support vector machine*, IEEE Access, vol. 8(2020), pp. 46895–46908.
[30] T. Deepika and P. Prakash, *Power consumption prediction in cloud data center using machine learning*, Int. J. Electr. Comput. Eng., vol. 10(2020), pp. 1524–1532.
[31] D. M. Manias, M. Jammal, H. Hawilo, A. Shami, P. Heidari, and A. Larabi, *Machine Learning for Performance-Aware Virtual Network Function Placement*, IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA,vol. (2019), pp. 1-6.
[32] L. Li, J. Dong, D. Zuo, and J. Wu, *SLA-Aware and Energy-Efficient VM Consolidation in Cloud Data Centers Using Robust Linear Regression Prediction Model*, IEEE Access, vol. 7(2019), pp. 9490–9500 .
[33] Y. Jararweh, M. B. Issa, M. Daraghmeh, M. Al-Ayyoub, and M. A. Alsmirat, *Energy efficient dynamic resource management in cloud computing based on logistic regression model and median absolute deviation*, Sustain. Comput. Informatics Syst., vol. 19(2018), pp. 262–274.
[34] R. Mijumbi, J. Serrat, J. L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, *Network function virtualization: State-of-the-art and research challenges*, IEEE Commun. Surv. Tutorials, vol. 18, no. 1, pp. 236–262, 2016.
[35] M. Mechtri, C. Ghribi, and D. Zeghlache, *VNF placement and chaining in distributed cloud*," IEEE Int. Conf. Cloud Comput. CLOUD, pp. 376–383, 2017.
[36] M. C. Luizelli, L. R. Bays, L. S. Buriol, M. P. Barcellos, and L. P. Gaspary,*Piecing together the NFV provisioning puzzle: Efficient placement and chaining of virtual network functions*, Proc. 2015 IFIP/IEEE Int. Symp. Integr. Netw. Manag. IM 2015, vol.(2015)pp. 98–106.
[37] F. Carpio, S. Dhahri, and A. Jukan, *VNF placement with replication for Loac balancing in NFV networks*, 2017 IEEE International Conference on Communications (ICC), vol.(2017), pp. 1–6.
[38] F. Carpio, W. Bziuk, and A. Jukan, *Replication of Virtual Network Functions: Optimizing link utilization and resource costs*, 2017 40th Int. Conv. Inf. Commun. Technol. Electron. Microelectron. MIPRO 2017 - Proc., vol.(2017)pp. 521–526.
[39] Z. Zhu, S. Li, and X. Chen, *"Design QoS-aware multi-path provisioning strategies for efficient cloud-assisted SVC video streaming to heterogeneous clients*,IEEE Trans. Multimed., vol. 15(2013), pp. 758–768.
[40] P. Farrow, M. Reed, M. Glowiak, and J. Mambretti, *Transcoder Migration For Real Time Video Streaming Systems*, pp. 1–13, 2015.
[41] M. Bagaa, T. Taleb, and A. Ksentini, *Service-aware network function placement for efficient traffic handling in carrier cloud*, IEEE Wirel. Commun. Netw. Conf. WCNC, vol. 3(2014), pp. 2402–2407.
[42] D. Dietrich, C. Papagianni, P. Papadimitriou, and J. S. Baras, *Network function placement on virtualized cellular cores*, 2017 9th Int. Conf. Commun. Syst. Networks, COMSNETS 2017, vol. (2017), pp. 259–266.
[43] M. Bouet, J. Leguay and V. Conan, *Cost-based placement of vDPI functions in NFV infrastructures*, Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft), vol.(2015), pp. 1-9.
[44] I. P. Bolodurina and D. I. Parfenov,*Neural network model for optimize network work in the infrastructure of the virtual data center*,2017 25th Telecommun. Forum, TELFOR 2017 - Proc., vol. (2017), pp. 1–4, 2018.

[45] S. Retal, M. Bagaa, T. Taleb and H. Flinck, *Content delivery network slicing: QoE and cost awareness*, 2017 IEEE International Conference on Communications (ICC), Paris, vo.l(2017), pp. 1-6.
[46] I. Benkacem, T. Taleb, M. Bagaa, and H. Flinck, *Optimal VNFs Placement in CDN Slicing over Multi-Cloud Environment*, IEEE J. Sel. Areas Commun., vol. 36(2018), pp. 616–627.

Resource Optimisation in Cloud Computing: Comparative Study of Algorithms Applied to Recommendations in a Big Data Analysis Architecture

Submitted: 18th April 2021; accepted: 9th February 2022

Aristide Ndayikengurukiye, Abderrahmane Ez-Zahout, Akou Aboubakr, Youssef Charkaoui, Omary Fouzia

DOI: 10.14313/JAMRIS/4-2021/28

Abstract:
Recommender systems (RS) have emerged as a means of providing relevant content to users, whether in social networking, health, education, or elections. Furthermore, with the rapid development of cloud computing, Big Data, and the Internet of Things (IoT), the component of all this is that elections are controlled by open and accountable, neutral, and autonomous election management bodies. The use of technology in voting procedures can make them faster, more efficient, and less susceptible to security breaches. Technology can ensure the security of every vote, better and faster automatic counting and tallying, and much greater accuracy. The election data were combined by different websites and applications. In addition, it was interpreted using many recommendation algorithms such as Machine Learning Algorithms, Vector Representation Algorithms, Latent Factor Model Algorithms, and Neighbourhood Methods and shared with the election management bodies to provide appropriate recommendations. In this paper, we conduct a comparative study of the algorithms applied in the recommendations of Big Data architectures. The results show us that the K-NN model works best with an accuracy of 96%. In addition, we provided the best recommendation system is the hybrid recommendation combined by content-based filtering and collaborative filtering uses similarities between users and items.

Keywords: *Cloud computing, Big Data, IoT, Recommender system and KNN algorithm*

1. Introduction

Nowadays, many countries are trying to put in place laws to make it easier for the public to express their opinion on a given issue and especially to make a choice of their leaders through voting. The conventional procedure is manual or paper voting which sometimes does not facilitate counting and is not convenient for voters which sometimes decreases the voting rate. With the evolution of information and communication technologies, electronic voting has become a necessity to overcome these problems and facilitate voting for any geographical area to participate.

This technological evolution has generated an explosion of a large mass of digital data, which makes it imperative to find solutions to enable adequate storage and analysis, thus giving rise to the concept of Big Data. Indeed, Big Data requires enormous storage and processing capacities that conventional methods cannot offer. To promote the deployment of Big Data comes the concept of Cloud Computing which allows access and use of computer resources such as storage and network on demand. In other words, it provides users with computing services that allow them to save and manipulate data, install models and access multiple resources on the web, according to their needs. One of the most advantageous features of the cloud is its elasticity, as it allows computing capacity to be adapted to demand and only the necessary resources to be allocated [1]: The most popular cloud-based services include Amazon web services, SAP (System Applications and Products), Oracle, Cloudera, Databricks, etc.

With this rapid increase in the volume of digital data on websites and applications worldwide has led to the problem of data overload which makes it difficult to extract relevant information. As a response to this problem, optimisation algorithms and recommender systems have been proposed and these various services mentioned above serve as a platform for the development and evaluation of recommender systems.

When search engines are not sufficient to process this large amount of online information, recommender systems try to guess what information or goods and services we really need. Any recommender system needs to know a user profile that, for example, contains the user's preferences. This profile can be acquired implicitly by analyzing the user's past behavior, or explicitly by asking the user about their preferences. In this case, recommender systems that take into account a community of users are often called collaborative approaches [2], while those based on the description of articles are called content-based approaches [3]. The hybrid approach [4] is a combination of two different methodologies or systems that aims to create a new model and is strongly rooted in information retrieval and information filtering. User preferences and article descriptions are somehow compatible so that it is possible to compare them and find an article that matches the user's preferences.

One of the central issues in recommender systems is how to model and then compare user preferences and item descriptions in a flexible and natural way. Obviously, user preferences, as well as item descriptions, can be imprecise, incomplete, subjective, and of

different importance. The first approach is based on the use of the fuzzy set theory adopted by Yager [5]. Another approach, which uses fuzzy clustering as a tool for a recommendation, was presented in [6].

Throughout political elections, voters have to make a decision about who to vote for. This decision has become particularly difficult in recent times when much more attention is paid to personalities, images, and campaign events than to party manifestos and policy issues. In addition, various applications have been created to help voters easily pick their favorite candidates. The general idea of these systems is to put x on the symbols of the two candidates. Finally, these choices are compared to select the candidates that are accessible.

In this paper, we present an overview of the different algorithms used in recommender systems and the different tools used in Big Data analysis. We also make a comparative study of some of these algorithms in recommender systems applied to a vote.

The rest of this article is organized as follows. Section 2 presents the different tools used in Big Data analysis and a brief overview of recommender systems. Section 3 presents some of the work done on this topic. Section 4 presents the methodology used in our work. Section 5 presents the results obtained in this article. We end with a conclusion.

2. Literature Review

2.1. Big Data Framework

HDFS. Hadoop Distributed File System (HDFS) is an architecture that consists of NameNode which is a server that regulates file access by clients and manages the file system namespace. HDFS allows user data to be stored in files through a file namespace. In HDFS the opening, closing, and renaming of files and directories are operations performed by the NameNode and also determines the mapping of blocks to Datanodes [4].

HDFS has the ability to scale to applications with datasets ranging in size from a few gigabytes to a few terabytes. It can also scale to hundreds of nodes in a single cluster and can provide high bandwidth for aggregated data [7].

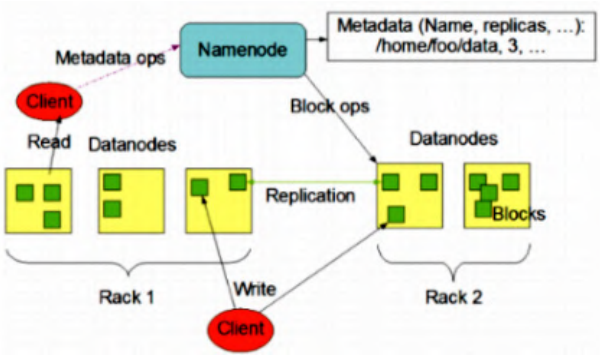

Fig. 1. Architecture of HDFS

Spark. In advanced data analysis in Big Data, Spark is the most widely used in-memory distributed processing framework. The reasons for its success include its Application Programming Interface (API) and a set of features that allow querying large amounts of data using Structured Query Language (SQL) to distill complex Machine Learning (ML) models using the most popular algorithms. Spark also reduces the execution time of an application by reducing the number of read/write operations on the disk [5].

Despite these advantages, it can mask some of the problems caused by a distributed code execution mechanism, making testing inefficient. Creating Dockers containers is the most efficient way to create a test environment that resembles the cluster environment. The advantage of this approach is that applications can be tested on different versions of the Framework by simply changing some parameters in the configuration file [6].

Another advantage is that it is designed to be run on different types of clusters. Cluster managers such as Yet Another Resource Negotiator(YARN), the Hadoop Platform Resource Manager, Mesos, or Kubernetes are supported by a spark [4]. The figure shows the architecture of HDFS [3].

MapReduce. The huge amount of data provides companies with many opportunities. One of the biggest problems is to be able to process it efficiently on traditional systems and therefore to turn to new solutions specifically designed for this purpose.

MapReduce makes data processing easier. It breaks down massive volumes of data into smaller chunks. These pieces of data are then processed in parallel on hadoop servers. Afterward, it sends the consolidated result to the application [8].

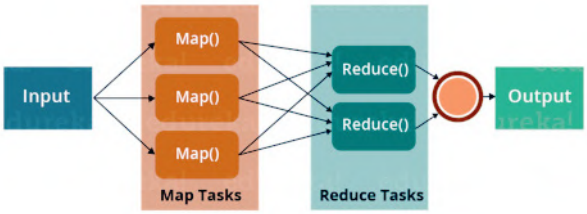

Fig. 2. Execution of MapReduce Job

2.2. Recommender Systems

With the evolution of technology and especially the advent of the web, the amount of data to be exploited and processed has become very large, making it difficult to search and find. One of the computer techniques that has been developed to extract relevant information is recommendation systems. They allow the user to be guided through the data in order to find the most relevant data.

In RS, the usefulness of the item is usually represented by a score that indicates how a user liked a particular item. The problem that arises is to solve the estimation of scores for items that have not been evaluated by the user. Whenever it is possible to estimate scores for items that have not yet been evaluated, users are recommended items with a higher estimated score [9].

RS are systems that produce personalised recommendations with the aim of guiding the user in a personalised way towards interesting or useful elements when many options are available. They have three phases: the first is the data collection phase, which consists of collecting relevant information about a user; the second is the learning phase, which uses the collected data to filter and exploit the user's features; and the final phase is the recommendation phase, which predicts or recommends the type of product that the user may prefer.

Collaborative filtering. This type of recommendation system aims to automate the recommendations that users sharing the same interests can make to each other. They allow a user to find the information that interests him/her based on the same tastes and preferences as other users. In this type of recommendation two methods can be used, namely user-based recommendation and item-based recommendation.

For example, the user-based recommendation is a technique used to predict which items a user might like based on the ratings given to that item by other users who have similar tastes to the target user. In this type of recommendation, the weight can be deduced as the correlation between users and , which is calculated by the following formula:

$$\boldsymbol{Sim(a,b)} = \frac{\sum_p (\boldsymbol{r_{ap}} - \bar{\boldsymbol{r}}_a)(\boldsymbol{r_{ab}} - \bar{\boldsymbol{r}}_b)}{\sqrt{\sum (\boldsymbol{r_{ap}} - \bar{\boldsymbol{r}}_a)^2}\sqrt{\sum (\boldsymbol{r_{bp}} - \bar{\boldsymbol{r}}_b)^2}} \quad (1)$$

In some cases, a user may have similarities to some users and few similarities to others. This means that the ratings given to a particular item by the most similar users should be given more weight than those given by less similar users, and so on. In this case, each user's rating is multiplied by a similarity factor, which is given by the formula:

$$\boldsymbol{r_{up}} = \bar{\boldsymbol{r}}_b + \frac{\sum_{i \in users} \boldsymbol{sim(u,i)} * \bar{\boldsymbol{r}}_{ip}}{\sum_{i \in users} | \boldsymbol{sim(u,i)} |} \quad (2)$$

Content-based filtering. To generate recommendations, these types of RS rely on user profiles and item descriptions while suggesting to the user items that best match their preferences based on similarity. In fact, profiles are built for users as well as for products. Thus, these descriptions (profiles, users) are represented as vectors of weighted terms. The formula below represents the calculation of similarities to predict user interests [10][11].

$$\boldsymbol{Sim(a,b)} = \boldsymbol{cos\theta} = \frac{\vec{\boldsymbol{a}}.\vec{\boldsymbol{b}}}{\| \vec{\boldsymbol{a}} \|.\| \vec{\boldsymbol{b}} \|} = \sum_{i=1}^{n} \frac{\boldsymbol{a_i b_i}}{\sqrt{\boldsymbol{a_i^2}.\boldsymbol{b_i^2}}} \quad (3)$$

Some authors in [12] have used probabilistic methods in content-based recommendations. The formula below represents the Bayesian classifier model checking the membership of an example a to a class taking into account the attributes.

$$\boldsymbol{P(C_i)} \prod_k \boldsymbol{P(A_k V_{kj} C_i)} \quad (4)$$

The other method that is used in this type of recommendation is the Term Frequency-Inverse Document Frequency (TF-IDF), which makes it possible to determine in what proportions certain words in a website, for example, or in a document, are evaluated in relation to the rest of the text. For example, the frequency criterion for increasing the relevance of a site, without the keyword density playing a major role. The TF is defined as follows[13]:

$$TF = \frac{Numberofoccurrencesofterm}{Totalnumberofterms} \quad (5)$$

The other measure will focus on measuring the word as a function of document usage rather than measuring it as a function of frequency. The formula below defines the IDF[13]:

$$\boldsymbol{IDF} = \boldsymbol{log} \frac{Totalnumberofdocuments}{Numberofdocumentscontainingterm} \quad (6)$$

The following formula calculates the TF weight:

$$W = IDF * TF \quad (7)$$

Hybrid recommender systems. Hybrid systems combine two or more techniques to achieve better performance, such as content-based filtering and collaborative filtering. The limitations of one technique can be overcome by another technique

Today, recommender systems have been studied in several areas such as smart cities[14], education[15], e-commerce[16], e-learning [17].

2. Related Works

In the past, many researchers have devoted much of their work to the development of electoral systems or IoT-based technologies. Previous work was mainly focused on a novel IoT recommender system that uses publicly available application data as a source for personalization. In [8], the authors proposed a system that infers the user's physical objects from the exploration of applications installed on their mobile devices such as smartphones and tablets and builds a digital inventory of each user's physical objects. These inventories can then be used to create personalized recommendations. In [18], the authors used recommendation systems for IoT users. In their work, they proposed a recommender system that works on the basis of the graph created between users, objects, and services while recommending an IoT device that is suitable for the customers based on their needs and interests. To recommend the best option to the users, they also took into account the characteristics of users and services. To cope with the difficult task of selecting services in cloud computing, the authors

in [19]. proposed a recommendation system for these services to help different cloud users better find what they are looking for. To generate reliable recommendations, they adopted the formal fuzzy concept analysis using the lattice representation. This method allowed them to transform the repository of different cloud services into a set of small clusters, where relationships between high-quality services and high-quality services are highlighted.

3. Our Methods

The process of designing the recommendation system with the progression of Internet of Things (IoT), this case of cloud computing is evolving rapidly using clusters. He gives elasticity to users of associated information technology (IT) services to save and manage data, install models and easily provide recommendations as and when the best needs. On the other hand, we have proposed a hybrid recommendation approach to the context to get the candidates are selected.

Our approach combines two different methods: content filtering and collaborative filtering. Each method was adapted to a stage-specific to the voter. First, the approach contained attempts to create a profile used to predict ratings on unseen items. Secondly, the collaborative approach uses the products using user notes (explicit or implicit) from the given history. It works in a database of user preferences for the items (Items) that are similar. Finally, we propose a more efficient recommendation system that is based on the hybrid recommendation to put to eliminate the inconvenience in our born gift. Fig. 3. shows the methodology of our work.

Furthermore, we have implemented fourteen algorithms, but the best respectively of them are: K-NN, K-NN with GridSearchCV, SVM, and Decision Tree (DT) Classifier of precisions: 96%, 96%, 92% and 89%, such that the procedures of each algorithm is written:

Algorithm K-NN. The K-Nearest Neighbors (K-NN) algorithm is a machine learning algorithm that belongs to the class of simple and easy to implement supervised learning algorithms that can be used to solve classification problems and regression.

Algorithm: K-NN

Input: the training set D, test objet x, category label set C.
Output: the category c(x) of test object x, c(x) belongs to C.

1. Begin
2. For each y belongs to D calculate the distance D(y,x) between y and z. end for
3. Select the subset N from the data set D, the N contains k training samples which are the k narest neighbors of the test sample x.
4. Calculate the category of x: $c_k = \arg max_{c \in C} \sum_{y \in N} \mathbf{1}(c = class(y))$
5. End

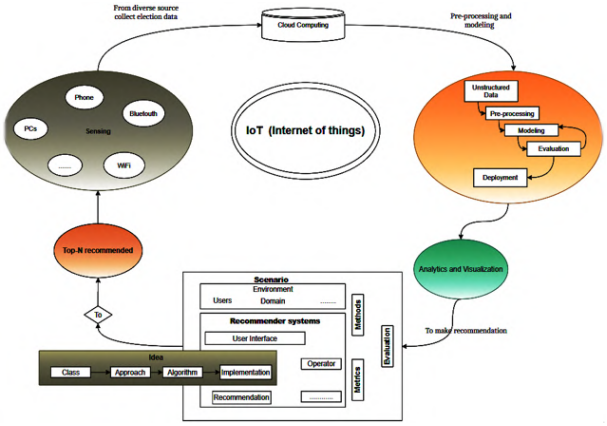

Fig. 3. Proposed analysis architecture

Algorithm SVM. SVM (Support Vector Machine) is supervised learning model with associated learning algorithms that analyze data for classification and regression analysis.

Algorithm: SVM

Input: Determine the various training and test data
Output: Determine the calculate accuracy
Select the optimal value of cost and gamma for SVM
While (stopping condition is not met) do

1. Implement SVM train step for each data point.
2. Implement SVM classify for testing data point.

End while

Return accuracy

Algorithm Decision Tree Classifier. The decision tree algorithm belongs to the family of supervised machine learning algorithms

Algorithm: Decision Tree

Input: S, where S= set of classified instances
Output: Decision Tree
Require: $S \neq \emptyset$, $num\_attibutes > 0$

1. **Procedure** Build Tree
2. **Repeat**
3. **maxGain ← 0**
4. **splitA ← null**
5. **e ← Entropy(Attributes)**
6. **For** all Attributes a in S **do**
7. **gain ← InformationGain(a,e)**
8. **If gain > maxGain then**
9. **maxGain ← gain**
10. **splitA ← a**
11. **End if**
12. **End for**
13. Partition(S,splitA)
14. **Until** all partitions processed
15. **End procedure**

We evaluated and compared the performance of several recommendation algorithms. We conducted an experimental study on Databricks and Jupyter notebook, a publicly available election dataset, to analyse

and evaluate several recommendation algorithms, namely: Term frequency-inverse document frequency [20], Rocchio [21], latent semantic analysis [22], K-nearest neighbours [23], Naïve Bayes [24], decision tree classifier[25], support vector machine [26], Linear regression(LR) [27], K-means [28], Slope One [29], Non-matrix factorisation [30], Singular decomposition vector [10], Co-clustering [31], Pearson correlation [32] and Artificial neural network5(ANN) [33]. The evaluation of the algorithms is based on precision, recall, f1 score, accuracy, area under the curve, RMSE, MSE, and MAE.

Definitions:

1. True positive (tp) = number of instances correctly predicted as a vote.
2. False positives (fp) = number of instances incorrectly predicted as votes.
3. True negative (tn) = number of instances correctly predicted as non-voting.
4. False negative (fn) = number of instances incorrectly predicted as non-voting.

a) **Precission:** this is the measure of quality. It is the measure of how well the predicted positive observations match the total sum of the predicted positive observations. It answers the following question: Of all the instances labeled as votes, how many times did this turn out to be true? It can be calculated as shown in equation 1:

$$\text{Precission} = \frac{tp}{tp + fp} \tag{8}$$

b) **Recall:** this is the measure of completeness. It determines the ratio of adequately predicted positive observations to the set of observations in the defined class – yes. Otherwise, it answers the following question: Of all the voting predictions that are true, how many have we labeled? Equation 2 shows how to calculate this:

$$\text{Recall} = \frac{tp}{tp + fn} \tag{9}$$

c) **F-score:** is the weighted average of both precision and recall values. Hence, this score regards both false positives and false negatives. It can be calculated as given in equation 3:

$$\text{F} - \text{score} = 2 * \frac{\text{precission} * \text{recall}}{\text{precission} + \text{recall}} \tag{10}$$

d) **Accuracy:** is a ratio of fittingly predicted instances to the total instances. The accuracy of an algorithm is its capability to differentiate the voting and no-voting cases accurately. The accuracy of a system can be measured as given in equation 4:

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn} \tag{11}$$

e) **RMSE:** Root Mean Square Error is a measure of the deviation of predictions from the actual value (calculated as the square root of an average square) as given in equation 5:

$$\text{RMSE} = \sqrt{\frac{\sum_{c \in C} \sum_{i \in Itest_C} (reco(c,i) - r_{c,i})^2}{\sum_{c \in C} Itest_C}} \tag{12}$$

f) **MSE:** Mean Squared Error is the average squared between the estimated values and the actual value as given in equation 6:

$$\text{MSE} = \frac{1}{N} \sum_{i=1} (Y - \hat{Y})^2 \tag{13}$$

g) **MAE:** Mean Absolute Error is the average over the verification sample of the absolute values of the differences between forecast and the corresponding their direction, as given in equation 7:

$$\text{MAE} = \frac{\sum_{c \in C} \sum_{i \in Itest_c} | reco(c,i) - r_{c,i} |}{\sum_{c \in C} Itest_c} \tag{14}$$

h) **AUC:** specifies the degree to which a model is proficient at differentiating between the given classes. Higher the value of AUC, superior the model is at foreseeing 0s as 0s and 1s as 1s i.e., the model can accurately distinguish between voting and no-voting cases.

