RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de Mohamed El-Bachir El-Ibrahimi - BBA

Faculté des Mathématiques et Informatiques



MÉMOIRE

Présenté en vue de l'obtention du diplôme

Master en Informatique

Spécialité : Réseaux & Multimédias

Thème

Optimisation de Routage dans les Réseaux VANET par Apprentissage Supervisé

Présenté par :

- MEHSAS Hanya Melak
- MEKKI Douaa

Soutenu le : 12/06/2025, devant la commission d'examen suivante :

Dr. MOUSSAOUI Ali	MCB - Université de BBA	Président
Dr. BENMESSAHEL Ilyes	MCB - Université de BBA	Examinateur
Dr. SENOUCI Oussama	MCB - Université de BBA	Encadrant

Promotion 2024 / 2025

“Je n’ai jamais rêvé de succès,
J’ai travaillé pour ça.”

- Estee Lauder -

Résumé

Ce mémoire traite de l'optimisation du routage dans les réseaux ad hoc véhiculaires (VANET), composante clé des systèmes de transport intelligents (ITS). Ces réseaux permettent aux véhicules d'échanger des données en temps réel (V2V et V2I) afin d'améliorer la sécurité routière et la gestion du trafic. Toutefois, leur nature dynamique, leur topologie instable et la mobilité élevée des nœuds compliquent le routage, nuisant à la qualité de service (QoS). Pour répondre à ces défis, nous proposons d'intégrer l'intelligence artificielle (IA), en particulier l'apprentissage automatique supervisé, dans le protocole HRLB-SVDN. Le modèle utilisé, CatBoostClassifier, est capable d'analyser des données hétérogènes avec grande précision. Entraîné sur des données simulées réalistes, il permet de prédire les meilleurs chemins de communication selon des paramètres tels que la densité de véhicules, la distance ou la stabilité des liens. Les résultats expérimentaux sont très prometteurs : une accuracy de 98,75%, un score ROC-AUC de 99,79%, une précision moyenne de 99,71% et une perte logarithmique de 0,038. Ces performances indiquent une excellente capacité à identifier les chemins les plus fiables dans des environnements réseau très variables. Les simulations ont mis en évidence une amélioration notable des performances du protocole enrichi, avec une augmentation de 10,75% du taux de livraison moyen, une hausse de 7,86% du débit, ainsi qu'une réduction de 10,08% de la latence moyenne. En conclusion, ce travail montre l'efficacité des approches IA pour concevoir des protocoles de routage intelligents, robustes et adaptatifs, adaptés aux exigences élevées des réseaux VANET modernes.

Mots clés : VANET, Routage, QoS, Intelligence Artificielle, Apprentissage Supervisé, CatBoostClassifier, ITS.

Abstract

This thesis focuses on optimizing routing in Vehicular Ad Hoc Networks (VANET), a key component of Intelligent Transportation Systems (ITS). These dynamic networks enable vehicles to exchange real-time data (V2V and V2I), thereby enhancing road safety and improving traffic management. However, their dynamic nature, unstable topology, and high node mobility complicate data routing, adversely affecting the quality of service (QoS). To address these challenges, we propose the integration of Artificial Intelligence (AI), specifically supervised machine learning, into the HRLB-SVDN protocol. The chosen model, *CatBoostClassifier*, can effectively analyze heterogeneous data with high precision. Trained on realistic simulated datasets, it predicts optimal communication paths based on parameters such as vehicle density, inter-node distance, and link stability. The experimental results are highly promising : an accuracy of 98.75%, a ROC-AUC score of 99.79%, an average precision of 99.71%, and a logarithmic loss of just 0.038. These metrics demonstrate a strong ability to identify reliable routing paths in dynamic network environments. Simulations also revealed a significant improvement in protocol performance, with a 10.75% increase in average delivery rate, a 7.86% increase in throughput, and a 10.08% reduction in average latency. In conclusion, this work highlights the effectiveness of AI-based approaches in designing intelligent, robust, and adaptive routing protocols tailored to the demanding requirements of modern VANET environments.

Keywords : VANET, Routing, QoS, Artificial Intelligence, Supervised Learning, CatBoostClassifier, ITS.

ملخص

يركز هذا البحث على تحسين عملية التوجيه في الشبكات اللاسلكية بين المركبات (VANET)، والتي تُعد من المكونات الأساسية في أنظمة النقل الذكية (ITS). تتيح هذه الشبكات للمركبات تبادل البيانات في الوقت الحقيقي، سواء بين المركبات (V2V) أو بين المركبة والبنية التحتية (V2I)، وذلك بهدف تعزيز السلامة على الطرق وتحسين إدارة حركة المرور. ومع ذلك، فإن الطبيعة الديناميكية لهذه الشبكات، إلى جانب تغير الطوبولوجيا وارتفاع معدل تنقل العقد، يجعل من التوجيه الفعّال تحدياًا يؤثر سلباًا على جودة الخدمة (QoS). ولمواجهة هذه التحديات، نقترح دمج تقنيات الذكاء الاصطناعي، وتحديداً التعلم الآلي المُشرف، في بروتوكول التوجيه HRLB-SVDN. وقد تم استخدام نموذج CatBoostClassifier، المعروف بقدرته العالية على معالجة البيانات غير المتجانسة بدقة كبيرة. وتم تدريب هذا النموذج باستخدام بيانات محاكاة واقعية، مما يتيح له التنبؤ بالمسارات المثلى بناءً على معايير سياقية مثل كثافة المركبات، والمسافة، واستقرار الروابط. أظهرت النتائج فعالية كبيرة، حيث حقق النموذج دقة بنسبة 98.75%، ودرجة ROC-AUC بلغت 99.79%، ومتوسط دقة 99.71%، مع خسارة لوجاريتمية منخفضة قدرها 0.038. كما أظهرت المحاكاة تحسّناًا واضحاًا في أداء البروتوكول، بزيادة بنسبة 10.75% في معدل التسليم، وتحسّن بنسبة 7.86% في معدل الإرسال، وانخفاض بنسبة 10.08% في زمن التأخير. تُبرز هذه النتائج فعالية التعلم الآلي في تصميم بروتوكولات توجيه ذكية، قوية وقابلة للتكيف مع طبيعة الشبكات المتغيرة في بيئات VANET الحديثة.