5. Results and Discussion

During our work we used two programming languages: Python and Apache Spark in Jupyter, Colab, and Cloud Databricks in an Acer Predator computer with the following performance: operating system (window 10 Home), Processor Intel R core i7, speed 2.60 GHz, part of display & graphics (Graphics controller manufacturer NVIDIA R, Model GForce R GTX 1660Ti, Memory Capacity UP to 6 GB) and storage 1 TB, Total hard drive capacity 1 TB, 16 Go RAM.

In this paper, we consider the hybrid approach the strongest recommender system for recommending a candidate to an elector.

In this project, we use data from this link (https://www.kaggle.com/unanimad/us-election-2020?select=governors_county_candidate.csv:shows all files used on format CSV: comma separated values) as test data. What distinguishes our system is the fact that we take into consideration the possibility that the candidate intentionally did not vote. Therefore, we have to find the highest count of voters to candidates using similar methods. More precisely, we will solve our problem by steps: import our data, pre-processing our data, also modeling, evaluations (like recall, precision, f1-measure, RMSE, MSE, MAE, AUC) and compare our results of algorithms for making the strongest recommendation.

From Table 1 and Table 2, it can be observed that K-Nearest Neighbors and SGDC with GridSearchCV have performed really well with an accuracy of 96% on 7-fold cross-validation. SVM follows next with 92% accuracy. Further, Rocchio's algorithm has a minimum accuracy of 66% as compared with other algorithms. Figures 1 to 12, depicts and compares the performance of the used algorithms on the basis of precision, recall, f1-score, RMSE, MSE, MAE, and AUC (Area under the ROC Curve evaluated) for the election dataset.

Based on the results, we can clearly observe that the K-NN and SGDC GridSearchCV outperformed all the other classifiers. In the future, we will use those classifiers in our smart Recommender system, where personalized recommendations would be generated to each user based on his condition voting or not voting.

Tab. 1. Comparison between the algorithms by the metrics: precision, recall, F1-score, and accuracy

Algorithm	Precision	Recall	Fi-score	Accuracy
Rocchio's	0.77	0.56	0.66	0.66
KNN	0.96	0.96	0.96	0.96
SGDC GridSearch CV	0.96	0.96	0.96	0.96
DTC	0.88	0.89	0.88	0.89
SVM	0.85	0.92	0.89	0.92

Tab. 2. This table shows the comparison between the algorithms by the metrics: MSE, RMSE, MAE, and AUC

Algorithm	MSE	RMSE	MAE	AUC
Rocchio's	22.893	4.784	2.17	0.96
KNN	1871268587.5	43258.16	565.19	1.0
SGDC GridSearch CV	195255897.58	12460.17	429.92	1.0
Naïve Bayes	231533.567	481.17	16.638	0.99
DTC	231514.49	481.15	15.831	0.89
SVM	0.85	0.92	0.89	0.92
LR	1.0	1.0	1.0	1.0
ANN	0.991	0.9995	0.01	0.5

Tab. 3. This table shows the comparison between the algorithms by the metrics: MSE, RMSE, MAE, and MAE

Algorithm	MSE	RMSE	MAE
Slope One	0.59	0.768	0.760
NMF	40110374.3	2002.608221	2002.56
SVD	4010369.24	2002.608225	2002.56
SVDpp	231533.567	481.17	16.638

According to Table 3, the slope one algorithm has the lowest errors in MSE, RMSE and MAE compared to all other algorithms followed by the SVDpp algorithm. It can be seen that the SVD algorithm is the one that admits colossal values in terms of errors, which induces slope one represents the best algorithm that better explains our variable which moreover presents an MSE of 0.59<1, RMSE of 0.768<1, and MAE= 0.760<1.

The rest of the curve is made up of the precision and recall values for the threshold values which are between 0 and 1. Our goal is to make the curve as close as (1,1), which means a good precision and a good recall which is in the first curve of Fig. 4. in class 11 of area 87%.

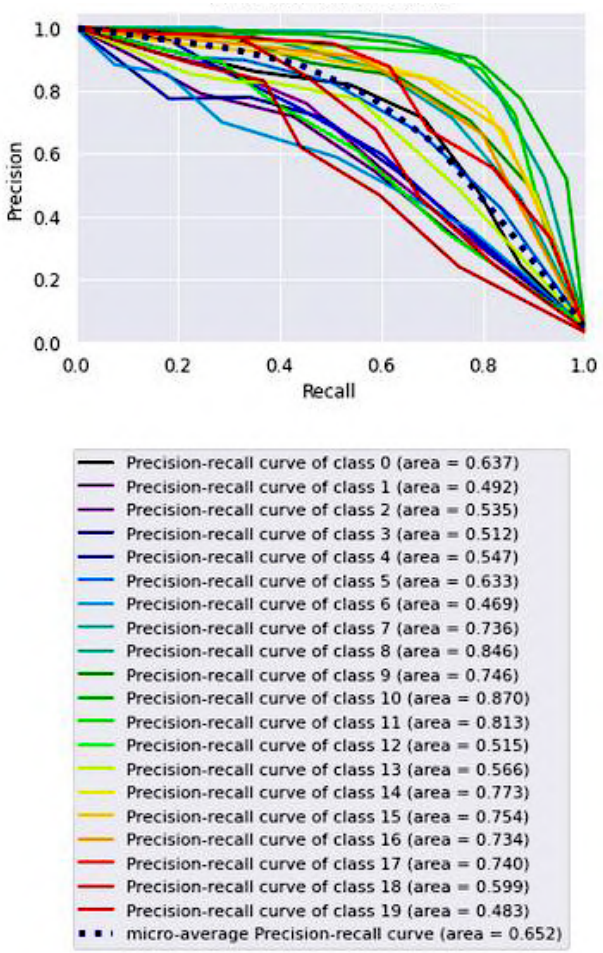

Fig. 4. ROC (Prediction Vs Test) of Rocchio's

In the Second curve Fig. 5., when 0.5 < AUC <1, there is a good chance that the classifier is able to distinguish positive class values from negative class values. This is because the classifier is able to detect more numbers of true positives and true negatives than false negatives and false positives. In our case, we have a multi-class binary classification problem using the technique One vs All. For this, the best class of ROC is Class 8 of the 96% area.

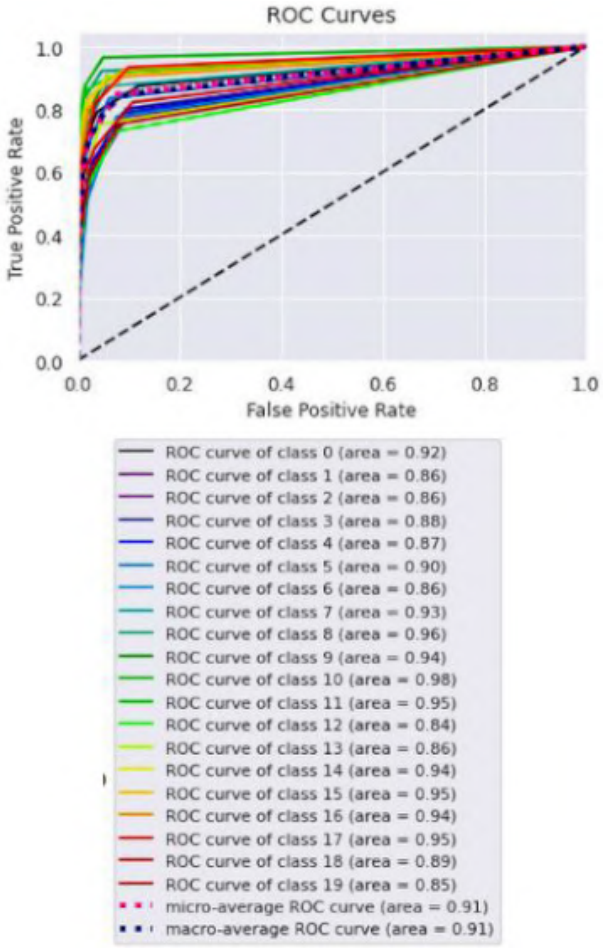

Fig. 5. ROC (Probability Vs Test) of Rocchio's

In Fig. 6., we have AUC = 1, then the classifier is able to distinguish perfectly all the positive and negative class points correctly. If, however, the AUC had been 0, then the classifier would have predicted all the negatives as positive and all the positives as negative.

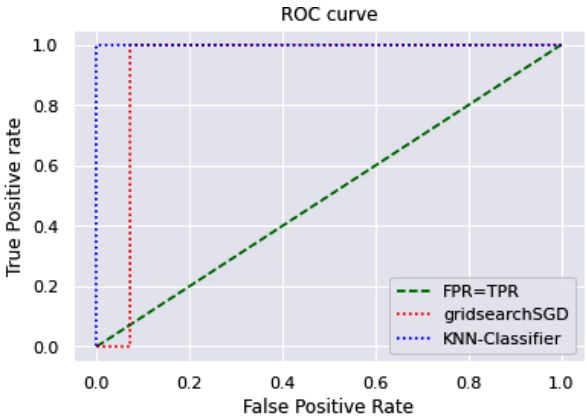

Fig. 6. ROC of SGDC GridSearchCV and KNN

In Fig. 7, we have AUC = 1, then the classifier is able to distinguish perfectly all the positive and negative class points correctly. If, however, the AUC had been 0, then the classifier would have predicted all negatives as positives and all positives as negatives.

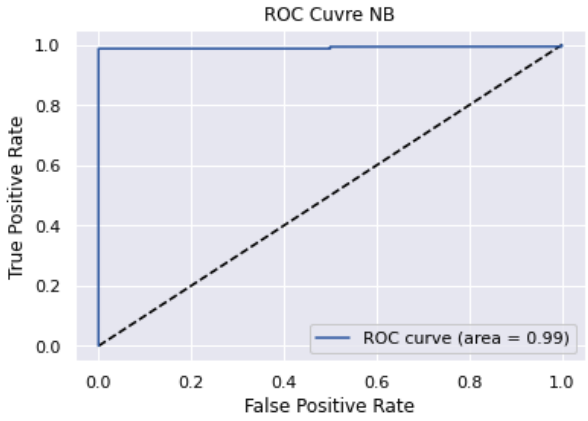

Fig. 7. ROC of NB

In Fig. 8., we have AUC = 1, then the classifier is able to distinguish perfectly all the positive and negative class points correctly. If, however, the AUC had been 0, then the classifier would have predicted all negatives as positives and all positives as negatives.

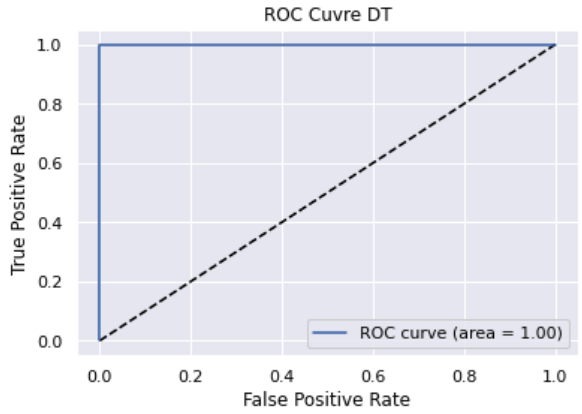

Fig. 8. ROC of DT

Fig. 9. shows that we can clearly see that the training score is always around the maximum of 1.0 and that the validation score, could be increased with more training samples of 0.88.

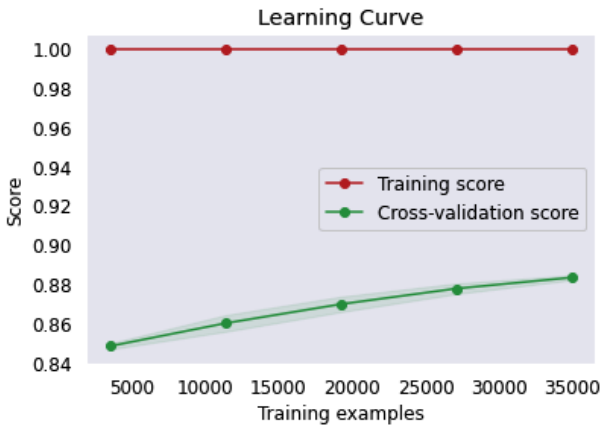

Fig. 9. Learning curve of DT

In Fig. 10., we can see clearly that the training score is still around the maximum of score 0.92 and the validation score could be increased with more training samples of the maximum score is 0.92.

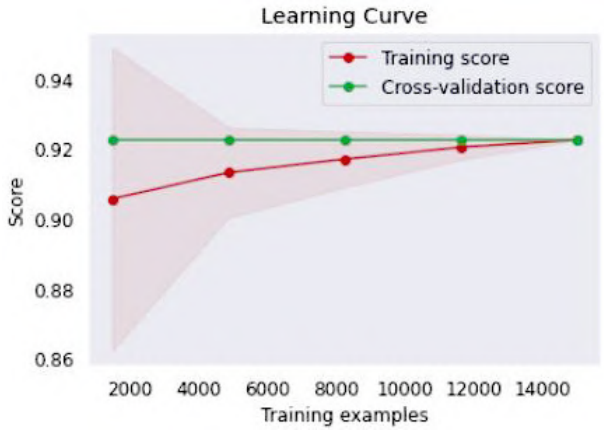

Fig. 10. Learning curve of SVM

Fig. 11. clearly shows that the training score is always around the maximum of 0.0 and that the validation score could be lowered with more training samples.

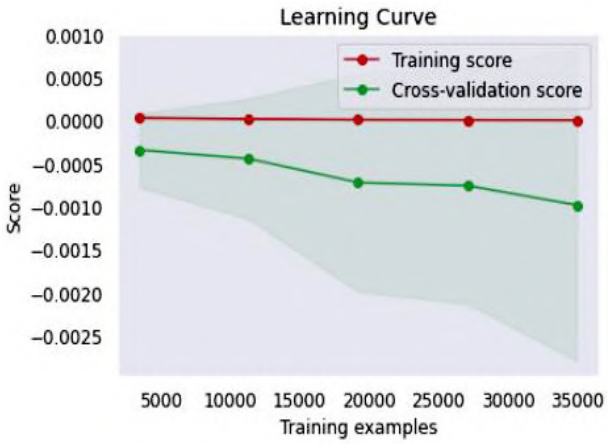

Fig. 11. Learning curve of LR

In Fig. 12., we use this curve to select the optimal number of clusters by fitting the model with a range of values for K in the K-means algorithm. We have shown a line drawn between SSE (Sum of Squared Errors) and the number of clusters to find the point representing the "elbow point" (this is a point after which the SSE or inertia starts to decrease linearly). The point on the SES / Inertia graph where the SES or inertia starts to decrease linearly is the elbow point. In this Fig. 12., it can be noted that it is at the number of clusters = 6 that the SSE starts to decrease linearly.

Fig. 12. Sum Squared Errors vs Number of clusters

In Fig. 13., we have AUC = 0.5, then the classifier is not able to distinguish between the Positive and Negative class points. This means that the classifier predicts a random class or a constant class for all data points.

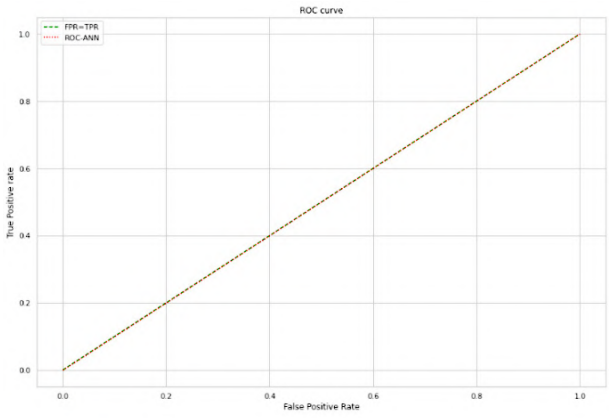

Fig. 13. ROC of ANN

Fig. 14., shows that the training score is still around the maximum score of 1.0 and the validation score could be increased with more training samples of the maximum score is 1.0.

In Fig. 15, we get the index of the first five products correlated with the product Id chosen by the user, then we see the ratings that the user could give to these 5 most correlated products by made.

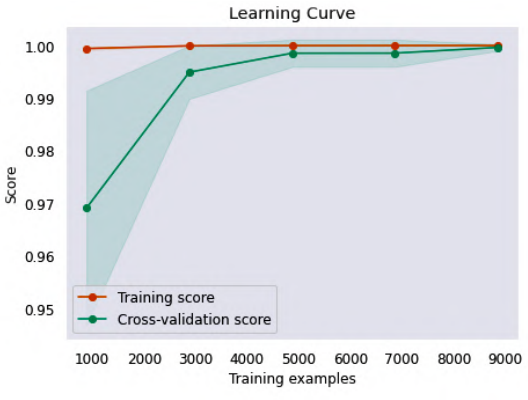

Fig. 14. Learning curve of Pearson

	index	productId	estimated_rating
0	0	CST	1
1	1	IND	1
2	2	MNP	1
3	3	REP	1
4	4	UNA	1

	index	productId	estimated_rating
0	0	CST	1
1	1	DEM	1
2	2	GRN	1
3	3	IAP	1
4	4	IND	1

Fig. 15. Results of the hybrid approach

6. Conclusion

The growth of IoT devices generating a large amount of data has made cloud computing an indispensable technology to facilitate data storage, processing, analysis and easy access. As the usage is diversified, it is very important that it is always in an optimal way. Several methods such as recommender systems can be used to improve its use. Recommender systems are known to provide relevant suggestions to a user based on his preferences or needs. However, with the rapid progress of technology. Therefore, current recommender systems rely on the Internet of Things (IoT) to achieve certain goals. In addition, there is too much information available on the Internet that needs to be analyzed or calculated for recommendation purposes. However, with the increase in the volume of data, the system performance gradually degrades. Therefore, a distributed environment such as the cloud is needed to perform and store all the computations on a single system. Based on this study, we observe that all the above areas are interdependent.

In this work, we have a comparative study on the performance of fourteen models applied to the election dataset among others: Rocchio, Latent Semantic Analysis, K-Nearest Neighbors, Naïve Bayes, Decision Tree Classifier, Support Vector Machine, Linear Regression, K-means, Slope One, Non-matrix Factorization, Singular Decomposition Vector, Co-clustering, Pearson Correlation and Artificial Neural Network. The evaluation of the algorithms is based on precision, recall, f1-score, accuracy, RMSE, MSE, MAE and area under the curve. The results, shows that K-NN and SGDC GridSearchCV perform best in terms of all evaluation parameters.

AUTHORS

Aristide Ndayikengurukiye* – Mohammed V University, Rabat, Morocco. e-mail: lemure.dede@gmail.com.

Abderrahmane Ez-Zahout – Mohammed V University, Rabat, Morocco. e-mail: abderrahmane.ezzahout@um5.ac.ma.

Akou Aboubakr – Mohammed V University, Rabat, Morocco. e-mail: aboubakar_aakaou@um5.ac.ma.

Youssef Charkaoui – Mohammed V University, Rabat, Morocco. e-mail: youssef_charkaoui@um5.ac.ma.

Omary Fouzia – Mohammed V University, Rabat, Morocco.

* Corresponding author

REFERENCES

[1] R. M. Frey, R. Xu and A. Ilic, "A Novel Recommender System in IoT". In: *2015 5th International Conference on the Internet of Things (IOT 2015)*, 2015, 10.3929/ETHZ-A-010561395.

[2] L. Mathiassen, "Collaborative practice research", *Information Technology & People*, vol. 15, no. 4, 2002, 321–345, 10.1108/09593840210453115.

[3] R. P. Adhikary, "Effectiveness of Content–based Instruction in Teaching Reading", *Theory and Practice in Language Studies*, vol. 10, no. 5, 2020, 10.17507/tpls.1005.04.

[4] S. Sharma, A. Sharma, Y. Sharma and M. Bhatia, "Recommender system using hybrid approach". In: *2016 International Conference on Computing, Communication and Automation (ICCCA)*, 2016, 219–223, 10.1109/CCAA.2016.7813722.

[5] H. J. Zimmermann, "Fuzzy set theory", *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 3, 2010, 317–332, 10.1002/wics.82.

[6] A. Gosain and S. Dahiya, "Performance Analysis of Various Fuzzy Clustering Algorithms: A Review", *Procedia Computer Science*, vol. 79, 2016, 100–111, 10.1016/j.procs.2016.03.014.

[7] T. Di Noia, R. Mirizzi, V. C. Ostuni and D. Romito, "Exploiting the web of data in model-based recommender systems". In: *Proceedings of the sixth ACM conference on Recommender systems - RecSys '12*, 2012, 253–256, 10.1145/2365952.2366007.

[8] A. A. Abdallat, A. I. Alahmad, D. A. A. Amimi and J. A. AlWidian, "Hadoop MapReduce Job Scheduling Algorithms Survey and Use Cases", *Modern Applied Science*, vol. 13, no. 7, 2019, 10.5539/mas.v13n7p38.

[9] M. Uta, A. Felfernig, V.-M. Le, A. Popescu, T. N. T. Tran and D. Helic, "Evaluating recommender systems in feature model configuration". In: *Proceedings of the 25th ACM International Systems and Software Product Line Conference – Volume A*, 2021, 58–63, 10.1145/3461001.3471144.

[10] N. Bhalse and R. Thakur, "Algorithm for movie recommendation system using collaborative filtering", *Materials Today: Proceedings*, 2021, 58–63, 10.1016/j.matpr.2021.01.235.

[11] K. Inagaki, S. Takaki, Y. Honda, M. Inoue, N. Mori, H. Kawakami, Y. Kawakami, M. Kawakami, H. Okamoto, K. Tuji, M. Tuge and K. Chayama, "A case of hepatitis E virus infection in a patient with primary biliary cholangitis", *Acta Hepatologica Japonica*, vol. 58, no. 3, 2017, 183–190, 10.2957/kanzo.58.183.

[12] M. Pazzani and D. Billsus, "Learning and Revising User Profiles: The Identification of Interesting Web Sites", *Machine Learning*, vol. 27, no. 3, 1997, 313–331, 10.1023/A:1007369909943.

[13] H. Christian, M. P. Agus and D. Suhartono, "Single Document Automatic Text Summarization using Term Frequency-Inverse Document Frequency (TF-IDF)", *ComTech: Computer, Mathematics and Engineering Applications*, vol. 7, no. 4, 2016, 285–294, 10.21512/comtech.v7i4.3746.

[14] R. Madani, A. Ez-Zahout and A. Idrissi, "An Overview of Recommender Systems in the Context of Smart Cities". In: *2020 5th International Conference on Cloud Computing and Artificial Intelligence: Technologies and Applications (CloudTech)*, 2020, 1–9, 10.1109/CloudTech49835.2020.9365877.

[15] V. Vaidhehi and R. Suchithra, "A Systematic Review of Recommender Systems in Education", *International Journal of Engineering & Technology*, vol. 7, no. 3.4, 2018, 188–191, 10.14419/ijet.v7i3.4.16771.

[16] Y. Ge, S. Zhao, H. Zhou, C. Pei, F. Sun, W. Ou and Y. Zhang, "Understanding Echo Chambers in E-commerce Recommender Systems". In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, 2261–2270, 10.1145/3397271.3401431.

[17] A. Felfernig, V.-M. Le, A. Popescu, M. Uta, T. N. T. Tran and M. Atas, "An Overview of Recommender Systems and Machine Learning in Feature Modeling and Configuration". In: *15th International Working Conference on Variability Modelling of Software-Intensive Systems*, 2021, 1–8, 10.1145/3442391.3442408.

[18] X. Zheng, L. D. Xu and S. Chai, "QoS Recommendation in Cloud Services", *IEEE Access*, vol. 5, 2017, 5171–5177, 10.1109/ACCESS.2017.2695657.

[19] Y. R. Shrestha and Y. Yang, "Fairness in Algorithmic Decision-Making: Applications in Multi-Winner Voting, Machine Learning, and Recommender Systems", *Algorithms*, vol. 12, no. 9, 2019, 10.3390/a12090199.

[20] I. Palomares, C. Porcel, L. Pizzato, I. Guy and E. Herrera-Viedma, "Reciprocal Recommender Systems: Analysis of state-of-art literature, challenges and opportunities towards social recommendation", *Information Fusion*, vol. 69, 2021, 103–127, 10.1016/j.inffus.2020.12.001.

[21] D. S. Tarragó, C. Cornelis, R. Bello and F. Herrera, "A multi-instance learning wrapper based on the Rocchio classifier for web index recommendation", *Knowledge-Based Systems*, vol. 59, 2014, 173–181, 10.1016/j.knosys.2014.01.008.

[22] R. M. Suleman and I. Korkontzelos, "Extending latent semantic analysis to manage its syntactic blindness", *Expert Systems with Applications*, vol. 165, 2021, 10.1016/j.eswa.2020.114130.

[23] R. J. Kuo, C.-K. Chen and S.-H. Keng, "Application of hybrid metaheuristic with perturbation-based K-nearest neighbors algorithm and densest imputation to collaborative filtering in recommender systems", *Information Sciences*, vol. 575, 2021, 90–115, 10.1016/j.ins.2021.06.026.

[24] S. Renjith, A. Sreekumar and M. Jathavedan, "An extensive study on the evolution of context-aware personalized travel recommender systems", *Information Processing & Management*, vol. 57, no. 1, 2020, 10.1016/j.ipm.2019.102078.

[25] R. Panigrahi and S. Borah, "Rank Allocation to J48 Group of Decision Tree Classifiers using Binary and Multiclass Intrusion Detection Datasets", *Procedia Computer Science*, vol. 132, 2018, 323–332, 10.1016/j.procs.2018.05.186.

[26] J. Hamidzadeh, E. Rezaeenik and M. Moradi, "Predicting users' preferences by Fuzzy Rough Set Quarter-Sphere Support Vector Machine", *Applied Soft Computing*, vol. 112, 2021, 10.1016/j.asoc.2021.107740.

[27] K. Luo, S. Sanner, G. Wu, H. Li and H. Yang, "Latent Linear Critiquing for Conversational Recommender Systems". In: *Proceedings of The Web Conference 2020*, 2020, 2535–2541, 10.1145/3366423.3380003.

[28] A. Yassine, L. Mohamed and M. Al Achhab, "Intelligent recommender system based on unsupervised machine learning and demogra-

phic attributes", *Simulation Modelling Practice and Theory*, vol. 107, 2021, 10.1016/j.simpat.2020.102198.

[29] Q.-X. Wang, X. Luo, Y. Li, X.-Y. Shi, L. Gu and M.-S. Shang, "Incremental Slope-one recommenders", *Neurocomputing*, vol. 272, 2018, 606–618, 10.1016/j.neucom.2017.07.033.

[30] T. V. R. Himabindu, V. Padmanabhan and A. K. Pujari, "Conformal matrix factorization based recommender system", *Information Sciences*, vol. 467, 2018, 685–707, 10.1016/j.ins.2018.04.004.

[31] A. L. Vizine Pereira and E. R. Hruschka, "Simultaneous co-clustering and learning to address the cold start problem in recommender systems", *Knowledge-Based Systems*, vol. 82, 2015, 11–19, 10.1016/j.knosys.2015.02.016.

[32] F. Maazouzi, H. Zarzour and Y. Jararweh, "An Effective Recommender System Based on Clustering Technique for TED Talks", *International Journal of Information Technology and Web Engineering (IJITWE)*, vol. 15, no. 1, 2020, 35–51, 10.4018/IJITWE.2020010103.

[33] C. Djellali and M. Adda, "A New Hybrid Deep Learning Model based-Recommender System using Artificial Neural Network and Hidden Markov Model", *Procedia Computer Science*, vol. 175, 2020, 214–220, 10.1016/j.procs.2020.07.032.

Int.J.of Computer Networks and Applications (IJCNA)

mar. 4 juin 04:47 (il y a 1 jour)

À moi, doukharim, ericniyukuri, eric.muheto.1, abderrahmane.ezzahout, omary

Dear Author,

Manuscript id: IJCNA-2024-M-132

Title: SOAVMP: Multi-objective Virtual Machine Placement in cloud computing based on the Seagull optimization algorithm

We are pleased to inform you that the above mentioned manuscript is **accepted for publication with minor revision** in *International Journal of Computer Networks and Applications (IJCNA)*. Kindly send us the revised version of the manuscript within **Three** days of receipt of this mail, that is **06/June/2024**. If you need more time to revise the manuscript feel free to contact us.

Reviewer comments are attached in this mail. Carefully revise the manuscript according to the reviewer's comments and send us the revised version of the manuscript within the stipulated time frame. **The first consecutive pages of the revised version of the manuscript must contain the changes you have made as a response to the reviewer's comments.**

The final decision of your manuscript depends upon the revised manuscript. The revised manuscript should be submitted for another round of review.

Procedure to submit the revised manuscript:

The following care must be taken in submitting the revised version of the manuscript.

1) Follow the URL to submit the revised version of the manuscript: http://www.ijcna.org/paper-submission.php

2) Choose the submission type as: "**Revised Submission**".

Comments for the author(s):

The first consecutive pages of the revised version of the manuscript must contain the changes you have made as a response to the reviewer's comments.

Reviewer 1:

Plagiarism rate of this paper is 19% which should be reduced to less than 10%. The same comment was given in the previous revision also. However, the authors failed to address the same. This could be the final chance. The report is attached. Find the same.

Reviewer 2:

Accepted.

IJCNA is a highly prestigious peer-reviewed international journal. Hence, Purely high quality research articles alone will be considered for publication in IJCNA. Authors are advised to submit original unpublished manuscripts for consideration.

International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2024/224431 Volume 11, Issue 1, January – February (2024)

RESEARCH ARTICLE

Optimizing Virtual Machines Placement in a Heterogeneous Cloud Data Center System

Aristide Ndayikengurukiye
Intelligent Processing and Security of Systems Team, Faculty of Sciences, Mohammed V University, Rabat, Morocco.
lemure.dede@gmail.com

Abderrahmane Ez-Zahout
Intelligent Processing and Security of Systems Team, Faculty of Sciences, Mohammed V University, Rabat, Morocco.
abderrahmane.ezzahout@um5.ac.ma

Fouzia Omary
Intelligent Processing and Security of Systems Team, Faculty of Sciences, Mohammed V University, Rabat, Morocco.
omary@fsr.ac.ma

Received: 20 July 2023 / Revised: 02 October 2023 / Accepted: 25 January 2024 / Published: 26 February 2024

Abstract – In a cloud computing environment, good resource management remains a major challenge for its good operation. Implementing virtual machine placement (VMP) on physical machines helps to achieve various objectives, such as resource allocation, load balancing, energy consumption, and quality of service. VMP (virtual machine placement) in the cloud is critical, so it's important to audit its implementation. It must take into account the resources of the physical server, including CPU, RAM, and storage. In this paper, a metaheuristic algorithm based on the Grey Wolf Optimization (GWO) method is used to optimize the placement of virtual machines in a cloud environment, effectively minimizing the number of active virtual machines used to host virtual servers. Experimental results demonstrate the effectiveness of the proposed method, called Grey Wolf Optimization for Virtual Machine Placement (GWOVMP). The method reduces power consumption by 20.99 and resource wastage by 1.80 compared with existing algorithms.

Index Terms – Cloud Computing, GWO Algorithm, Metaheuristics Algorithm, Optimization, Virtual Machine Placement, Data Center, Power Consumption.

1. INTRODUCTION

Nowadays, Cloud Computing (CC) has developed rapidly and has become an indispensable technology in the field of information technology. This has led to the emergence of ever larger modern data centers with a high number of physical devices, which causes a remarkable increase in energy consumption and huge cooling expenses. In these data centers, three types of physical equipment consume electricity, including servers, cooling systems, and network equipment. To get a better idea of the benefits a user can get, CC offers four main deployment modes: public cloud, private cloud, hybrid cloud, and community cloud. The various CC resources are provided in three main services: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS).

In the public cloud model, the various resources are managed by the service provider that owns the cloud, and its resources are sold to the public according to its own needs. In this kind of cloud, part of the resources can be rented and managed by the end users. Examples of some of the best-known cloud providers are Amazon, Google, and Microsoft [1]. For the private cloud model, the ownership of this type would be an entity or an organization that wants to use certain applications that contain very sensitive information. If it is a community cloud, the provider is a set of organizations that come together for a common purpose [2]. In this case, several companies for example can decide to federate their efforts by building and managing a cloud together. Through virtualization, these servers are offered to customers in the form of VMs, which allows a high level of flexibility for the proper management of resources. These data centers and above all facilitates the execution of workloads in an elastic way; certain techniques such as hardware consolidation are therefore combined to maximize energy efficiency. In a DC (data center), the virtualization technique also allows different applications to be run on VMs or containers. This technology also allows efficient pooling of physical resources, such as CPU, RAM,

RESEARCH ARTICLE

and I/O interfaces, while allowing physical servers to host multiple VMs.

Although cloud computing has several advantages, the biggest challenges of cloud computing are the costs associated with energy consumption in data centers as well as carbon dioxide emissions. To overcome these difficulties, the advantages of virtualization are used to minimize the number of active PMs (Physical Machines) and to switch off inactive PMs. This requires the implementation of a strategy to better manage the placement of virtual machines (VMPs). To minimize the total power of a DC, it is therefore necessary to restrict the consumption of the PMs. In addition, there are other challenges such as security, which some studies have focused on [3]. Researchers have been interested in the fact that they can exploit the large storage and processing capacities that cloud computing offers [4]. VMP in the cloud environment remains a topic that attracts the attention of various researchers. The optimization of VMP in data centers can improve the efficiency of the energy quantity and the optimal use of resources while maximizing the quality of services (QoS) offered.

The optimization of VMP problems in a cloud environment is considered an NP-hard problem for which it is very rare to converge on an optimal solution. Solving such a problem can be expensive in terms of computation time when common methods such as graph theory are used. Therefore, it is preferable to use metaheuristic and heuristic techniques. Metaheuristics suffer from high execution time but provide optimal solutions. Heuristic algorithms have a moderate execution time, but not with a good quality of solutions [5]. Several algorithms and objectives have been implemented to solve the VMP problem, including the fuzzy logic algorithm [6].

In this paper, an algorithm called GWOVMP is adapted from the GWO method to optimize the VMP in a heterogeneous DC. To compare GWOVMP with existing algorithms, we perform similarities using the CloudSIM tool. The main contributions of this article are:

- Formulation of mathematical models of the problem of placing VMs in the cloud;
- A metaheuristic algorithm based on the grey wolf hierarchy called GWOVMP is proposed to solve this VMP problem in a cloud environment;
- We perform a simulation using CloudSIM and evaluate the proposed algorithm against those in the state of the art.

The remainder of this document is organized as follows. Section 2 presents a review of the literature and presents the various related works. Section 3 presents the problem formulation and the GWO for virtual machine placement. The experimental evaluation of the GWOVMP algorithm and the analysis of the results of our work are found in Section 4. Finally, Section 5 presents the conclusion and future work.

2. LITERATURE REVIEW

2.1. Virtual Machine Placement

Generally speaking, the placement problem is an old and classic one. It consists of filling boxes with different objects while using the minimum number of boxes. In the case of DCs, the placement of a virtual machine in the optimal assignment of VMs in CC consists of assigning VMs in a minimum number of PMs of a DC. This results in VMs being considered as objects and PMs as boxes.

The objective of optimal placement in the DC is to distribute loads across all physical servers while minimizing factors such as power consumption, increasing QoS, meeting service-level agreement (SLA), and reducing resource wastage. In a cloud-based DC, virtualization is one of the key technologies to rely on.

The benefits of virtualization, such as consolidation, migration, and load balancing of different VMs across MPs, increase DC efficiency and reduce operating costs. Figure 1 illustrates the simplified architecture of this virtualization technique.

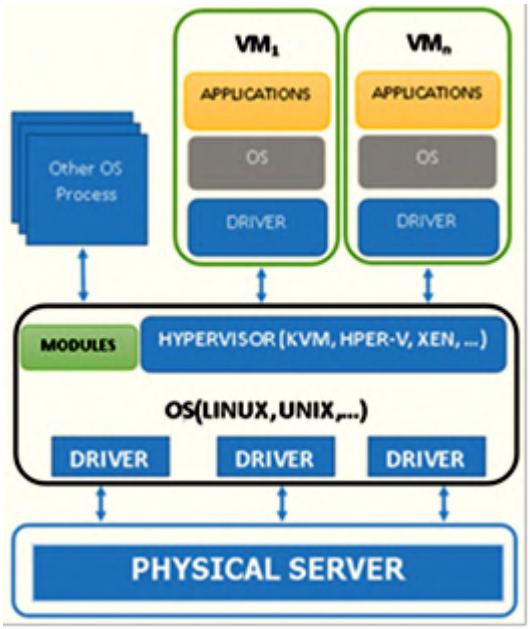

Figure 1 Virtualization Architecture

2.2. Grey Wolf Optimization

In the last two decades, metaheuristic optimization has become very popular. Some metaheuristic algorithms are well-known and used in different application areas [7]. Algorithms such as Ant Colony Optimization (ACO), Genetic Algorithm (GA), and Particle Swarm Optimization (PSO)

RESEARCH ARTICLE

have been used in various research including the VMP problem. The Grey Wolf Optimizer is one of the life-based metaheuristic algorithms that harkens back to the hunting techniques of the grey wolf and is inspired by the grey wolf's leadership chain. These grey wolves live in packs ranging from 5 to 12 on average and there are four types of these grey wolves such as α, β, δ, and ω. The first level is composed of the alphas which are male or female grey wolves that are stronger and guide the others in appropriate areas.

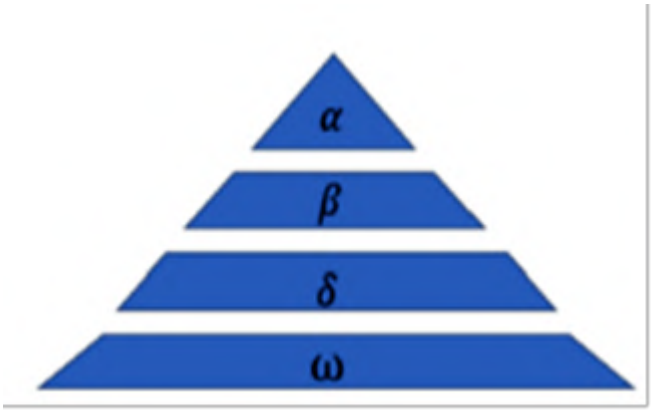

Figure 2 Social Hierarchy of Grey Wolves [8]

Figure 2 shows the social hierarchy of grey wolves. The equations (1) and (2) below present the mathematical modeling of this algorithm which is based on the encirclement of the prey by these grey wolves [9]:

$$\overrightarrow{D} = \overrightarrow{C}.\overrightarrow{X_p}(t) - \overrightarrow{X}(t) \quad (1)$$

$$\overrightarrow{X}(t+1) = \overrightarrow{X_p}(t) - \overrightarrow{A}.\overrightarrow{D} \quad (2)$$

In Equations (2) and (5), the current iteration is represented by t, the coefficients are represented in equations (3) and (4) by and $\vec{C}$, $\vec{X}$ is the vector that indicates the position of the grey wolf while the vector indicates the position of the prey. The following equations how the vectors $\vec{A}$, $\vec{C}$ and $\vec{a}$ shall be calculated[10]:

$$\overrightarrow{A} = 2\overrightarrow{a}.\overrightarrow{r_1} - \overrightarrow{a} \quad (3)$$

$$\overrightarrow{C} = 2\overrightarrow{a}.\overrightarrow{r_2} \quad (4)$$

$$\overrightarrow{a} = 2 - t\frac{2}{NumOfIt} \quad (5)$$

The vectors $\vec{r}$1 and $\vec{r}$2 are random in the interval [0, 1], and during the exploration and exploitation phase, the vector decreases linearly from 2 to 0 during these iterations [11]. The best solution is represented first by the alphas, then the betas, and finally the deltas.

The omegas are considered undesirable solutions and are therefore not taken into account. The Figure 3[8] shows the different positions of the wolves during the hunt and can be defined mathematically by the equations below[12], [13]:

$$\overrightarrow{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (6)$$

Where $\vec{X}_1 + \vec{X}_2 + \vec{X}_3$ are defined in Equations (7)-(9) below[13]:

$$\overrightarrow{X_1} = \overrightarrow{X_\alpha} - \overrightarrow{A_\alpha}.\overrightarrow{D_1} \quad (7)$$

$$\overrightarrow{X_2} = \overrightarrow{X_\beta} - \overrightarrow{A_\beta}.\overrightarrow{D_2} \quad (8)$$

$$\overrightarrow{X_3} = \overrightarrow{X_\delta} - \overrightarrow{A_\delta}.\overrightarrow{D_3} \quad (9)$$

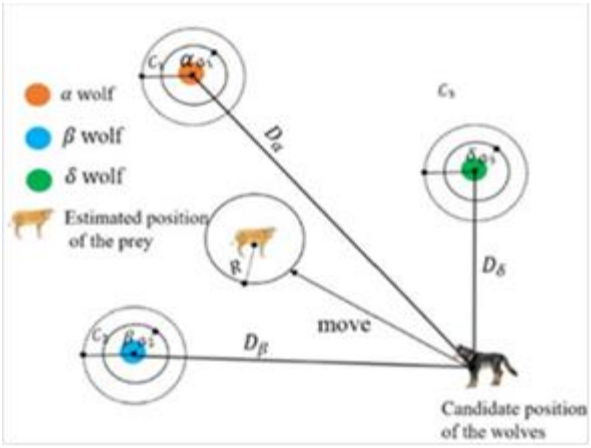

Figure 3 Update on the Position of Grey Wolves during the Hunt [14]

In this case, $\vec{X}_1 + \vec{X}_2 + \vec{X}_3$ are the best solutions to an iteration $\overrightarrow{A_\alpha}$, $\overrightarrow{A_\delta}$ which are defined in Equation (3), and are defined in Equations (10) -(12) as follow [15]:

$$\overrightarrow{D_\alpha} = \left|\overrightarrow{C_1}.\overrightarrow{X_\alpha} - \vec{X}\right| \quad (10)$$

$$\overrightarrow{D_\beta} = \left|\overrightarrow{C_2}.\overrightarrow{X_\beta} - \vec{X}\right| \quad (11)$$

$$\overrightarrow{D_\delta} = \left|\overrightarrow{C_3}.\overrightarrow{X_\delta} - \vec{X}\right| \quad (12)$$

2.3. Related Works

In [16] they proposed an algorithm that performs VMP in a heterogeneous and multidimensional cloud environment in a random way. To optimize energy and resource utilization, they used GRVMP (Greedy Randomized Virtual Machine Placement) which is inspired by the two-choice power model and places VMs on the most energy-efficient PMs. They evaluated the performance of their algorithms using synthetic and real production scenarios with performance matrices. Their results show a significant reduction in active PMs as well as the total resource wastage of their algorithm compared to the methods of the previous study.

RESEARCH ARTICLE

The authors have implemented smart techniques to deal with the VMP optimization problem [17]. Their objective is to be able to decrease the number of active physical servers as well as the energy consumption. They adapted two algorithms from the Grey Wolf Optimization model to a VMP problem. The proposed algorithms have been evaluated in a CloudSIM environment composed of homogeneous and heterogeneous servers. Compared to existing techniques, their algorithms minimize the number of active PMs used to host VMs.

For optimal placement in a Cloud Data Center (CDC), in [18] the authors proposed an algorithm based on the hybrid discrete whale and the multi-objective optimization algorithm. The multi-verse optimizer with chaotic functions is also used to optimize placement in a cloud environment. The main objective is to minimize the energy consumption that is consumed in the CDCs by reducing the active physical machines, as well as to reduce the waste of resources by using the optimal placement of VMs on the MPs. The applied method avoids the increase in the migration rate of VMs on PMs. The results they obtained show the performance of the proposed algorithm compared to other state-of-the-art algorithms.

In [19], [20], the authors provided a technique based on a particle swarm algorithm to improve task scheduling in a Cloud system. In the proposed technique, the choice of an appropriate objective function allowed them to perform dynamic VMP on the PM, balance the workload of VMs, reduce the time of all tasks while maximizing the utilization of all resources, and increase productivity. The proposed solution provides optimal results for task scheduling, proper allocation of tasks in VMs placement on the PM, and on the process of outsourcing VMs, the time was improved by 0.02.

In [20], the authors developed two algorithms for optimal resource management in the cloud. The first algorithm used unsupervised planning with the K-means algorithm and the second is the KNN algorithm for supervised learning. This also applies to the systematic analysis of VM and PM models, which allowed them to establish a rule for hybrid and dynamic deployment of cloud data center resources.

The authors of [5] proposed a technique to solve the problem related to the optimization of VMP in the cloud taking into account power and traffic. The proposed method minimizes CDC energy consumption, resource wastage, and network consumption. The method was compared with state-of-the-art algorithms. The results obtained show a 29% reduction in network consumption, an 18% reduction in overall energy consumption, and a 68% reduction in resource wastage. The proposed approach jointly minimizes energy consumption, optimal network utilization, and resource wastage in a heterogeneous and multidimensional cloud system. Their results were obtained after comparison with state-of-the-art fat tree technology.

In [21], a GWO algorithm was proposed based on the reliability capacity of resources for adaptive load balancing. The method they used first finds the inactive or occupied nodes and then calculates the threshold and suitability function for each node. They used CloudSim for simulation and their results improved the cost and response time of the proposed method lower than compared to other techniques.

The authors in [22] proposed an improved genetic algorithm (l-GA) based on the virtual hierarchy architecture model (VHAM) to solve the problem of VM placement. They used the finite element analysis (FEA) application to conduct the experiments. Their results show the efficient reduction of the number of MPs used to host VMs and thus reduce the power consumption in data centers while ensuring the availability of placement. I-GA performs better than other algorithms even in small instances.

In [23], the authors have formalized the VMP problem into an intelligent optimization while building on the different specifications of a data center. The objective of their work is to perform a good placement of VMs on PMs without affecting performance. With the experiments they conducted, the method used is a metaheuristic Particle Swarm Optimization Algorithm that gave efficiency on a significant decrease of energy of the data center and optimal balancing between different resources.

Table 1 provides a comparative study of some of the work carried out on VMP. It shows the different parameters used, the simulation environment (SE), and the different algorithm names implemented.

Table 1 Comparative Study of the State-of-the-Art

work	Parameters								Algorithm	SE
	Resource utilization				Resource wastage	SLA	Energy	Others		
	CPU	RAM	Storage	Bandwidth						
[15]	×	×			×		×		GRVMP	Amazon EC2

RESEARCH ARTICLE

[17]	×	×					×		GWO-VMP	CloudSIM
[18]	×	×	×	×	×	×	×			Amazon EC2
[19]	×	×			×		×			MATLAB
[20]	×				×		×		EHML	CloudSIM Amazon Cloud datacenter
[5]	×				×		×	×		Java
[21]					×		×	×		CloudSIM
[22]	×	×			×		×		I-GA	CloudSIM
[23]	×	×					×		RPMOTA	
This Work	×	×	×		×		×		GWOVMP	CloudSIM

3. PROBLEM FORMULATION AND GWO FOR VIRTUAL MACHINE PLACEMENT (GWOVMP)

3.1. Problem Formulation

In this article, the basic objective is to implement an optimal placement scheme for virtual machines. The main goal of the VMP is to be able to modify the original allocation of virtual machines to physical nodes to minimize energy costs.

In this work, the goal is to minimize this energy consumption in cloud computing according to the available resources. Let us note the matching function f. This virtual machine placement problem can be denoted as f: V→P [5]. The minimum number of active servers for an optimal VMP solution can be formulated as shown in equation (13):

$$f(S) = \min \sum_{i=1}^{n} Y_i \qquad (13)$$

In Cloud Computing, DCs are composed of heterogeneous physical machines. In this paper, a number m of PMs and a number n of VMs are considered. P denotes the PM set where i is the i^{th} PM, i belongs to $[1, \dots, m]$, and $P_i \in P$. Similarly, V denotes the set of heterogeneous VMs, where V_j denotes the j^{th} VM, j belongs to $[1, \dots, n]$, and $V_j \in V$.

To represent the capacity of i^{th}PM represented by the P_i r-dimension vector $P_i = \{P_i^1, P_i^2, \dots, P_i^r\}$, which P_i^k represents the capacity of the resource k^{th} of i^{th} PM, $\forall k = \{1,2, \dots, r\}$. Represent the capacity of i^{th} VM represented by the V_j r-dimensional vector $V_j = \{R_i^1, R_i^2, \dots, R_i^r\}$, where R_i^k represents the capacity of the resource k^{th} of J^{th} VM, $\forall k = \{1,2, \dots, r\}$. To express the relationship between VM and PM, this function is represented by the binary matrix $X_{m \times n}$ as shown in the equation (14):

$$x_{ij} = \begin{cases} 1 & if\ V_j\ is\ assigned\ to\ P_i \\ 0 & Otherwise\ ,\forall i \in Pet\ \forall j \in V \end{cases} \qquad (14)$$

To designate the state of the PM, the matrix y_n, and P_i are used and is active if $y_i = 1$ and inactive $y_i = 0$. This is shown in equation (15):

$$y_{ij} = \begin{cases} 1 & if\ \sum_{i=1}^{n} P_i \geq 1\ \ \forall i \in P_i \\ 0 & Otherwise \end{cases} \qquad (15)$$

In this work, processor, memory, and storage resources are considered. For the capabilities of computing power are represented by P^{cpu} and that of memory by P^{ram} . Similarly, the processor and memory are represented by V^{cpu} and V^{ram} which leads us to the following constraints [24]:

$$\sum_{i}^{M} V^{cpu} \cdot x_{ij} \leq P^{cpu} \cdot y_i \ \ \forall i \in P \qquad (16)$$

$$\sum_{i}^{N} V^{ram} \cdot x_{ij} \leq P^{ram} \cdot y_i \ \ \forall i \in P \qquad (17)$$

In equations (16) and (17) we specify that the CPU and memory should not exceed, which specifies the capacity constraints. Equation (18), states that the total storage space

RESEARCH ARTICLE

requirements for VMs on PMs must not exceed the storage space of the PM it is hosting.

$$\sum_i^N V^{sto} \cdot x_{ij} \leq P^{sto} \cdot y_i \quad \forall i \in P \qquad (18)$$

Several works show that the energy consumption of DCs in CC is linear with a linear load [24], [25], [26], [27] . Shutting down idle servers will reduce the cost of energy consumption. Energy consumption in the cloud is represented by equation (19):

$$P^{cpu} = \begin{cases} P_{iddle} + (P_{full} - P_{iddle}) \times U_i^{cpu} & if\ U_i^{cpu} > 0 \\ 0, & otherwise \end{cases} \qquad (19)$$

Where P_{iddle} is the power of energy consumption by the PM when it is in an idle state, P_{full} is the power that is consumed in a fully utilized CPU, and $P^{cpu}{}_i \in [0, 1]$. The equation (20) shows that U^{cpu} is the processor (CPU) rate of the PM P_i as a function of the VM demand V_i.

$$U_i^{cpu} = \frac{\sum_{j=1}^{n} x_{ij} \times R_i^{cpu}}{P_i^{cpu}} \qquad (20)$$

To calculate the waste of different resources of a PM of dimension r, we generalize the equation (21) so that it is applied to resources [28]:

$$W_i = \frac{|R_i^{cpu} - R_i^{ram}|}{|P_i^{cpu} + P_i^{ram}|} + \varepsilon \qquad (21)$$

Where W_i represents the waste of different PM resources. R^{cpu} and R^{ram} represent the normalized processor and memory resources of the physical machine, respectively; the values of P^{cpu} and P^{ram} describe the normalized use of the machine. ϵ is the small real number equal to 0.0001. This model aims to best include the resources of all PMs and allows for the balance between multiple resources to be achieved. The mathematical equation (22) indicates the total waste of resources W^{tot} of all PMs in a CDC [29]:

$$W^{tot}{}_i = \sum_{i=1}^{m} y_i \times \frac{\sum_{k=1}^{r} |R_i^{cpu} - R_i^{ram}|}{\sum_{k=1}^{r} P_k^r} + \varepsilon \qquad (22)$$

3.2. GWO for Virtual Machine Placement

The effective reduction of power consumption in a cloud-based data center is usually achieved by dramatically reducing the number of active physical machines. In this case, an algorithm adapted from the GWO algorithm is used to solve the VMP problem to obtain an optimal solution that allocates VMs to a minimum number of active physical servers.

The best solution that has a minimum number of active PMs is chosen as S. In the initial state, the optimal placement of the VMs in the MPs is not known and the best solution that has been generated, designated SI, is considered first. Indeed, to match the m VMs to the n PMs, a random distribution of the m VMs to the n PMs and each VM is subjected to a single PM. There is therefore m^n possible solutions.

To search for prey, wolves continuously update their positions. After the solution step which is randomly distributed for each wolf that joins the pack. GWOVMP generates a new solution while updating the existing solution of each wolf to find the most optimal distribution of VMs on the MP. For each iteration t, the wolf positions are updated based on the best solutions generated by α, β, and δ.

To minimize the number of active physical machines and to minimize power consumption in a cloud-based system, the solution must be improved to reduce the number of active running VMs under all conditions. In general, the GWO algorithm updates the path of the best solution set based on equation (6), where one by one some VM locations are updated. It can be seen that some locations are updated outside the [1, ..., n] boundary. In this case, the algorithm relies on the binary model to recalculate the allocation of these Virtual Machines while ensuring that the physical servers are within the limit:

$$W_{ij}(t+1) = \begin{cases} 1 & if\ sigmoid\ (\frac{\vec{x}_1 + \vec{x}_2 + \vec{x}_3}{3}) \\ 0, & otherwise \end{cases} \qquad (23)$$

In general, without taking into account the case treated in this paper, we have in Equation (23) the rand which represents a random number extracted from the uniform distribution that ∈ [0, 1]. x_{ij} represents the binary position that is updated at iteration t and in the determined dimension, the sigmoid function is determined by the equation (24):

$$sigmoid\ (a) = \frac{1}{1 + e^{-10(x - 0.5)}} \qquad (24)$$

Input: VM, PM

Output: Placement of VMs on PMs

Step 1: Initialization of a population of *n* grey wolves positions randomly. Set parameters a via Equation (4) and (5). *n* is defined as the number of wolves considered as a search factor. Determine the total number of iterations. Set *it= 1* as the initial iteration.

Step 2: Let *n* Wolves build and save the top tree solutions α, β and δ.

Step 3: Update the solutions of α, β and δ with Equations (16) and (17).

Step 4: Calculate the different fitness values for all solutions and identify the tree best solution for the current iteration

Step 5: Terminate the algorithm. Check the current number of iterations; the algorithm terminates if the number reached exceeds the maximum number of iterations. Otherwise, increment and return to step (3).

RESEARCH ARTICLE

Step 6: Return α.

Algorithm 1 GWOVMP

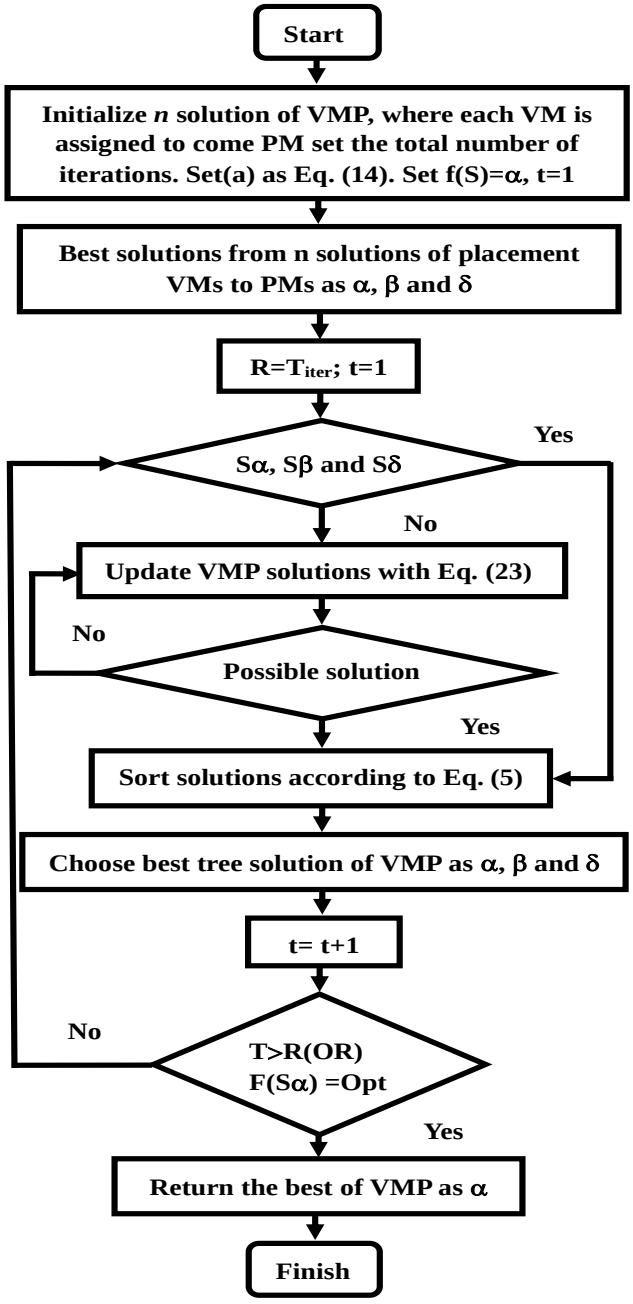

Figure 4 GWOVMP Flowchart

Algorithm 1 describes the GWOVMP based on an adjustment of the various basic operations of the Gray Wolf optimization algorithm, and the corresponding flowchart is shown in Figure 4.

In case a VM has not been allocated during an update, deletion, or duplicate overload, an operation is used to reallocate it. The constraint of the VMs is evaluated by Equations (16) and (17) after an update of the wolf positions. In case the value of x_{ij} corresponding to V_j has not been allocated to any PM during the update, the VM is reallocated to a non-overloaded PM with sufficient resources to meet the demand.

As mentioned, GWO is based on a master strategy in which wolves update their positions each time to attack the prey based on the positions of the top three wolves. In addition, the CDC's energy consumption needs to be considerably reduced by limiting the number of active physical servers. This can be achieved by improving the solution on active servers while reducing their number.

In this case, it can be seen that the exploration phase carried out by the wolves can sometimes be done in the opposite direction involving the binary search space. It is on the generalities of this algorithm that if each bit of the values α, β and δ goes to one, the value that corresponds to xij to be submitted to P_i is updated.

To balance and maximize the use of memory and CPU resources, VMs are reallocated to PMs that give the smallest absolute dissimilarity that is estimated between the use of CPU and memory resources after the virtual machine is used. The difference between the different CPU and memory resource usage after adding a virtual machine to the PM is calculated after adding the different resource requirements of the virtual machine to the resources used by the physical server.

Table 2 lists the various symbols used in this study. Each symbol has a particular meaning and has been carefully selected to contribute to the clarity and understanding of the various equations. This provides a handy visual reference for readers to quickly interpret the information contained in the work.

Table 2 Symbols and Descriptions

Symbol	Descriptions
P	Set of PMs
V	Set of VMs
P^{cpu}	CPU capacities of the PM
P^{ram}	CPU capacities of the PM
V^{cpu}	CPU capacities of the VM
V^{ram}	CPU capacities of the VM
P^{sto}	Storage capacities of the PM
V^{sto}	Storage capacities of the VM
P_{iddle}	Power that is consumed in an idle state

RESEARCH ARTICLE

P_{full}	Power that is consumed in an idle state
U_i^{cpu}	Normalized CPU utilization
R_i^{cpu}	Normalized CPU resources of the PM
R_i^{ram}	Normalized RAM resources of the PM
W_{tot}	Total waste of resources of all PMs in a CDC
x_{ij}	Assigning V to P or not
y_{ij}	Status of the machine (Active if y=1, Otherwise)

4. PERFORMANCE EVALUATION AND DISCUSSION

In this part of the paper, experimental tests were carried out to determine the performance of the proposed algorithm, called GWOVMP. This algorithm was implemented in Java like all the other algorithms compared. The simulator that was used in this work is CloudSIM and the toolkit version 5.0 of the CloudSIM simulator was used. It supports many IaaS featuressuch as on-demand resource provisioning and power consumption solutions [30], [31]. It also allows us to evaluate the performance of new applications and cloud strategies before they are developed in the real environment. The various tests and experiments were carried out on a computer equipped with an AMD Ryzen 5800U processor with 4.4 GHz and 16GB of RAM. The operating system used was Windows 11.

To evaluate the performance of this method, it is compared with algorithms such as First fit decreasing FFD[32], Modified best fit decreasing (MBFD)[33], Random first fit (RFF)[34] and Best Fit Decreasing (BFD)[35]. To test the effectiveness of the GWOVMP algorithm, instances of various sizes ranging from 100 to 4000 were created. Each VM is equipped with a 16-core processor and 32 GB RAM. A CPU of 1 to 4 cores is required for each VM, and RAM of 1 to 8 GB is randomly generated. In cases where the CPU is one of the most important resources, the ratio between the different memory and CPU requirements is almost 10:9.

The associated parameters with the GWOVMP algorithm are a = 2, which decreases linearly with each iteration, and the random vectors r1 and r2 are included in [0,1]. This gives this algorithm the advantage of defining a few parameters. In the case of the other algorithms, the parameters are set according to the basic literature. It is taken into account the 100% achievability of the resource usage and its implementation uses 100 iterations which stops early on the fifth iteration.

4.1. Resource Utilisation Performance

Table 3 shows the results obtained after comparing some existing algorithms with the proposed model. In this case, 100 to 2000 VMs are used to see the evolution of the results and to be able to conclude if a small number of evolving is applied to all the methods used for comparison. In the state of the art, FFD produces the worst result of all algorithms for all instances. GWOVMP performs well in many cases, especially as the number of active VMs increases. The BFD algorithm is the second-best performer. The RFF algorithm gives the worst results in all cases, which also shows that this algorithm does not fit a heterogeneous environment

Table 3 Number of Active Physical Machines

No	VM	BFD	RFF	MBFD	FFD	GWOVMP
1	100	16	16	16	18	16
2	200	32	33	33	37	32
3	300	48	46	46.03	53	48
4	400	63	64	64.4	74	63
5	500	78	77	77.63	91	78
6	600	94	95	95.77	111	99
7	1000	158	159	161.23	184	153.26
8	2000	316	317	319.64	366	278.08

The GWOVMP algorithm, which is the proposed and implemented solution, generates better results than all other solutions, except in the case of using a small number of VMs. As the problem size increases, the GWO result gives a satisfactory result.

Figure 5 shows the active number of physical servers that are used to support the different sets of virtual machines for the different algorithms used. The optimal PM solution used for the GWOVMP case is observed when a significant number of VMs are used.

RESEARCH ARTICLE

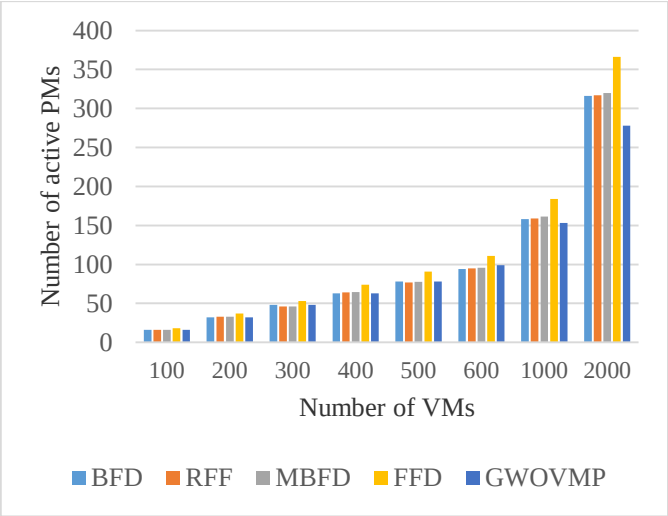

Figure 5 Number of Active PMs

Referring to Figure 6, the proposed GWOVMP model is the most efficient in terms of CPU utilization. This model provides optimal CPU utilization for the different algorithms, which becomes very noticeable when using a large number of virtual machines. Although the FFD algorithm is the least efficient, it is worth noting that it gives an optimal solution when allocating a very small number of VMs on PMs, compared with the BFD, RFF, and MBFD algorithms.

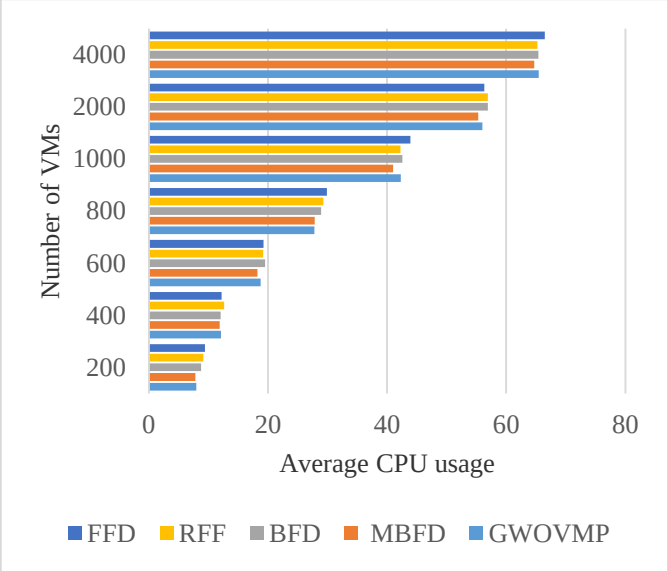

Figure 6 Average CPU Utilization

Figure 7 shows the average RAM usage of active servers in the allocation obtained by BFD, FFD, RFF, MBFD, and GWOVMP for the different tests. FFD and RFF achieved the highest memory usage. For FFD, low memory utilization is achieved by using a very small number of VMs up to 1000 VMs. MBFD gives better optimal solutions for small to medium numbers of VMs. The difference in this algorithm is observed when the number of VMs increases considerably. In this case, it becomes the second algorithm to offer a more optimal solution. In the case of the proposed GWOVMP algorithm, memory usage remains within reasonable limits in all possible cases.

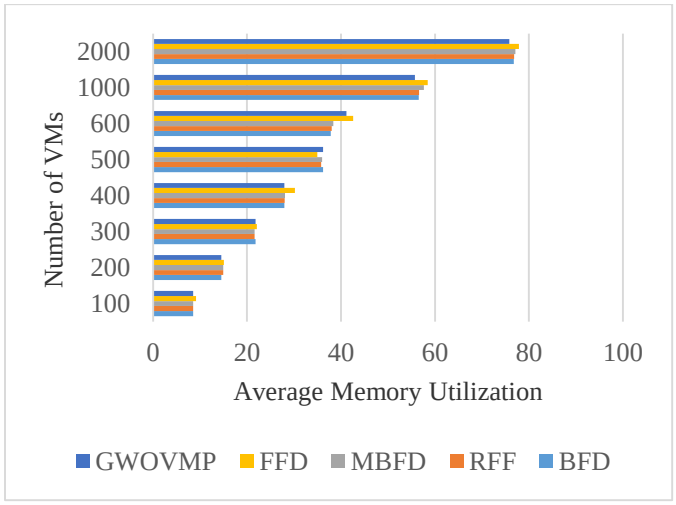

Figure 7 Average RAM Utilization

Figure 8 shows that the FFD method consistently achieves exceptionally high storage utilization in multiple scenarios, closely followed by the MBFD algorithm. However, it is GWOVMP, the proposed solution that stands out, particularly when faced with a substantial increase in the number of active virtual machines.

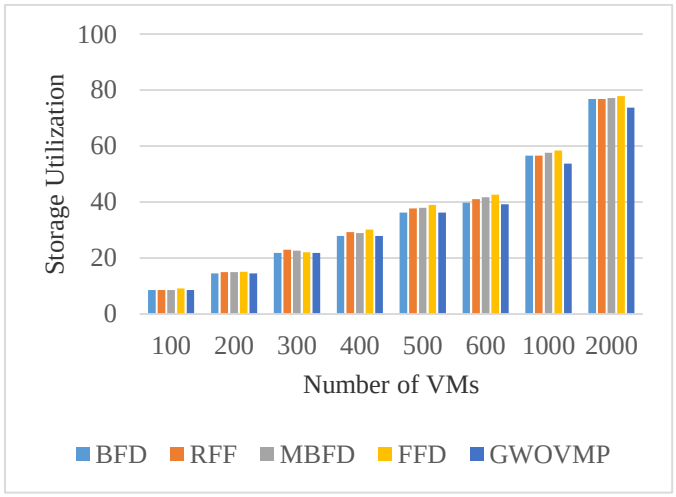

Figure 8 Storage Utilization

Remarkably, GWOVMP boasts an average improvement rate of 1.8% over all the methods studied. This underlines its efficiency and superiority in optimizing storage utilization, even in the face of increased operational demands. These results not only highlight the shortcomings of state-of-the-art

RESEARCH ARTICLE

methods such as FFD but also underscore GWOVMP's potential as a transformative solution to storage efficiency challenges in the cloud.

4.2. Resources Wastages

Figure 9 reveals the results on wasted resource utilization rates in a data center. We can see that the FFD algorithm consistently shows a high rate of resource wastage in multiple scenarios, closely followed by the MBFD algorithm.

However, it is the GWOVMP algorithm that stands out, particularly when confronted with a substantial increase in the number of active virtual machines. GWOVMP achieves an average improvement rate of 1.6 over all the methods studied. These results underline its effectiveness in mitigating resource wastage, making it a better solution than the methods compared.

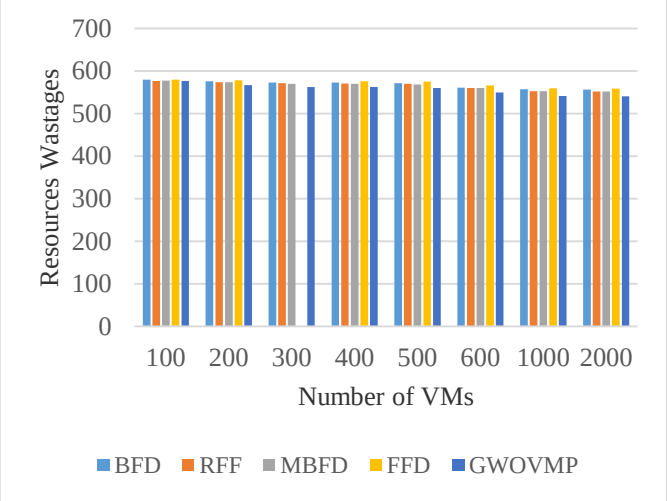

Figure 9 Resources Wastages

4.3. Energy Consummation Performance

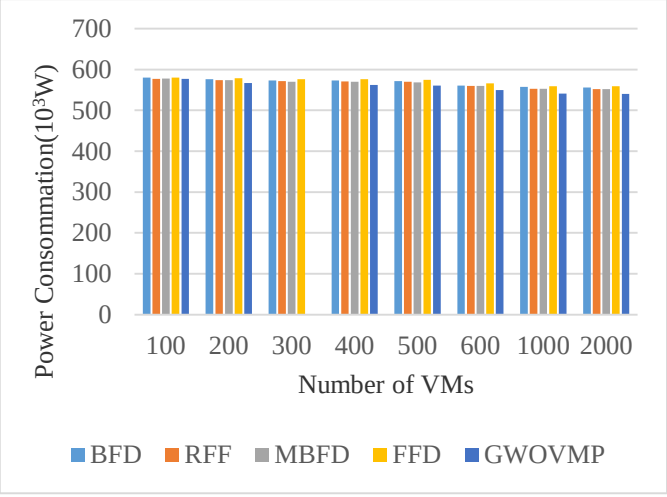

Figure 10 Power Consummation

This part of the article, reveals key results on the energy consumption of various algorithms that have been used to optimize the cloud data center environment.

Figure 10 shows a notable trend in which the energy consumption of the BFD, RFF, MBFD, FFD, and GWOVMP algorithms increases in direct correlation with the growing number of active virtual machines. Intriguingly, GWOVMP stands out from these leading methods by demonstrating a remarkable 20.99% reduction in energy consumption within the cloud infrastructure.

They highlight the maximum energy consumption, which peaks at 6580 kWh for BFD, and the minimum energy consumption, notably achieved by RFF with 6588 kWh. MBFD and FFD record energy consumptions of 65983 kWh and 6899 kWh, respectively, for an equivalent number of active user and PM requests. These statistics provide valuable quantitative evidence of GWOVMP's efficiency compared to its peers.

Collectively, these results not only highlight the escalating demand for energy in cloud computing but also underscore GWOVMP's potential as a more sustainable, energy-efficient solution for cloud infrastructures.

4.4. Discussion and Limitations

The choice of CloudSim as the simulation environment was dictated by its performance, as observed in state-of-the-art work. It enabled us to automate the simulation life cycle while reducing processing time. In this simulation environment, the proposed model was configured in a library according to the proposed configuration parameters.

The results obtained in this study allow us to state that the GWOVMP algorithm is better than the other algorithms for solving the VMP optimization problem. Like most of the work that has been used to compare the proposed algorithm, the work carried out does not take into account the network and certain evaluation metrics (e.g. migration, QoS, risk-aware resource allocation). To remedy this, this method needs to be improved by involving many more evaluation parameters. This work focused on the problem of VM placement to reduce energy, CPU, RAM, and storage usage. The other metric taken into account is resource wastage. Despite GWOVMP's performance, it needs to be improved to give much better results on storage utilization and resource wastage, which gives a low rate of improvement. To solve this problem, we'll have to improve this method while adding certain parameters, such as network bandwidth. Virtual machine migration will also need to be taken into account to strengthen the model.

But also, it would be preferable to implement and compare the method that has been proposed in this work in other

RESEARCH ARTICLE

simulators used in some research works and compare the results.

5. CONCLUSION

In a cloud data center, the physical machines as well as the cooling system consume a lot of energy. The significant reduction in the number of physical machines turned on will impact significantly the energy consumption in these cloud-based environments. The optimal placement of virtual machines in a cloud computing environment is considered one of the most difficult tasks. A lot of research has been conducted to solve this type of optimization problem. Metaheuristics have become a very important solution for this type of difficult optimization problem. The GWO-inspired algorithm that has been proposed in this work is one of the nature-inspired methods. In this paper, we have implemented a metaheuristic approach called GWOVMP to provide an optimal or near-optimal solution to the VMP problem in a cloud environment. The objective of GWOVMP is to be able to decrease the large power consumption in a cloud environment while minimizing the active MPs and minimizing the waste of CPU and memory resources. The results of this work show that the proposed technique gives the optimal use of PM resources by placing the VM on a suitable PM while reducing the power consumption compared to other algorithms.

In future work, we will try to use the GWOVMP algorithm to achieve other VM placement goals in a cloud environment. It will be compared with other algorithms such as SSA (Salp Swarm Algorithm), and PSO (particle swarm optimization). Furthermore, these algorithms will be coupled with different machine-learning mechanisms and applied to the case of containers in cloud computing.

REFERENCES

[1] S. Abohamama and E. Hamouda, "A hybrid energy–Aware virtual machine placement algorithm for cloud environments," Expert Syst Appl, vol. 150, p. 113306, 2020, doi: 10.1016/j.eswa.2020.113306.

[2] S. P. M. Ziyath and S. Senthilkumar, "MHO: meta heuristic optimization applied task scheduling with load balancing technique for cloud infrastructure services," J Ambient Intell Humaniz Comput, vol. 12, no. 6, pp. 6629–6638, Jun. 2021, doi: 10.1007/s12652-020-02282-7.

[3] A. Ndayikengurukiye, A. Ez-Zahout, and F. Omary, "A Big Data-Driven Model for Secured Systems: A Blockchain-Based Architecture for Cloud Computing and IoT Security," in Advances on Smart and Soft Computing, F. Saeed, T. Al-Hadhrami, E. Mohammed, and M. Al-Sarem, Eds., Singapore: Springer Singapore, 2022, pp. 409–418.

[4] A. Ndayikengurukiye, A. Ez-Zahout, A. Aboubakr, Y. Charkaoui, and O. Fouzia, "Resource Optimisation in Cloud Computing: Comparative Study of Algorithms Applied to Recommendations in a Big Data Analysis Architecture," Journal of Automation, Mobile Robotics and Intelligent Systems, vol. 2021, no. 4, pp. 65–75, 2021, doi: 10.14313/JAMRIS/4-2021/28.

[5] S. Omer, S. Azizi, M. Shojafar, and R. Tafazolli, "A priority, power and traffic-aware virtual machine placement of IoT applications in cloud data centers," Journal of Systems Architecture, vol. 115, May 2021, doi: 10.1016/j.sysarc.2021.101996.

[6] U. Chourasia, R. Gandhi, P. Vishwavidyalaya, and D. Sanjay, "Optimal Load Balancing in Cloud using ANFIS and MGWO based Polynomial Neural Network", doi: 10.21203/rs.3.rs-529618/v1.

[7] M. S. Kumar and G. R. Karri, "EEOA: Cost and Energy Efficient Task Scheduling in a Cloud-Fog Framework," Sensors, vol. 23, no. 5, Mar. 2023, doi: 10.3390/s23052445.

[8] Y. Li, X. Lin, and J. Liu, "An improved gray wolf optimization algorithm to solve engineering problems," Sustainability (Switzerland), vol. 13, no. 6, Mar. 2021, doi: 10.3390/su13063208.

[9] J. Too, A. R. Abdullah, N. M. Saad, N. M. Ali, and W. Tee, "A new competitive binary grey wolf optimizer to solve the feature selection problem in EMG signals classification," Computers, vol. 7, no. 4, Dec. 2018, doi: 10.3390/computers7040058.

[10] E. G. Dada, S. B. Joseph, D. O. Oyewola, A. A. Fadele, H. Chiroma, and S. M. Abdulhamid, "Application of Grey Wolf Optimization Algorithm: Recent Trends, Issues, and Possible Horizons," Gazi University Journal of Science, vol. 35, no. 2, pp. 485–504, Jun. 2022, doi: 10.35378/gujs.820885.

[11] G. Shial, S. Sahoo, and S. Panigrahi, "An Enhanced GWO Algorithm with Improved Explorative Search Capability for Global Optimization and Data Clustering," Applied Artificial Intelligence, vol. 37, no. 1, 2023, doi: 10.1080/08839514.2023.2166232.

[12] N. M. Sallam, A. I. Saleh, H. Arafat Ali, and M. M. Abdelsalam, "An Efficient Strategy for Blood Diseases Detection Based on Grey Wolf Optimization as Feature Selection and Machine Learning Techniques," Applied Sciences (Switzerland), vol. 12, no. 21, Nov. 2022, doi: 10.3390/app122110760.

[13] Y. Hou, H. Gao, Z. Wang, and C. Du, "Improved Grey Wolf Optimization Algorithm and Application," Sensors, vol. 22, no. 10, May 2022, doi: 10.3390/s22103810.

[14] H. M. R. Al-Khafaji, "Improving Quality Indicators of the Cloud-Based IoT Networks Using an Improved Form of Seagull Optimization Algorithm," Future Internet, vol. 14, no. 10, Oct. 2022, doi: 10.3390/fi14100281.

[15] P. Hu, J. S. Pan, and S. C. Chu, "Improved Binary Grey Wolf Optimizer and Its application for feature selection," Knowl Based Syst, vol. 195, May 2020, doi: 10.1016/j.knosys.2020.105746.

[16] S. Azizi, M. Shojafar, J. Abawajy, and R. Buyya, "GRVMP: A Greedy Randomized Algorithm for Virtual Machine Placement in Cloud Data Centers," IEEE Syst J, vol. 15, no. 2, pp. 2571–2582, Jun. 2021, doi: 10.1109/JSYST.2020.3002721.

[17] A. Al-Moalmi, J. Luo, A. Salah, and K. Li, "Optimal virtual machine placement based on grey wolf optimization," Electronics (Switzerland), vol. 8, no. 3, Mar. 2019, doi: 10.3390/electronics8030283.

[18] S. Gharehpasha, M. Masdari, and A. Jafarian, "Virtual machine placement in cloud data centers using a hybrid multi-verse optimization algorithm," Artif Intell Rev, vol. 54, no. 3, pp. 2221–2257, Mar. 2021, doi: 10.1007/s10462-020-09903-9.

[19] A. Yousefipour, A. M. Rahmani, and M. Jahanshahi, "Improving the load balancing and dynamic placement of virtual machines in cloud computing using particle swarm optimization algorithm," International Journal of Engineering Transactions C: Aspects, vol. 34, no. 6, pp. 1419–1429, Jun. 2021, doi: 10.5829/ije.2021.34.06c.05.

[20] B. Liang, D. Wu, P. Wu, and Y. Su, "An energy-aware resource deployment algorithm for cloud data centers based on dynamic hybrid machine learning," Knowl Based Syst, vol. 222, Jun. 2021, doi: 10.1016/j.knosys.2021.107020.

[21] S. S. Sefati, M. Mousavinasab, and R. Zareh Farkhady, "Load balancing in cloud computing environment using the Grey wolf optimization algorithm based on the reliability: performance evaluation," Journal of Supercomputing, vol. 78, no. 1, pp. 18–42, Jan. 2022, doi: 10.1007/s11227-021-03810-8.

[22] J. Lu, W. Zhao, H. Zhu, J. Li, Z. Cheng, and G. Xiao, "Optimal machine placement based on improved genetic algorithm in cloud

RESEARCH ARTICLE

computing," Journal of Supercomputing, vol. 78, no. 3, pp. 3448–3476, Feb. 2022, doi: 10.1007/s11227-021-03953-8.
[23] J. Luo, W. Song, and L. Yin, "Reliable Virtual Machine Placement Based on Multi-Objective Optimization with Traffic-Aware Algorithm in Industrial Cloud," IEEE Access, vol. 6, pp. 23043–23052, Mar. 2018, doi: 10.1109/ACCESS.2018.2816983.
[24] A. Ndayikengurukiye, A. Ez-Zahout, and F. Omary, "An overview of the different methods for optimizing the virtual resources placement in the Cloud Computing," J Phys Conf Ser, vol. 1743, p. 012030, Jan. 2021, doi: 10.1088/1742-6596/1743/1/012030.
[25] H. Li, L. Wen, Y. Liu, and Y. Shen, "More than Meets One Core: An Energy-Aware Cost Optimization in Dynamic Multi-Core Processor Server Consolidation for Cloud Data Center," Electronics (Switzerland), vol. 11, no. 20, Oct. 2022, doi: 10.3390/electronics11203377.
[26] J. Wang, H. Gu, J. Yu, Y. Song, X. He, and Y. Song, "Research on virtual machine consolidation strategy based on combined prediction and energy-aware in cloud computing platform," Journal of Cloud Computing, vol. 11, no. 1, Dec. 2022, doi: 10.1186/s13677-022-00309-2.
[27] S. Nabavi, L. Wen, S. S. Gill, and M. Xu, "Seagull optimization algorithm based multi-objective VM placement in edge-cloud data centers," Internet of Things and Cyber-Physical Systems, vol. 3, pp. 28–36, 2023, doi: 10.1016/j.iotcps.2023.01.002.
[28] H. Talebian et al., "Optimizing virtual machine placement in IaaS data centers: taxonomy, review and open issues," Cluster Comput, vol. 23, no. 2, pp. 837–878, Jun. 2020, doi: 10.1007/s10586-019-02954-w.
[29] M. Ghetas, "A multi-objective Monarch Butterfly Algorithm for virtual machine placement in cloud computing," Neural Comput Appl, vol. 33, no. 17, pp. 11011–11025, Sep. 2021, doi: 10.1007/s00521-020-05559-2.
[30] T. B. Hewage, S. Ilager, M. A. Rodriguez, and R. Buyya, "CloudSim express: A novel framework for rapid low code simulation of cloud computing environments," Softw Pract Exp, 2023, doi: 10.1002/spe.3290.
[31] I. Behera and S. Sobhanayak, "Task scheduling optimization in heterogeneous cloud computing environments: A hybrid GA-GWO approach," J Parallel Distrib Comput, vol. 183, Jan. 2024, doi: 10.1016/j.jpdc.2023.104766.
[32] M. A. Khan, A. Paplinski, A. M. Khan, M. Murshed, and R. Buyya, "Dynamic virtual machine consolidation algorithms for energy-efficient cloud resource management: A review," in Sustainable Cloud and Energy Services: Principles and Practice, Springer International Publishing, 2017, pp. 135–165. doi: 10.1007/978-3-319-62238-5_6.
[33] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing," Future Generation Computer Systems, vol. 28, no. 5, pp. 755–768, May 2012, doi: 10.1016/j.future.2011.04.017.
[34] M. Masdari, S. S. Nabavi, and V. Ahmadi, "An overview of virtual machine placement schemes in cloud computing," Journal of Network and Computer Applications, vol. 66. Academic Press, pp. 106–127, May 01, 2016. doi: 10.1016/j.jnca.2016.01.011.
[35] D. Zhu, "Max–min Bin Packing Algorithm and its application in nano-particles filling," Chaos Solitons Fractals, vol. 89, pp. 83–90, Aug. 2016, doi: 10.1016/j.chaos.2015.09.024.

Authors

Aristide Ndayikengurukiye graduated with an Msc in Computer Engineering from the University of Burundi in 2017. PhD Student in Intelligent Processing Systems & Security Team (IPSS), Computer Science Department Faculty of Science of Rabat, Mohammed V University, Morocco. His research interests include Cloud computing optimization, artificial intelligence, IoT, and Big Data. He can be contacted at e-mail: lemure.dede@gmail.com.

Abderrahmane Ez-Zahout is currently an assistant professor of computer science at the Department of Computer Science/faculty of Sciences at Mohammed V University. His research is in the fields of computer sciences, digital systems, big data, and computer vision. Recently, he has worked on intelligent systems. He has served as an invited reviewer for many journals. Besides, he is also involved in NGOs, and student associations, and managing non-profit foundations. He can be contacted at email: abderrahmane.ezzahout@um5.ac.ma.

Fouzia Omary is the leader of IPSS team research and is currently a full professor of computer science at the Department of Computer Science/ENS, Mohammed V University. She is a full-time researcher at Mohammed V University and she is the chair of the international conference on cybersecurity and blockchain and also the leader of the national network of blockchain and cryptocurrency. She can be contacted at email: omary@fsr.ac.ma.

How to cite this article:

Aristide Ndayikengurukiye, Abderrahmane Ez-Zahout, Fouzia Omary, "Optimizing Virtual Machines Placement in a Heterogeneous Cloud Data Center System", International Journal of Computer Networks and Applications (IJCNA), 11(1), PP: 1-12, 2024, DOI: 10.22247/ijcna/2024/224431.

International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2024/24 Volume 11, Issue 3, May – June (2024)

RESEARCH ARTICLE

SOAVMP: Multi-Objective Virtual Machine Placement in Cloud Computing Based on the Seagull Optimization Algorithm

Aristide Ndayikengurukiye
Intelligent Processing and Security of Systems Team, Faculty of Sciences, Mohammed V University, Rabat, Morocco.
✉ lemure.dede@gmail.com

Rim Doukha
Intelligent Processing and Security of Systems Team, Faculty of Sciences, Mohammed V University, Rabat, Morocco.
doukharim@gmail.com

Eric Niyukuri
Faculty of Engineering Sciences, University of Burundi, Bujumbura, Burundi.
ericniyukuri@gmail.com

Eric Muheto
Morgan Stanley, Montréal, Canada.
eric.muheto.1@gmail.com

Abderrahmane Ez-zahout
Intelligent Processing and Security of Systems Team, Faculty of Sciences, Mohammed V University, Rabat, Morocco.
abderrahmane.ezzahout@um5.ac.ma

Fouzia Omary
Intelligent Processing and Security of Systems Team, Faculty of Sciences, Mohammed V University, Rabat, Morocco.
omary@fsr.ac.ma

Received: 17 March 2024 / Revised: 28 May 2024 / Accepted: 09 June 2024 / Published: 30 June 2024

Abstract – Virtual machine placement (VMP) involves selecting the most appropriate physical machine (PM) to run a virtual machine (VM) in cloud data centers (CDCs). Unfortunately, current VMP methods only consider limited resources, resulting in load imbalance and unnecessary activation of certain PMs in the data center (DC). This paper proposes a new approach called Multi-Objective Seagull Optimization Algorithm Virtual Machine (MOSOAVMP) to address these issue s and enhance resource management in CDCs. The aim is to optimize resource utilization, minimize energy consumption, reduce SLA violations, and improve overall DC efficiency. The aim is to achieve an optimal deployment that will meet these different objectives while minimizing the costs associated with operating the CDCs. The results show the proposed MOSOAVMP's efficiency compared with existing algorithms for the different measurements considered. These experimental results show that MOSOAVMP reduces resource wastages, and energy consumption by 5.44%, improves average CPU usage by 14.84%, memory usage by 11.54%, average storage usage by 5.37%, and average bandwidth usage by 6.88%.

Index Terms – Cloud Computing, Seagull Optimization Algorithm, Metaheuristics Algorithm, SLA, Virtual Machine Placement, Data Center, Power Consumption.

1. INTRODUCTION

In recent years, cloud computing has become a leading IT model for providing and managing services over the Internet.

RESEARCH ARTICLE

Its adoption is now ubiquitous and constantly growing, offering great scalability and various services. Cloud computing significantly impacts our daily life, particularly through social and sensor networks. The rise of smart devices has accelerated its adoption, leading to rapid growth in the number and size of cloud data centers. Additionally, it helps reduce network infrastructure costs.

Various techniques and technologies are used to meet user requirements and optimize cloud performance. These include migration, which consists of transferring a virtual machine (VM) from a physical server (PM) with insufficient resources to another with the necessary resources. What's more, virtualization is central to cloud computing. Virtualization has revolutionized IT operations in data centers (DCs), where virtual machines (VMs) are deployed on physical machines (PMs) to run users' applications [1]. Optimum placement of these VMs on physical hosts is crucial to guarantee good performance. This VMP problem is an optimization challenge whose objective is to determine the best VM allocation among the available PMs. Placement can be static or dynamic, depending on whether PMs can be replaced due to changes in the DC environment; such as variations in workload, resource availability, or hardware constraints. Migration of VMs may be necessary to balance the load, even though this may result in a breach of service level agreements (SLAs) [2]. However, it is important to avoid excessive resource utilization, as this can lead to performance degradation. The Cloud also makes it easier to store and analyze large volumes of data [3], which means that several challenges, such as safety [4], need to be considered. While cloud computing offers many advantages, it also presents several challenges. One of the most important is to reduce energy consumption in cloud-based data centers. In addition, optimizing the use of different cloud resources is also crucial. To overcome these challenges, various metrics are taken into account and analyzed.

It is therefore essential to find an optimal solution for VMP that meets the above-mentioned objectives. This implies finding a balance between the different objectives, as improving one of them may lead to the deterioration of another. A multi-objective optimization approach is therefore needed to effectively solve this VMP problem.

This article presents a new algorithm, designed to address the challenge of optimizing the placement of VMs on PMs to improve IT resource management. The proposed algorithm has three main objectives. The first objective focuses on reducing energy consumption in CDCs by minimizing the number of active physical servers. The second objective is to reduce the waste of various cloud resources while optimizing the VMP. Finally, the algorithm aims to minimize and reduce the SLAs among active PMs in CDCs. In addition, the algorithm considers other factors such as CPU usage, memory, bandwidth, storage space, as well as the number of active machines and migrations.

The use of this solution significantly reduced the number of migrations to CDCs. The proposed MOSOAVMP algorithm was compared with other commonly used algorithms such as DMOSCA-SSA [4], MOILP [5], PIAS [6], MGGAVP [7], and MBFD [8]. As demonstrated earlier in the summary, the proposed approach demonstrated substantial improvements over the aforementioned methods. The metrics used for this comparison include, among others, energy consumption, utilization of various resources (CPU, RAM, storage, and network), migrations, as well as compliance with service level agreements (SLAs).

The remainder of this article is organized as follows: The second section reviews the various algorithms, instead of works. Section 3 gives a general introduction to the various mathematical concepts of the Seagull Optimization Algorithm (SOA), and is followed by the mathematical formulation of the multi-objective optimization applied to VMP, and the description of the proposed algorithm is found in section 4. Section 5 presents the results of the performance evaluation of the proposed method in comparison to other commonly used algorithms. Finally, section 6 provides conclusions and outlines prospects for future work.

2. RELATED WORK

As mentioned above, VMP aims to ensure good cloud performance and optimal resource management. There are many algorithms available to solve this problem, and in this section, we review some of the work that addresses the VMP problem.

Lu et al. [9] proposed an improved genetic algorithm (I-GA) to address the VMP problem, aiming to optimize the availability and energy consumption of CDCs. A combination of virtual hierarchy architecture and GA was utilized to achieve a near-optimal solution, to improve energy consumption and resource availability. A key step in their approach is the generation of the initial I-GA population, which is achieved using finite element analysis in the background. CloudSim is employed for simulating the experiments. The results demonstrate a significant improvement in energy management and high availability in the DCs. Their model presents better results based on the benchmark results of the different state-of-the-art methods used for the VMP optimization problem. The authors efficiently optimize the VMP by utilizing the I-GA approach, contributing to better resource management and reduced energy consumption in CDCs.

Caviglione et al [10] proposed a multi-objective approach to determine the best techniques for VMP. It takes into account the impact of hardware failures, DC power consumption, and user satisfaction in terms of performance. A deep

RESEARCH ARTICLE

reinforcement framework facilitates the selection of the best heuristic for the placement of VMs. Their results demonstrate that the method exceeds the performance of current state-of-the-art heuristics for both real and synthetic workloads [10].

Alharbi et al [11] introduced a method combining the assignment of different applications and the VMP of a DC to optimize the energy efficiency of enterprise distribution centers. They formulated the problem related to energy optimization of enterprise DCs as an Int2LBP (Integrated Two-Layer Bin Packing) problem. This approach was considered as an initial solution and was further developed to optimize the energy efficiency of corporate distribution centers. In terms of energy efficiency, the various simulations carried out on DC prove the performance of the Int2LBP_FFD algorithm compared with the Consec2LBP_FFD algorithm. In addition, the Int2LBP_ACS algorithm outperforms the Int2LBP_FFD in efficient energy management. The Int2LBP_FFD and Int2LBP_ACS algorithms are well suited to managing different applications and VMs on DCs for large enterprises.

Ghetas et al. [12] introduced a new method called MBO-VM based on the VMP Monarch Butterfly Optimization (MBO) algorithm. This method aims to maximize packaging performance and reduce PMs. The CloudSim simulator was utilized to implement and assess the performance of this approach. This tool also enabled them to test real and synthetic workloads in the cloud. The results obtained from the simulations demonstrate that MBO-VM outperforms known VMP techniques in terms of performance. By using MBO-VM, it is possible to more effectively reduce the number of active hosts while maximizing packaging efficiency. This approach thus offers an optimal solution for VMP, improving the overall efficiency of CDCs.

To solve the VMP optimization problem, a new multi-objective ILP algorithm has been presented by Regaieg et al [5], considering that at cloud service providers (CSPs) the environment is made up of VMs of homogeneous and heterogeneous types. This model aims to reduce resource wastage while maximizing the number of VMs that are hosted on physical servers, thereby reducing the number of PMs used. Due to the diversity of the different types of VMs available in a heterogeneous cloud environment, the test results showed the effectiveness of achieving the above objectives. These results have the advantage of reducing the costs associated with operating DCs while maintaining a high level of Quality of Service (QoS).

Rashida et al. [13] presented a VMP algorithm called MGGAVP, which considers correlation and addresses the VMP problems. This algorithm is based on escalation algorithm hybridization and genetic clustering and is extended to operate in a multi-cloud environment. After simulation, the results reveal the net performance offered by their algorithm to those of the other benchmark algorithms. It achieves energy savings of 51.93% and reduces energy costs by 70.41%.

Rahimi Zadeh et al. [6] proposed a new scheduling scheme called PIAS (Profit-aware Interference-aware Scheduling) for the efficient consolidation of VMs in the Infrastructure-as-a-Service (IaaS) model. The PIAS scheme takes into account several factors such as profit, energy costs, operational interference, resource utilization, and SLAs. To minimize workload execution costs and increase vendor profit, the optimization problem was modeled in the form of stochastic dynamic programming, mimicking the operational behavior of VMs. Simulations based on real workloads show that PIAS outperforms competing approaches on average, with improvements of at least 40% for profit, 29% for energy efficiency, and 35% for service downtime.

To improve quality of service (QoS) and efficient traffic management in Internet of Things (IoT) networks, a meta-heuristic based on enhanced seagull optimization is proposed by Gharehpasha et al. [14]. This approach enables better management of packet forwarding and a significant improvement in QoS in terms of delay. The performance of this method has been evaluated and compared with previous methods, demonstrating its accuracy, efficiency, and superiority.

Nabavi et al. [15] presented a multi-objective approach to VMP in edge-cloud DCs. To optimize network traffic and traffic power, an SOA-based model was presented. The strategy focuses on trying to reduce network traffic between PMs while consolidating VM communications on the same PMs. This reduces data transfer across the network and lowers PMs power consumption. The authors conducted simulations using CloudSim and tested the proposed approach on two network topologies, VL2, and three-tier. The results highlight the proposed method's effectiveness in significantly decreasing energy consumption and network traffic in edge-cloud environments. Test results for the proposed algorithm show a clear reduction in energy efficiency of 5.5%, a remarkable 70% reduction in network traffic, and an 80% reduction in the energy consumption of network components.

Table 1 provides a summary of these various state-of-the-art works. It presents the methodology used, the objectives, the weaknesses, the simulator used, and the proposed method.

The choice of the MOSOAVMP algorithm to optimize the placement VMs in the cloud is based on the inherent qualities of the SOA algorithm. These include its simplicity, ease of implementation, and ability to converge rapidly on optimal solutions, even with many iterations. This testifies to its robustness and ability to solve complex problems such as VM placement. The analysis presented in Table 1 reveals a thorough evaluation of the metrics used in previous work, enabling us to discern which metrics are paramount and

RESEARCH ARTICLE

which are often overlooked. This analysis enabled us to select a wide range of metrics relevant to our study, ensuring a comprehensive and rigorous assessment of cloud performance.

Table 1 Summary of the Work Reviewed

Algorithm	Vulnerability	Objectives	Methodology Applied	Simulator
I-GA [9]	Very long runtimes and certain metrics such as storage and bandwidth are not taken into use.	Ensure availability and optimize the energy consumption of a DC	Proposal of an I-GA algorithm to optimize VMP problems.	CloudSim
DLR-VMP [10]	Resource wastage	VM placement while ensuring Quality of Experience (QoE), by reducing the power used.	Placement strategies are determined through a deep reinforcement learning framework that identifies the optimal placement heuristic for each VM within the workload.	Python
ACS, Int2LBP [11]	Runtime execution, applications are allocated dynamically, dynamic VMP, VM consolidation, and VM migration	Implementation of the Int2LBP_FFD algorithm and the Int2LBP_ACS algorithm, based on the initial results of the first algorithm, to optimize the energy efficiency of corporate distribution centers.	To solve the problem of energy consumption in an enterprise DC, the authors call it Int2LBP	Amazone EC2 T2
MBO-VM [12]	Storage, Bandwidth, and SLA	Minimize the number of active physical hosts in the DC to optimize energy consumption and lower maintenance costs.	Server consolidation and resource utilization	CloudSim
MOILP [5]	The effect of workload on efficient energy consumption is not taken into account.	Minimize DC operating costs to increase customer satisfaction.	Optimizing hosted VMs, limiting the waste of different resources, and reducing the number of active physical servers to reduce energy consumption in DCs.	Amazone EC2, Julia Language (JL)
MGGAVP [13]	QoS, SLA	Hybridization of metaheuristic escalation algorithms and genetic clustering for optimization in a multi-cloud environment	Optimization of the VMP problem in a heterogeneous environment.	MATLAB R2015
PIAS [6]	Lack of peak period prediction to increase or reduce DC capacity during peak or idle periods. To avoid saturation, VM resource utilization should remain below.	Cost of energy efficiency, operational VM interference, reduced resource utilization, and SLA optimization.	VM behavior is modeled using stochastic dynamic programming to minimize workload execution costs and maximize provider profit.	CloudSim
ISOA [14]	Many of the metrics (e.g. CPU, RAM, storage, ...) to assess the efficiency of the cloud have not been taken into account.	improve QoS and efficient traffic management in Internet of Things (IoT) networks.	Use SOA to minimize lead times, average implementation times, virtual server computing costs, and network costs.	MATLAB R2017

RESEARCH ARTICLE

SOAVMP [15]	Little use of test resources, SLA, and resources wastage	Traffic, power, and energy	Reduced energy consumption thanks to consolidation technology and reduced network traffic between PMs by concentrating as many VMs as possible on a single PM.	CloudSim
Our method	Security and task scheduling	Energy consumption, SLA, resource utilization(CPU, RAM, Storage, and Bandwidth), Migration, and resource wastage.	MOSOAVMP optimizes virtual machine placement	CloudSim

3. SEAGULL VIRTUAL MACHINE PLACEMENT OPTIMIZATION ALGORITHM

3.1. Bio-Inspired Paradigm

Optimization in general is a field that has been developing steadily over the last few years, with new bio-inspired algorithms [16], [17], [18] as well as hybrid algorithms [19], [20]. Inspired by the natural migration and attack patterns of seagulls in the wild, SOA is a metaheuristic that was proposed by Dhiman and Kumar in 2017[21]. This algorithm is population-based and uses migratory movements to search for abundant food sources. Their positions are updated according to the best position found, while fascination and attack strategies are used to attract prey. The algorithm is divided into diversification and intensification, the two main phases, and can switch from one phase to the other depending on the position of the prey. Seagulls can continuously modify their angle of attack during migration [22].

The migratory behavior of seagulls is characterized by the following features [23]:

- Seagulls move in groups during migration, adopting a collective movement strategy to avoid collisions between them;
- Within the group, seagulls can adjust their trajectory by moving in the direction that ensures the best survival for the oldest seagull. In other words, they may follow the direction chosen by the most experienced seagull to maximize their chances of survival.

This noisy bird lives, feeds, and sleeps in colonies. Its food consists of insects, their larvae, earthworms, small crustaceans, mollusks, and small fish caught in flight. As shown in Figure 1, during the attack they perform a spiral movement.

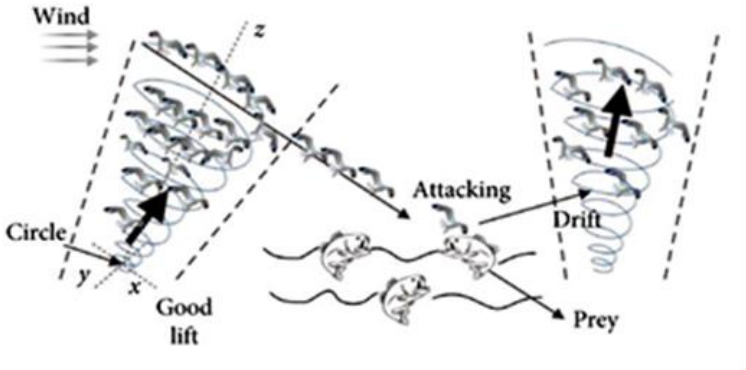

Figure 1 Migration and Attack Phases of Seagull Behavior [21]

3.2. Mathematical Formulation

This algorithm simulates the movement of a group of seagulls from one location to another during migration. During this migration phase, three conditions must be met by each of the seagulls:

3.2.1. Migration phase (or Exploration phase)

Collision Avoidance: To prevent collisions with other seagulls when computing the new position of the search agent, a variable named A is introduced. [24].

$$\overrightarrow{C_s} = A \times \overrightarrow{P_s}(x) \qquad (1)$$

In Equation (1), $\overrightarrow{C_s}$ is a representation of the search agent's position that avoids collisions between neighbors, P_s represents its current position, x designates the current iteration, and A characterizes the agent's various movements within the search space. Figure 2 shows how search agents avoid colliding with each other. These variables are applied to compute the optimal position for the search agent while

International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2024/24 Volume 11, Issue 3, May – June (2024)

RESEARCH ARTICLE

avoiding collisions with other agents[24], [25]. This is shown in Equations (2) and (3).

$$A = f_c - \left(x \times \left(\frac{f_c}{Max_{it}} \right) \right) \qquad (2)$$

$$x = 0,1,2,\ldots, Max_{it} \qquad (3)$$

In Equation (2), variable A reproduces the motion behavior of the search agent. The parameter f_c controls this variable and decreases nearly from f_c to 0. The maximum number of iterations used is represented by Max_{it}. These elements are essential for regulating the agent's movement and defining the limits of the optimization process. Figure 3 shows how the different agents move and search for the best neighbor.

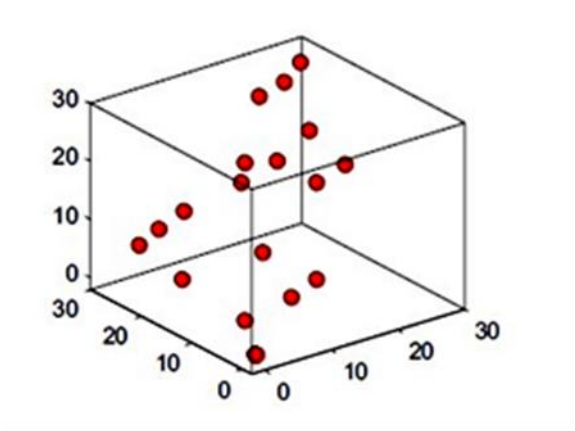

Figure 2 Avoiding Search Agent Collisions

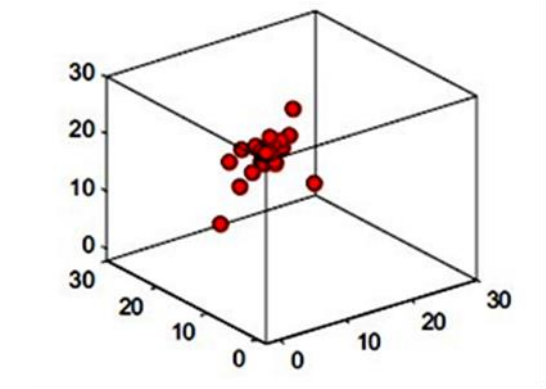

Figure 3 Moving Agents and Finding the Best Neighbor

Seagulls head for the best-performing seagull in their group, ensuring cohesion. This is represented in the mathematical equation (4)[26], [27]:

$$\overrightarrow{M_s} = B \times \left(\overrightarrow{P_{bs}}(x) - \overrightarrow{P_s}(x) \right) \qquad (4)$$

Where $\overrightarrow{M_s}$represents the position of the $\overrightarrow{P_s}$ of gulls in relation to the P_{bs} of the best-performing seagulls. Control parameter B, as indicated by reference [28], plays a pivotal role in establishing equilibrium between diversification and intensification. It is represented mathematically in Equation (5)[27],[29] where Ran denotes a random number within the range of [0; 1][28]:

$$B = 2 \times A^2 \times Ran \qquad (5)$$

Approaching the Optimal Seagull: Seagulls actively seek out food sources during their migration, moving in the direction that offers the best opportunities for survival. Figure 4 illustrates the updating of the search agent's position relative to the position of the best search agent.

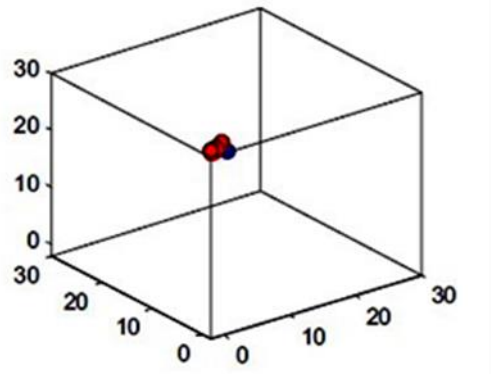

Figure 4 Approaching the Optimal Search Agent

In this case, updates of other seagull's positions rely on the position of the best seagull. The Equation (6) below represents this update [30]:

$$\overrightarrow{D_s} = \left| \overrightarrow{C_s} + \overrightarrow{M_s} \right| \qquad (6)$$

3.2.2. Attack Phase (or Exploitation Phase)

In the exploitation phase, the seagulls utilize their previous knowledge and experience. They can constantly adjust their angle of attack and speed. To maintain altitude, they use their wings and body weight.

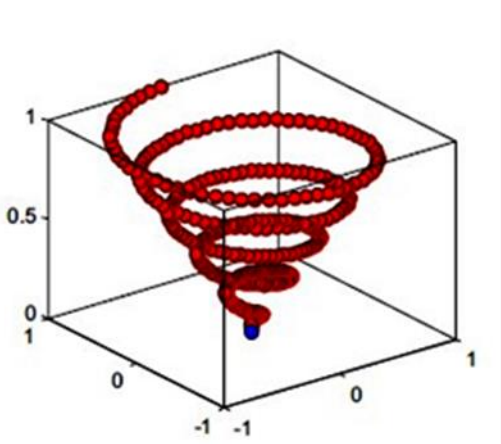

Figure 5 Natural Behavior of Seagull Attack

RESEARCH ARTICLE

As illustrated in Figure 5, the seagulls engage in a spiral motion while attacking their prey in the air. The Equations (7)-(10) below, calculate the three dimensions (x, y, z) of this behavior [31], [32], [33]:

$$x = r \times \cos(\theta) \quad (7)$$
$$y = r \times \sin(\theta) \quad (8)$$
$$z = r \times \theta \quad (9)$$
$$r = v \times e^{\theta u} \quad (10)$$

Updating the search agent's position within the spiral involves the utilization of various parameters. The radius of the spiral is represented by r, while θ is a random element between$[0 < \theta < 2\pi]$[34]. The parameters and u influence the shape of the spiral. Using the natural logarithm of base e, the following Equation (11) calculates the updated position of the search agent [35], [36]:

$$\overrightarrow{P_s}(x) = (\overrightarrow{D_s} \times x \times y \times z) + \overrightarrow{P_{bs}}(x) \quad (11)$$

The $\overrightarrow{P_s}(x)$ variable in this equation plays an essential role in maintaining the best available option and taking into account the current situation of the other search agents. It maintains a reference to the best solution throughout the process.

4. MATHEMATICAL FORMULATION AND MOSOAVMP FOR VIRTUAL MACHINE PLACEMENT

4.1. Problem Formulation

The optimal placement of VMs in CDCs is pivotal in cloud computing. The primary aim is to minimize energy consumption, resource wastage, SLA violations, and maximize DC efficiency.

Cloud computing presents complex challenges when it comes to VMP. The issue of VMP in a CDC is unpredictable and follows no consistent pattern. The complexity of this problem can be illustrated by the fact that the maximum number of VM mappings on PMs is equal to m^n , where n is the number of VMs and m is the number of PMs. With this context, we determine the first objective, which is to minimize the energy consumption of the PMs in the CDC. The various mathematical symbols used in the mathematical formulation are described in Table 2. Recent research has shown a linear correlation between CPU usage and server energy consumption in CDC. A linear equation can be used to precisely calculate energy consumption in the context of cloud computing. This relationship shows that when CPU utilization is increased, host power demand increases proportionally [37], [38], [39]:

$$P_i^{power} = \begin{cases} (P_i^{busy} - P_i^{iddle}) \times U_i^{cpu} + P_i^{iddle} & if\ U_i^{cpu} > 0 \\ 0, & otherwise \end{cases} \quad (12)$$

Equation (12) is used to determine the energy consumption of a PM in a DC. These different variables P_i^{power} , P_i^{busy} , P_i^{iddle} and U_i^{cpu} represent, respectively, the amount of energy consumed when the machine is fully loaded, idle, and inactive, and CPU utilization in MIPS. According to this equation, power consumption is directly proportional to CPU utilization. As a result, an increase in CPU will lead to an increase in power consumption in the DCs.

Energy consumption in CDCs is directly linked to CPU power consumption. The increase in the CPU usage of the PMs will directly influence the increase in the energy consumption of the CDCs. Consequently, the overall energy consumption in the CDCs can be calculated based on the following Equation (13):

$$\sum_{i=1}^{m} P_i^{power} = \sum_{i=1}^{m} b_i \times ((P_i^{busy} - P_i^{idle}) \times \sum_{v=1}^{m} a_{jp}.C_j + P_i^{idle} \quad (13)$$

Table 2 Various Mathematical Symbols and Descriptions

Symbol	Descriptions
P	Group of PMs
V	Group of VMs
α	F1 balance coefficient
β	F2 balance coefficient
δ	F3 balance coefficient
a, b	Binary variables
P_i^{iddle}	Power consumption in inactivity mode
P_i^{power}	Power consumption of PM i
P_i^{busy}	Energy consumption of PMs during periods of activity
U_i^{cpu}	The normalized CPU utilization of PM i
U_i^{memo}	The normalized RAM utilization of PM i
$R_i^{wastage}$	Inefficient use of PM resources
NR_i^{cpu}	Normalized remaining CPU of PM
NR_i^{memo}	Normalized remaining memory of PM
TU^{CPU}	Global CPU usage PMs in operation
TU^{memo}	Global memory usage PMs in operation
TU^{sto}	Global storage usage PMs in operation

RESEARCH ARTICLE

TU^B	Global bandwidth usage PMs in operation
T_i^{cpu}	Peak CPU usage of PMs
T_i^{memo}	Peak memory usage of PMs
T_i^{sto}	Peak storage usage of PMs
T_i^{band}	Peak bandwidth usage of PMs
C_j	CPU request of VM j in MIPS
M_j	Memory request of VM j in MB
H_j	Storage request of VM j in GB
B_j	Bandwidth request of VM j in Mbps

Preventing wasted resources is a crucial aspect of placing VMs in CDCs. Each server has hardware resources that must be optimally utilized to host VMs. Efficient management of unused server resources is essential. Equation (14) quantifies the waste of resources [40].

$$R_i^{wastage} = \frac{| NR_i^{cpu} - NR_i^{memo} |}{| U_i^{cpu} + U_i^{memo} |} + \varepsilon \quad (14)$$

Equation (14) is used to quantify the remaining unused resources on a PM p in a CDC. NR_i^{cpu}p represents the amount of unused CPU, while NR_i^{memo}p represents the amount of memory unused by VMs on that same PM i. The variables U_i^{cpu} and U_i^{memo} represent CPU and memory usage, respectively, by VMs on the PM i in the CDC. Equation (15) represents the total quantity of the various resources consumed in (CDC).

$$\sum_{i=1}^{m} R_i^{wastage} = \sum_{p=1}^{m} \left[a_i \times \frac{\left| (T_i^{cpu} - \sum_{j=1}^{n} (b_{ji}.C_j)) - (T_i^{memo} - \sum_{j=1}^{n} (b_{ji}.B_j)) \right| + \varepsilon}{(\sum_{j=1}^{n} (b_{ji}.C_j) + (\sum_{j=1}^{n} (b_{ji}.M_j)} \right] \quad (15)$$

When the values a_i and b_{vi} are equal, this indicates that the PM P is active and hosting the VM v.

To guarantee a minimum level of service between the cloud service provider and the customer, an SLA contract must be respected. Equation (16) represents this SLA:

$$SLA = \frac{1}{1 + e^{U_{cpu} - 0.9}} \quad (16)$$

The objectives of placing VMs on PMs in CDCs are defined from the previous equations, and include the following:

$$Min\ F(x) = \alpha F_1(x) + \beta F_2(x) + \delta F_3(x) \quad (17)$$

The objective function used in Equation (17) takes into account several objectives in the placement of VMs. The coefficients α, β and δ keep these objectives in balance. The first objective, F1, concerns the energy consumption of the PMs. The second objective, F2, measures resource wastage. Finally, the third objective, F3, takes into account SLA compliance. The values of these functions can be determined through the following Equations (18)-(20):

$$Min \sum_{i=1}^{m} P_i^{power} \quad (18)$$

$$Min \sum_{i=1}^{m} R_i^{wastage} \quad (19)$$

$$Min \sum_{i=1}^{m} SLA \quad (20)$$

When optimally placing VMs on PMs in a CDC, several constraints must be taken into account, including:

$$\sum_{i=1}^{m} b_{ji} = 1 \quad (21)$$

$$\sum_{j=1}^{m} b_{ji} = b_{ij}.C_j \le T_i^{cpu}.a_i TU^B \quad (22)$$

$$\sum_{j=1}^{m} b_{ji} = b_{ij}.M_j \le T_i^{memo}.a_i \quad (23)$$

$$\sum_{j=1}^{m} b_{ji} = b_{ij}.H_j \le T_i^{sto}.a_i \quad (24)$$

$$\sum_{j=1}^{m} b_{ji} = b_{ij}.B_j \le T_i^{band}.a_i \quad (25)$$

Each VM is assigned to a single PM, as shown in equation (21). Equations (22), (23), (24) and (25) ensure that the cumulative of resources (CPU, memory, storage space, and bandwidth) demanded by the various VMs on a PM cannot exceed the capacity of that PM.

4.2. Proposed Algorithm

In CDCs, a collection of homogeneous or non-homogeneous VMs utilize the hardware resources of the PMs to run their different services. Hosting these VMs is the responsibility of the PMs in these centers. More hardware resources are needed to meet the growing request for cloud services, and this is driving up costs. Optimizing VMP will maximize the use of available resources while enabling each PM to host a significant number of VMs. This approach avoids wasting the various resources available in the cloud and also makes the most of the processing power, memory, and storage systems

RESEARCH ARTICLE

of the PMs. This guarantees the performance of a cloud-based DC.

This study presents a new solution based on MOSOAVMP, a discrete multi-objective approach to optimal VMP. The main objective is to optimize resource management, minimize energy consumption, and reduce the SLA in the CDC. Algorithm 1 shows the solution proposed in this work.

Input: Initializing the agent population

Output: The best solution for optimizing VMP

Begin

1. Initialize SOA Parameters
2. Set the population size of the Seagulls
3. Compare the cost function of each agent
4. Determine the maximum number of iterations
5. Initialize the seagull position randomly

While t < maxiter do

For each search agent do

Apply mutation and crossover on the solutions as migration and seagull behavior

Calculate objectives values for all search agents

End for

Update seagull position

For each i=1 to number of seagulls do

Compute the costs involved with using all search agents

Generate the most effective solutions using the latest search agents

End for

Consider the most efficient solution.

End While

Return the best solution

Algorithm 1 MOSOAVMP

The first phase of all metaheuristic algorithms is the random phase called initialization, where random solutions are generated within the search space. This phase significantly influences the algorithm's efficiency and final results. To achieve an optimal solution, a specific number of agents is defined. In this version of AOS, which is proposed in Algorithm 1, the various steps are as follows:

Step 1: This is the crucial step of initializing the parameters taken into account in this algorithm. It involves mapping VMP in the CDC to the seagull and resetting the VM position. The maximum number of iterations, the number of agents, and the dimensions of the space are taken into account during initialization;

Step 2: This is the first phase in the search for the optimal solution. This solution is obtained as a function of resource constraints, using the equation representing the objective function. The solution with the smallest value is then considered to be the optimal solution;

Step 3: This step involves updating the seagull's position;

Step 4: The search process is interrupted if the maximum number of iterations is reached. At this point, the position of the leading agent is considered. If the maximum iterations are not yet reached, the process returns to step 2.

5. RESULTS AND DISCUSSIONS

5.1. Performance and Evaluation

This section presents the configuration used, the various performance measurements, as well as the different simulation results of the new algorithm(MOSOAVMP), evaluated and compared against various existing algorithms dealing with the VMP problem such as DMOSCA-SSA MOILP, MBO-VM, PIAS, MGGAVP, and MBFD.

The main objective is to perform the optimal placement of VMs on PMs in a Cloud environment to limit resource wastage, energy consumption costs, and SLA, and maximize total performance. In this study, we took into account several key evaluation metrics such as resource wastage, energy consumption, and utilization of different resources (CPU, memory, storage, and bandwidth). Added to this are the number of active machines and migration.

The experimental study of the MOSOAVMP algorithm was carried out using the CloudSim simulator version 5.0[41]. The CloudSim simulator supports the modeling and simulation of a cloud-based DC [42]. It was chosen for its flexibility in implementing many IaaS functionalities such as energy management, and resource management, as well as evaluating new applications before implementation in a real environment.

The experiments were carried out on a computer equipped with an AMD Ryzen 5800U processor clocked at 4.4 GHz with 8 cores and logical processors equal to 16, plus 16 GB of 3200MHz RAM, running the Windows 11 operating system.

To conduct simulations of the heterogeneous Cloud environment, a single DC is established, comprising 1800 PMs with two configurations, as outlined in Table 3.

This is done for a set ranging from 200 to 2,000 VMs, with an interval of 200 VMs for each iteration. The specifications of the VMs are detailed in Table 3.

RESEARCH ARTICLE

Table 3 Different Test Configurations

		Host name	Parameter and value
Host Type	PMs	HP ProLiant ML110 G3	MIPS: 3000 RAM: 4GB Cores: 2 Storage: 1024 GB Bandwidth: 3Gbps
		HP ProLiant G4	MIPS: 3720 RAM: 4GB Cores: 2 Storage: 1024 GB Bandwidth: 3Gbps
	VMs	-	MIPS: 2000 RAM: 1024MB Cores: 1 Storage: 2.5GB Bandwidth: 512MB

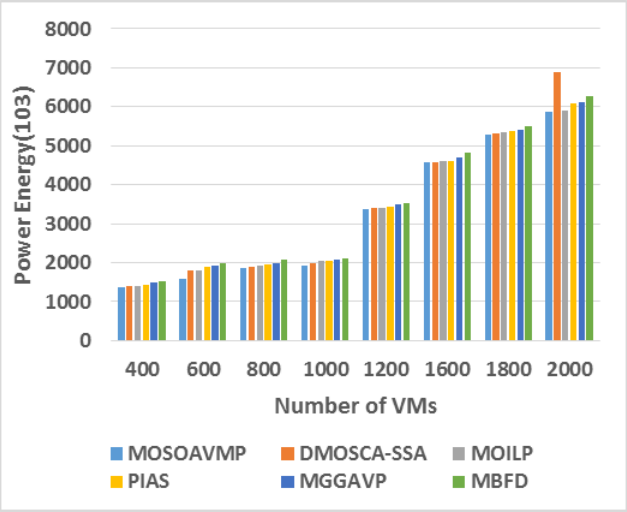

Figure 6 Power Energy Consumption

Figure 6 shows the energy consumption results for the algorithms used in the tests. Energy consumption increases with the number of VMs active in the CDC. In all cases, MOSOAVMP shows a clear improvement in energy consumption over the other algorithms compared. Considering the basic configuration of 400 VMs, the proposed MOSOAVMP model shows a low power consumption of 5.28% on average, compared with the state-of-the-art DMOSCA-SSA, MOILP, PIAS, and MGGAVP algorithms. For the maximum configuration of 200 VMs, MOSOAVMP demonstrates a 6.18% improvement over the other models evaluated. Overall, the proposed algorithm achieves a 5.44% reduction in energy consumption compared to other advanced algorithms.

Figure 7 shows VM migration comparisons. The proposed method shows the performance and stability of migration numbers. To migrate these VMs, we consider the resources available on the physical servers. In the case of state-of-the-art algorithms, VM correlation, and sudden resource changes are not taken into account, generating a large number of VMs that need to be migrated. The proposed algorithm limits VM migration while respecting the constraints since it is based on resource correlation. To achieve the best performance, the proposed algorithm minimizes the competition that can arise between VMs hosted on the same host by selecting the most appropriate VMs and PMs.

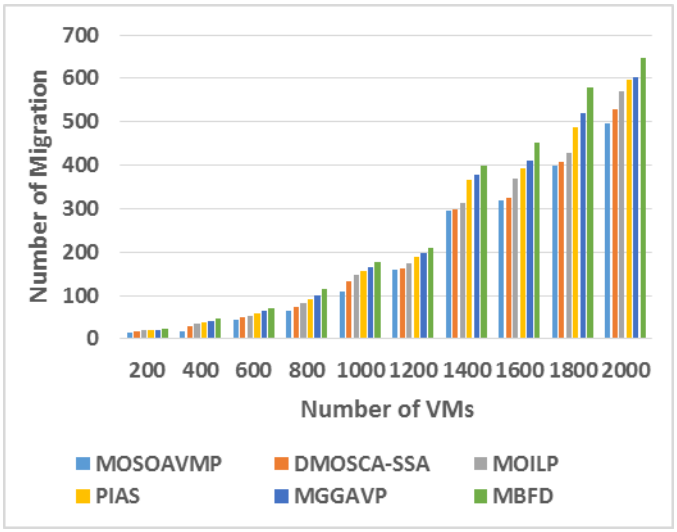

Figure 7 Number of Migration

The results in Figure 7 show the effectiveness of MOSOAVMP with the lowest migration costs, followed by DMOSCA-SSA, MOILP, PIAS, MGGAVP, and MBFD. In general, VM migration involves a considerable amount of data transfer. To obtain an optimal VMP, we reduce the number of migrations while placing VMs on the minimum possible PMs. In this way, the proposed algorithm reduces the number of active PMs and network links, while prioritizing the placement of VMs. In this way, the algorithm will reduce the number of active servers while prioritizing the placement of correlated VMs on the same server or neighboring servers.

Figure 8 shows the results for the number of active PMs in proportion to the number of VMs hosted on them. The test results show that the proposed MOSOAVMP algorithm

RESEARCH ARTICLE

requires fewer physical hosts to host VMs than the state-of-the-art algorithms. This improvement becomes significant as the number of VMs assigned to it increases. On average, MOSOAVMP improves by 7.92% over the compared algorithms. For example, let's compare MOSOAVMP with DMOSCA-SSA, the second best-performing algorithm. We see an improvement of 5.22% when using a configuration of 200VMs for both algorithms and an improvement of 1.22% in the case of 2000VMs. In the case of the MBFD algorithm, which presents the weakest results, MOSOAVMP shows a clear improvement of 19.92% for the 200VMs configuration and 8.10% for the 200% configuration. For all test configurations, MOSOAVMP performed well against DMOSCA-SSA, MOILP, PIAS, MGGAVP, and MBFD.

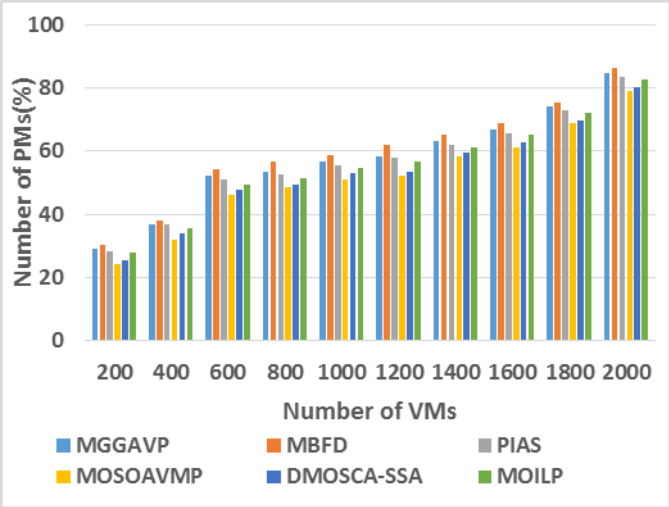

Figure 8 Number of Active PMs

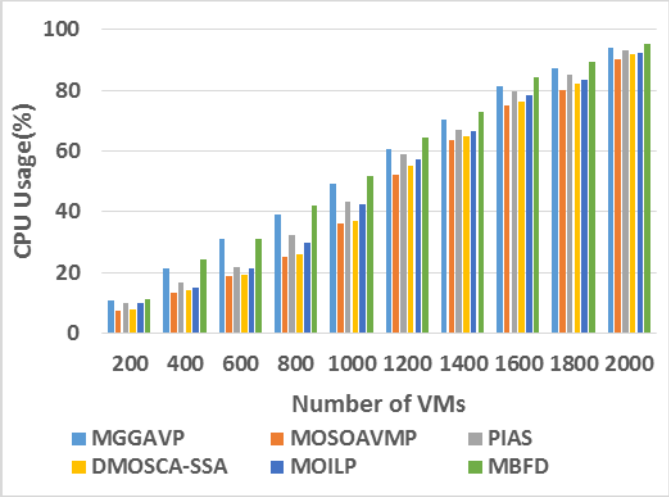

Figure 9 Average CPU Usage

Figure 9, Figure 10, Figure 11, and Figure 12 represent respectively the average use of PM resources such as CPU, memory, storage, and bandwidth as a function of active VMs in the DC. The results demonstrate the effectiveness of the proposed method in all situations. For CPU usage, MOSOAVMP shows an average improvement of 14.84% compared to state-of-the-art algorithms, followed by DMOSCA-SSA. For memory, MOSOAVMP shows an 11.54% improvement, a 5.37% improvement in storage space, and a 6.88% improvement in bandwidth.

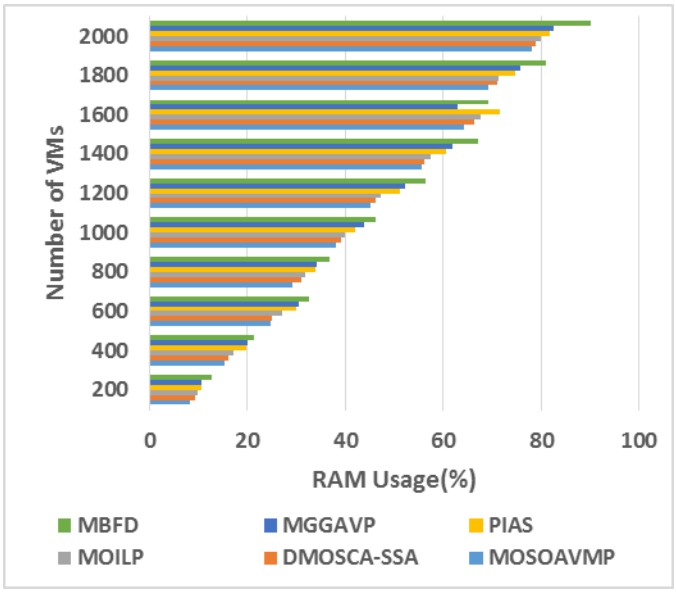

Figure 10 Average RAM Usage

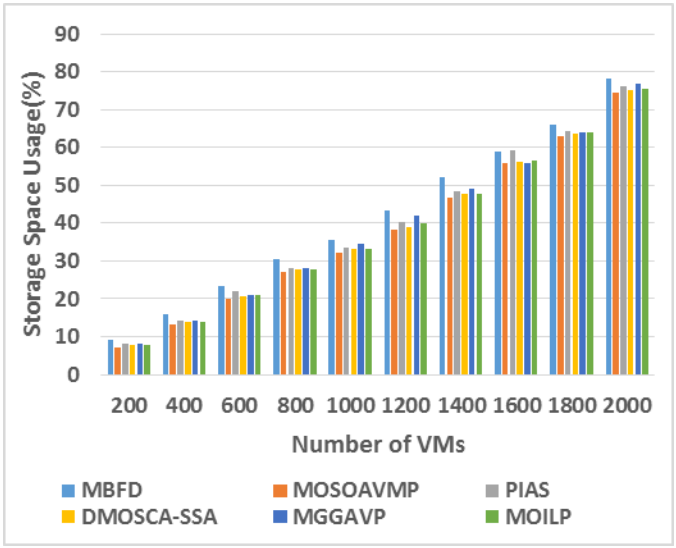

Figure 11 Average Storage Usage

Maintaining effective resource management in a cloud environment is a primary objective. Resource wastage significantly impacts cloud computing performance, as highlighted in the other metrics discussed earlier. According

RESEARCH ARTICLE

to the findings illustrated in Figure 13, the proposed algorithm effectively reduces resource wastage in the cloud when compared to existing algorithms.

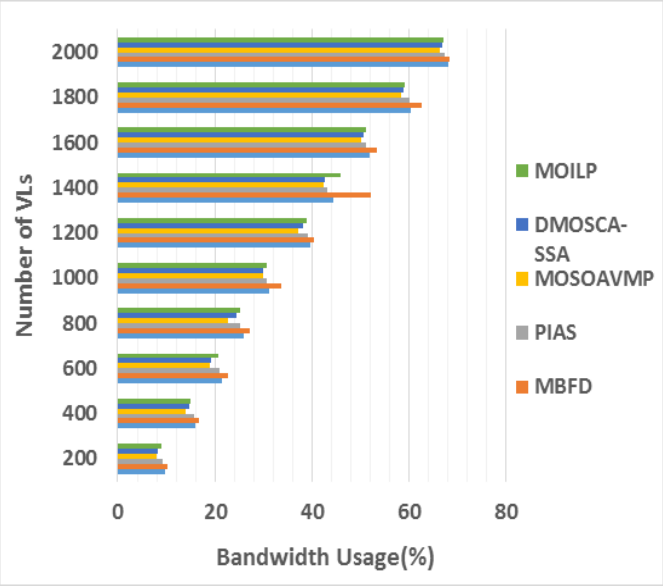

Figure 12 Average Bandwidth

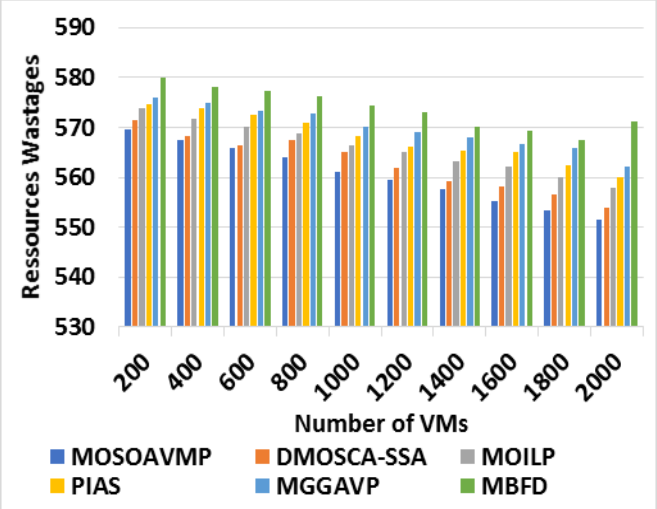

Figure 13 Resources Wastages

SLA reduction translates into a more reliable user experience and customer satisfaction. Figure 14 shows the effectiveness of the proposed MOSOAVMP algorithm when managing SLA in cloud computing, compared with DMOSCA-SSA, MOILP, PIAS, MGGAVP, and MBFD.

The various previous fluctuations observed with the existing algorithms have been reduced. As shown in Figure 14, the results demonstrate the superior performance of MOSOAVMP compared to state-of-the-art algorithms.

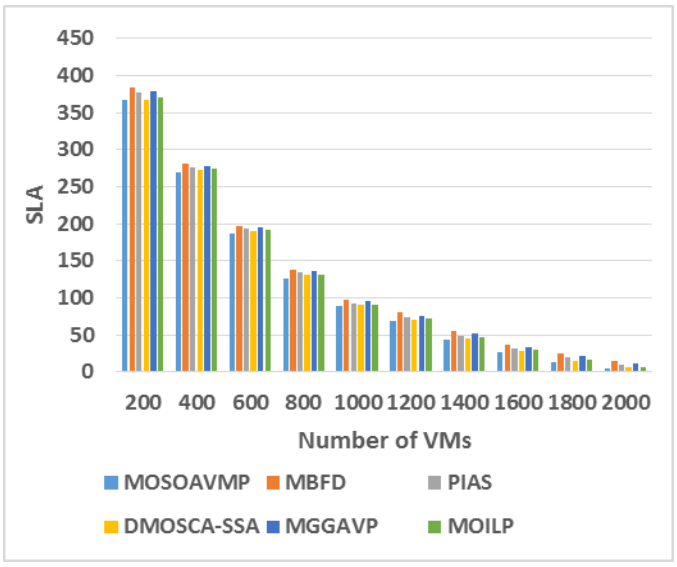

Figure 14 SLA

The convergence time for finding an optimal solution to various problems is crucial for metaheuristics. This includes the time elapsed during both the exploration and exploitation phases. To evaluate the performance of MOSOAVMP against state-of-the-art algorithms, the number of iterations is set to 100. Figure 15 illustrates the performance of the proposed method in comparison to other algorithms. As the number of iterations increases, the fitness of the DMOSCA-SSA, MOILP, PIAS, MGGAVP, and MBFD algorithms decreases. Under the same conditions, the MOSOAVMP algorithm demonstrates high efficiency. Increasing the number of iterations allows the proposed method to converge faster to the optimal solution than the other algorithms. Like any metaheuristic, it must balance between the local optimum value and the global optimum value throughout the search for the best solution. This balance determines MOSOAVMP's ability to effectively solve complex problems, particularly VMP problems.

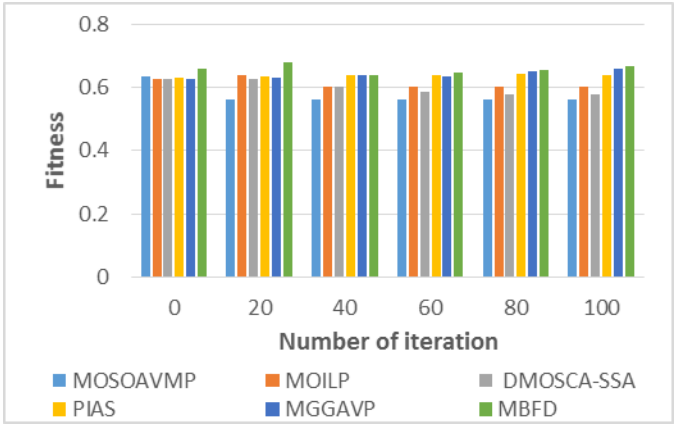

Figure 15 Convergence Time

RESEARCH ARTICLE

Figure 16 shows the comparison between the time of execution of the method that is proposed and the state methods for finding an optimal solution for VMP. To carry out this comparison, a range from 100VMs to 500VMs is used to see the evolution of execution time. Referring to Figure 15, MOSOAVMP is better at finding the optimal VMP solution at different problem sizes than other algorithms.

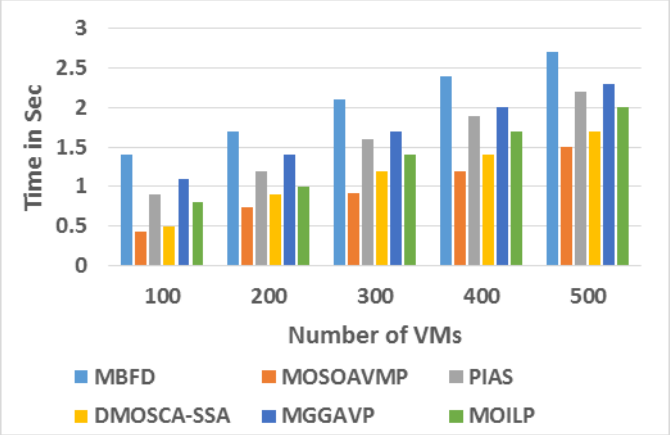

Figure 16 Average Storage Usage

5.2. Discussion and Limitation

The results show that the MOSOAVMP algorithm outperforms other leading algorithms in terms of energy consumption and resource management in cloud-based data centers. As shown in Figure 56, energy consumption increases proportionally with the number of active virtual machines (VMs). However, MOSOAVMP reduces energy consumption by 5.44% on average compared to DMOSCA-SSA, MOILP, PIAS, and MGGAVP, notably with a 6.18% improvement for 200 VMs.

In terms of VM migrations (Figure 57), the proposed algorithm demonstrates superior stability by minimizing the number of migrations required. By taking resource correlation into account, MOSOAVMP optimizes VM placement, reducing the number of active physical servers and associated costs. For example, for 2000 VMs, MOSOAVMP improves efficiency by 1.22% over DMOSCA-SSA and by 8.10% over MBFD.

Figures 59 to 62 illustrate that MOSOAVMP optimizes the use of PM resources such as CPU, memory, storage, and bandwidth, with respective improvements of 14.84%, 11.54%, 5.37%, and 6.88% compared to other algorithms. This efficient resource management is crucial to reducing waste and improving overall cloud computing performance.

When it comes to meeting SLAs, MOSOAVMP shows increased efficiency (Figure 64), reducing the fluctuations observed with other algorithms and guaranteeing a more reliable user experience. Finally, Figure 65 shows that MOSOAVMP converges faster to an optimal solution by increasing the number of iterations, demonstrating its robustness in solving complex VM placement problems.

The MOSOAVMP algorithm proves to be a superior solution for minimizing energy consumption, optimizing resources, and ensuring effective SLA management in cloud environments, systematically outperforming the other algorithms. Although the proposed method offers all these advantages, some improvements can be made by enhancing evaluation metrics such as workload handling, flexibility, adaptability, and security. These metrics, which have not been considered in this work, may impact QoS, highlighting a weakness of the proposed approach.

6. CONCLUSION

Cloud data centers consist of various power-intensive technologies, such as physical servers, switching devices, and cooling systems. However, the power consumption of these devices raises operational expenses and can lead to significant heat generation issues. In this paper, we propose an algorithm that can help optimize the VMP, intending to reduce energy consumption, enhance resource utilization, and minimize service level agreement violations. We formulate this problem as a multi-purpose problem optimization and employ a seagull-based approach for its solution. The test results show that the proposed solution achieves all the objectives set out in this work, compared with the algorithms of existing works. MOSOAVMP showed good exploitation and exploration performance. Regarding accuracy and speed of finding the best solution, MOSOAVMP has a better convergence speed.

In future research, we plan to compare this method with new metaheuristic algorithms such as [43], [44], [45], [46], [47], while considering additional metrics such as cloud security and task scheduling. Additionally, our objectives include exploring the placement of containers in the cloud using metaheuristics and deep learning. Furthermore, we will investigate the integration of game theory to enhance the optimization process.

REFERENCES

[1] M. Masdari and M. Zangakani, “Green Cloud Computing Using Proactive Virtual Machine Placement: Challenges and Issues,” J Grid Comput, 2019, doi: 10.1007/s10723-019-09489-9.

[2] J. Singh and N. K. Walia, “A Comprehensive Review of Cloud Computing Virtual Machine Consolidation,” IEEE Access, vol. 11. Institute of Electrical and Electronics Engineers Inc., pp. 106190–106209, 2023. doi: 10.1109/ACCESS.2023.3314613.

[3] A. Ndayikengurukiye, A. Ez-Zahout, A. Aboubakr, Y. Charkaoui, and O. Fouzia, “Resource Optimisation in Cloud Computing: Comparative Study of Algorithms Applied to Recommendations in a Big Data Analysis Architecture,” Journal of Automation, Mobile Robotics and Intelligent Systems, vol. 2021, no. 4, pp. 65–75, 2021, doi: 10.14313/JAMRIS/4-2021/28.

[4] S. Gharehpasha, M. Masdari, and A. Jafarian, “Power efficient virtual machine placement in cloud data centers with a discrete and chaotic

RESEARCH ARTICLE

hybrid optimization algorithm," Cluster Comput, vol. 24, no. 2, pp. 1293–1315, Jun. 2021, doi: 10.1007/s10586-020-03187-y.

[5] R. Regaieg, M. Koubàa, Z. Ales, and T. Aguili, "Multi-objective optimization for VM placement in homogeneous and heterogeneous cloud service provider data centers," Computing, vol. 103, no. 6, pp. 1255–1279, Jun. 2021, doi: 10.1007/s00607-021-00915-z.

[6] K. RahimiZadeh and A. Dehghani, "Design and evaluation of a joint profit and interference-aware VMs consolidation in IaaS cloud datacenter," Cluster Comput, vol. 24, no. 4, pp. 3249–3275, Dec. 2021, doi: 10.1007/s10586-021-03310-7.

[7] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing," Future Generation Computer Systems, vol. 28, no. 5, pp. 755–768, May 2012, doi: 10.1016/j.future.2011.04.017.

[8] Z. Xiao, J. Jiang, Y. Zhu, Z. Ming, S. Zhong, and S. Cai, "A solution of dynamic VMs placement problem for energy consumption optimization based on evolutionary game theory," Journal of Systems and Software, vol. 101, pp. 260–272, Mar. 2015, doi: 10.1016/j.jss.2014.12.030.

[9] J. Lu, W. Zhao, H. Zhu, J. Li, Z. Cheng, and G. Xiao, "Optimal machine placement based on improved genetic algorithm in cloud computing," Journal of Supercomputing, vol. 78, no. 3, pp. 3448–3476, Feb. 2022, doi: 10.1007/s11227-021-03953-8.

[10] L. Caviglione, M. Gaggero, M. Paolucci, and R. Ronco, "Deep reinforcement learning for multi-objective placement of virtual machines in cloud datacenters," Soft comput, vol. 25, no. 19, pp. 12569–12588, Oct. 2021, doi: 10.1007/s00500-020-05462-x.

[11] F. Alharbi, Y. C. Tian, M. Tang, M. H. Ferdaus, W. Z. Zhang, and Z. G. Yu, "Simultaneous application assignment and virtual machine placement via ant colony optimization for energy-efficient enterprise data centers," Cluster Comput, vol. 24, no. 2, pp. 1255–1275, Jun. 2021, doi: 10.1007/s10586-020-03186-z.

[12] M. Ghetas, "A multi-objective Monarch Butterfly Algorithm for virtual machine placement in cloud computing," Neural Comput Appl, vol. 33, no. 17, pp. 11011–11025, Sep. 2021, doi: 10.1007/s00521-020-05559-2.

[13] S. Y. Rashida, M. Sabaei, M. M. Ebadzadeh, and A. M. Rahmani, "A memetic grouping genetic algorithm for cost efficient VM placement in multi-cloud environment," Cluster Comput, vol. 23, no. 2, pp. 797–836, Jun. 2020, doi: 10.1007/s10586-019-02956-8.

[14] H. M. R. Al-Khafaji, "Improving Quality Indicators of the Cloud-Based IoT Networks Using an Improved Form of Seagull Optimization Algorithm," Future Internet, vol. 14, no. 10, Oct. 2022, doi: 10.3390/fi14100281.

[15] S. Nabavi, L. Wen, S. S. Gill, and M. Xu, "Seagull optimization algorithm based multi-objective VM placement in edge-cloud data centers," Internet of Things and Cyber-Physical Systems, vol. 3, pp. 28–36, 2023, doi: 10.1016/j.iotcps.2023.01.002.

[16] G. Dhiman, M. Garg, A. Nagar, V. Kumar, and M. Dehghani, "A novel algorithm for global optimization: Rat Swarm Optimizer," J Ambient Intell Humaniz Comput, vol. 12, no. 8, pp. 8457–8482, Aug. 2021, doi: 10.1007/s12652-020-02580-0.

[17] S. Mohapatra and P. Mohapatra, "American zebra optimization algorithm for global optimization problems," Sci Rep, vol. 13, no. 1, Dec. 2023, doi: 10.1038/s41598-023-31876-2.

[18] Z. Ma, G. Wu, P. N. Suganthan, A. Song, and Q. Luo, "Performance assessment and exhaustive listing of 500+ nature-inspired metaheuristic algorithms," Swarm Evol Comput, vol. 77, Mar. 2023, doi: 10.1016/j.swevo.2023.101248.

[19] R. Boopathi and E. S. Samundeeswari, "Amended Hybrid Scheduling for Cloud Computing with Real-Time Reliability Forecasting," International Journal of Computer Networks and Applications, vol. 10, no. 3, pp. 310–324, May 2023, doi: 10.22247/ijcna/2023/221887.

[20] P. Jain and S. K. Sharma, "A Load Balancing Aware Task Scheduling using Hybrid Firefly Salp Swarm Algorithm in Cloud Computing," International Journal of Computer Networks and Applications, vol. 10, no. 6, pp. 914–933, Nov. 2023, doi: 10.22247/ijcna/2023/223686.

[21] G. Dhiman and V. Kumar, "Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems," Knowl Based Syst, vol. 165, pp. 169–196, Feb. 2019, doi: 10.1016/j.knosys.2018.11.024.

[22] V. Kumar, Di. Kumar, M. Kaur, Di. Singh, S. A. Idris, and H. Alshazly, "A Novel Binary Seagull Optimizer and its Application to Feature Selection Problem," IEEE Access, vol. 9, pp. 103481–103496, 2021, doi: 10.1109/ACCESS.2021.3098642.

[23] G. Dhiman et al., "EMoSOA: a new evolutionary multi-objective seagull optimization algorithm for global optimization," International Journal of Machine Learning and Cybernetics, vol. 12, no. 2, pp. 571–596, Feb. 2021, doi: 10.1007/s13042-020-01189-1.

[24] F. M. Alamgir and M. S. Alam, "An artificial intelligence driven facial emotion recognition system using hybrid deep belief rain optimization," Multimed Tools Appl, vol. 82, no. 2, pp. 2437–2464, Jan. 2023, doi: 10.1007/s11042-022-13378-x.

[25] Q. Wang, M. Zhang, and S. Abdolhosseinzadeh, "Application of modified seagull optimization algorithm with archives in urban water distribution networks: Dealing with the consequences of sudden pollution load," Heliyon, vol. 10, no. 3, p. e24920, Feb. 2024, doi: 10.1016/j.heliyon.2024.e24920.

[26] P. Krishnadoss, V. K. Poornachary, P. Krishnamoorthy, and L. Shanmugam, "Improvised Seagull Optimization Algorithm for Scheduling Tasks in Heterogeneous Cloud Environment," Computers, Materials and Continua, vol. 74, no. 2, pp. 2461–2478, 2023, doi: 10.32604/cmc.2023.031614.

[27] X. Liu, G. Li, and P. Shao, "A Multi-Mechanism Seagull Optimization Algorithm Incorporating Generalized Opposition-Based Nonlinear Boundary Processing," Mathematics, vol. 10, no. 18, Sep. 2022, doi: 10.3390/math10183295.

[28] B. B. Al-Onazi, H. Alshamrani, F. O. Aldaajeh, A. S. A. Aziz, and M. Rizwanullah, "Modified Seagull Optimization with Deep Learning for Affect Classification in Arabic Tweets," IEEE Access, vol. 11, pp. 98958–98968, 2023, doi: 10.1109/ACCESS.2023.3310873.

[29] I. Abdullaev, N. Prodanova, K. A. Bhaskar, E. L. Lydia, S. Kadry, and J. Kim, "Task Offloading and Resource Allocation in IoT Based Mobile Edge Computing Using Deep Learning," Computers, Materials and Continua, vol. 76, no. 2, pp. 1463–1477, Aug. 2023, doi: 10.32604/cmc.2023.038417.

[30] J. Xue, X. Liu, H. Xu, and D. Zhang, "Research on the seagull optimization algorithm-based convolutional neural network rolling bearing fault diagnosis method," Engineering Research Express, vol. 5, no. 3, Sep. 2023, doi: 10.1088/2631-8695/acf09a.

[31] Q. Xia, Y. Ding, R. Zhang, H. Zhang, S. Li, and X. Li, "Optimal Performance and Application for Seagull Optimization Algorithm Using a Hybrid Strategy," Entropy, vol. 24, no. 7, Jul. 2022, doi: 10.3390/e24070973.

[32] E. S. Ghith and F. A. A. Tolba, "Tuning PID Controllers Based on Hybrid Arithmetic Optimization Algorithm and Artificial Gorilla Troop Optimization for Micro-Robotics Systems," IEEE Access, vol. 11, pp. 27138–27154, 2023, doi: 10.1109/ACCESS.2023.3258187.

[33] M. Hanif, N. Mohammad, K. Biswas, and B. Harun, "Seagull Optimization Algorithm for Solving Economic Load Dispatch Problem," in 3rd International Conference on Electrical, Computer and Communication Engineering, ECCE 2023, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/ECCE57851.2023.10101516.

[34] Nair, Dhanya G., KP Sanal Kumar, and S. Anu H. Nair. "Automated Social Distance Recognition and Classification using Seagull Optimization Algorithm with Multilayer Perceptron." Dandao Xuebao/Journal of Ballistics 35.3 (2023): 11-24.

[35] A. Mohammadzadeh, D. Javaheri, and J. Artin, "Chaotic hybrid multi-objective optimization algorithm for scientific workflow scheduling in multisite clouds," Journal of the Operational Research Society, 2023, doi: 10.1080/01605682.2023.2195426.

[36] S. Kaushaley, B. Shaw, and J. Ranjan Nayak, "Optimized Machine Learning based forecasting model for Solar Power Generation by using Crow Search Algorithm and Seagull Optimization Algorithm," 2022, doi: 10.21203/rs.3.rs-1987438/v1.

RESEARCH ARTICLE

[37] A. Ndayikengurukiye, A. Ez-Zahout, and F. Omary, "An overview of the different methods for optimizing the virtual resources placement in the Cloud Computing," J Phys Conf Ser, vol. 1743, p. 012030, Jan. 2021, doi: 10.1088/1742-6596/1743/1/012030.
[38] S. Omer, S. Azizi, M. Shojafar, and R. Tafazolli, "A priority, power and traffic-aware virtual machine placement of IoT applications in cloud data centers," Journal of Systems Architecture, vol. 115, May 2021, doi: 10.1016/j.sysarc.2021.101996.
[39] S. Azizi, M. Shojafar, J. Abawajy, and R. Buyya, "GRVMP: A Greedy Randomized Algorithm for Virtual Machine Placement in Cloud Data Centers," IEEE Syst J, vol. 15, no. 2, pp. 2571–2582, Jun. 2021, doi: 10.1109/JSYST.2020.3002721.
[40] S. Farzai, M. H. Shirvani, and M. Rabbani, "Multi-objective communication-aware optimization for virtual machine placement in cloud datacenters," Sustainable Computing: Informatics and Systems, vol. 28, Dec. 2020, doi: 10.1016/j.suscom.2020.100374.
[41] T. B. Hewage, S. Ilager, M. A. Rodriguez, and R. Buyya, "CloudSim express: A novel framework for rapid low code simulation of cloud computing environments," Softw Pract Exp, 2023, doi: 10.1002/spe.3290.
[42] I. Behera and S. Sobhanayak, "Task scheduling optimization in heterogeneous cloud computing environments: A hybrid GA-GWO approach," J Parallel Distrib Comput, vol. 183, Jan. 2024, doi: 10.1016/j.jpdc.2023.104766.
[43] Y. Yuan et al., "Coronavirus Mask Protection Algorithm: A New Bio-inspired Optimization Algorithm and Its Applications," J Bionic Eng, vol. 20, no. 4, pp. 1747–1765, Jul. 2023, doi: 10.1007/s42235-023-00359-5.
[44] F. A. Zeidabadi, M. Dehghani, P. Trojovský, Š. Hubálovský, V. Leiva, and G. Dhiman, "Archery Algorithm: A Novel Stochastic Optimization Algorithm for Solving Optimization Problems," Computers, Materials and Continua, vol. 72, no. 1, pp. 399–416, 2022, doi: 10.32604/cmc.2022.024736.
[45] W. Zilong and S. Peng, "A Multi-Strategy Dung Beetle Optimization Algorithm for Optimizing Constrained Engineering Problems," IEEE Access, vol. 11, pp. 98805–98817, 2023, doi: 10.1109/ACCESS.2023.3313930.
[46] Y. Du, H. Yuan, K. Jia, and F. Li, "Research on Threshold Segmentation Method of Two-Dimensional Otsu Image Based on Improved Sparrow Search Algorithm," IEEE Access, vol. 11, pp. 70459–70469, 2023, doi: 10.1109/ACCESS.2023.3293191.
[47] Aristide Ndayikengurukiye, Abderrahmane Ez-Zahout, Fouzia Omary, "Optimizing Virtual Machines Placement in a Heterogeneous Cloud Data Center System", International Journal of Computer Networks and Applications (IJCNA), 11(1), PP: 1-12, 2024, DOI: 10.22247/ijcna/2024/224431.

Authors

Aristide Ndayikengurukiye graduated with an Msc in Computer Engineering from the University of Burundi in 2017. PhD Student in Intelligent Processing Systems & Security Team (IPSS), Computer Science Department Faculty of Science of Rabat, Mohammed V University, Morocco. His research interests include Cloud computing optimization, artificial intelligence, and Big Data.

Rim Doukha is currently a Ph.D. student in computer science at the Department of Computer Science, Faculty of Science, Mohammed V University. Her research focuses on the fields of cloud computing and deep learning. She holds a Master's degree in Internet of Things and Mobile Services from ENSIAS. Notably, she has completed several internships in Morocco and France, worked in Belgium as a Cloud Computing research engineer as part of a European project, and is currently a project officer in cloud computing and technical standardization in Luxembourg.

Eric Niyukuri is currently a software developer at Orion Systems & Design. He has been working in the FinTech industry, especially on electronic payment systems such as EFT, RTGS and Cheque Truncation systems, Mobile Banking, and E-Tax systems. Before joining the FinTech industry he developed software for insurance. He holds a Master of Science degree in Computer Engineering from the University of Burundi.

Eric Muheto As a seasoned Unix SME at Morgan Stanley, I excel in DevOps and middleware, holding a computer science degree and Lib@Web certification. Proficient in Unix and cloud management with tools like Aquilon and Ansible, I specialize in automation using Python scripts. Notable projects include enhancing safety at Canadian National Railways and containerizing legacy apps with Docker. Key contributions at the National Bank of Canada.

Abderrahmane Ez-Zahout is a Professeur Habilité of computer science at the Department of Computer Science/faculty of Sciences at Mohammed V University. His research is in the fields of computer sciences, digital systems, big data, and computer vision. Recently, he has worked on intelligent systems. He has served as an invited reviewer for many journals. Besides, he is also involved in NGOs, and student associations, and managing non-profit foundations.

Fouzia Omary is the leader of IPSS team research and is currently a full professor of computer science at the Department of Computer Science/ENS, Mohammed V University. She is a full-time researcher at Mohammed V University and she is the chair of the international conference on cybersecurity and blockchain and also the leader of the national network of blockchain and cryptocurrency.

How to cite this article:

Aristide Ndayikengurukiye, Rim Doukha, Eric Niyukuri, Eric Muheto, Abderrahmane Ez-zahout, Fouzia Omary, "SOAVMP: Multi-Objective Virtual Machine Placement in Cloud Computing Based on the Seagull Optimization Algorithm", International Journal of Computer Networks and Applications (IJCNA), 11(3), PP: 375-389, 2024, DOI: 10.22247/ijcna/2024/24.

Résume

La gestion des ressources virtuelles dans le cloud computing est cruciale pour garantir une utilisation efficace des ressources. La modélisation consiste à représenter les composants sous forme de modèles mathématiques pour analyser le comportement du système et prévoir l'utilisation des ressources. Ces modèles permettent également d'évaluer les compromis entre coût et performance. L'optimisation des ressources vise à allouer les ressources en minimisant les coûts tout en répondant aux besoins des applications. Les approches incluent des algorithmes heuristiques, rapides mais parfois imprécis, et des méthodes d'optimisation mathématique, plus précises mais complexes. Des algorithmes comme Grey Wolf Optimization (GWO) et Seagull Optimization (MOSOAVMP) ont été développés pour améliorer le placement des machines virtuelles (VMP). GWO permet de réduire la consommation d'énergie de 20,99% et les gaspillages de 1,80%. MOSOAVMP, en optimisant l'utilisation des ressources et en minimisant les violations des accords de niveau de service, améliore l'efficacité des centres de données, réduisant les gaspillages et augmentant l'utilisation du CPU, de la mémoire et de la bande passante. Ces approches garantissent une meilleure gestion des ressources dans le cloud, tout en réduisant les coûts. En parallèle, les systèmes de recommandation, utilisant des algorithmes comme K-NN, contribuent à divers secteurs, y compris la gestion électorale, en proposant des suggestions pertinentes en exploitant les données massives et l'apprentissage automatique.

Mots-clefs (5) : Cloud Computing, Optimisation, Système de Recommandation, Métaheuristiques et Placement des Machines Virtuelles

Abstract

Virtual resource management in cloud computing is crucial to ensure efficient resource utilization. Modeling involves representing components as mathematical models to analyze system behavior and predict resource utilization. These models can also be used to assess trade-offs between cost and performance. Resource optimization aims to allocate resources while minimizing costs and meeting application needs. Approaches include heuristic algorithms, which are fast but sometimes imprecise, and mathematical optimization methods, which are more precise but complex. Algorithms such as Grey Wolf Optimization (GWO) and Seagull Optimization (MOSOAVMP) have been developed to improve the placement of virtual machines (VMPs). GWO reduces energy consumption by 20.99% and wastage by 1.80%. MOSOAVMP, by optimizing resource utilization and minimizing SLA violations, improves data center efficiency, reduces waste, and increases CPU, memory, and bandwidth utilization. These approaches guarantee better resource management in the cloud while reducing costs. At the same time, recommendation systems, using algorithms like K-NN, contribute to various sectors, including election management, by offering relevant suggestions by exploiting massive data and machine learning.

Keywords (5) : Cloud Computing, Optimization, Recommender System, Metaheuristics and Virtual Machine Placement