الكلمات المفتاحية: VANET، التوجيه، جودة الخدمة (QoS)، الذكاء الاصطناعي، التعلم الآلي المُشرف، CatBoostClassifier، أنظمة النقل الذكية (ITS).

Remerciements

Louange à **Allah**, le Tout-Puissant, pour nous avoir accordé la force, la patience et la persévérance nécessaires à la réalisation de ce travail.

Nous exprimons notre **profonde gratitude** et nos **remerciements les plus sincères** à notre encadrant, **Dr. Senouci Oussama**, pour son accompagnement précieux, ses conseils éclairés, sa disponibilité constante, ainsi que pour son soutien moral et ses encouragements qui nous ont guidés tout au long de cette étude. Son expertise et ses orientations pertinentes ont grandement contribué à l'élaboration et à la qualité de ce mémoire.

Nous remercions également les **membres du jury** pour l'attention qu'ils ont portée à notre travail et pour avoir accepté de l'évaluer.

Nos remerciements vont à **l'ensemble des enseignants** qui nous ont transmis leur savoir avec passion et rigueur, ainsi qu'au **personnel administratif et technique** pour leur assistance, leur disponibilité et leur bienveillance tout au long de notre parcours.

Nous n'oublions pas toutes les personnes qui, de près ou de loin, ont apporté leur aide, leur soutien ou leurs conseils durant nos stages de fin d'études et la réalisation de ce mémoire. Qu'elles trouvent ici l'expression de notre reconnaissance sincère.

Enfin, nous adressons nos **remerciements les plus chaleureux** à l'**Université Mohamed El Bachir El Ibrahimi**, et tout particulièrement à son **corps professoral**, pour les efforts remarquables déployés afin d'assurer un cadre d'étude favorable à l'épanouissement des étudiants.

Merci à toutes et à tous.

Dédicace

« Heureux de ce qu'Allah leur a accordé de par Sa grâce »

Louange à Dieu, par Sa grâce les œuvres se réalisent, et grâce à Son aide les objectifs se concrétisent. Je Le remercie de m'avoir accordé la force, la patience et la persévérance pour accomplir ce travail.Je dédie ce succès aux êtres les plus chers à mon cœur, à ceux qui ont été, après Dieu, la source de ma force tout au long de ce parcours.

À mon cher père, Ahmed Lakhdar

À la lumière qui a illuminé mon chemin, à l'homme dont je porte fièrement le nom, à mon premier mentor, celui qui a œuvré toute sa vie pour que nous soyons les meilleurs. Porter ton nom a été mon tout premier honneur. Tu as, dans mon cœur, toute la considération et le respect.

À ma mère bien-aimée

À celle par qui j'ai vu le chemin de ma vie, à celle qui m'a donné la confiance et la fierté de moi-même, au cœur aimant, à mon pilier dans cette vie et source de sécurité. Tes prières m'ont toujours entourée, tu as toujours été mon refuge et ma main droite à chaque étape.

À mes frères Anis et Mohamed El-Amine

À mes deux frères, compagnons de cœur et de vie, qui ont toujours su m'offrir leur soutien, leur protection et leur présence réconfortante. Merci pour vos encouragements silencieux, vos regards pleins de fierté et votre amour discret mais si précieux tout au long de ce chemin.

À ma sœur unique, Basmala

À celle qui partage avec moi bien plus que des liens de sang, ma confidente, ma force douce, ma plus fidèle alliée. Ta présence m'a été indispensable, ton sourire un apaisement, et ta joie, un moteur de plus dans mon parcours. Ce succès est aussi le tien.

À ma chère binôme Mehsas Hanya Malek

À celle avec qui j'ai partagé avec moi chaque détail de cette aventure scientifique, les efforts, les doutes et les réussites. Merci pour ta rigueur, ton amitié, ton esprit d'équipe et ton soutien constant,grâce auxquels cette étape a pu être franchie avec succès.

À toute ma famille et à ceux qui m'ont offert leur amour, leur soutien, et dont la présence a marqué mon chemin.

À mes fidèles amis, compagnons de route qui ont rendu ce voyage plus agréable et plus léger, et qui ont été un soutien inoubliable.

À tous ceux qui ont cru en moi, et m'ont soutenue d'une manière ou d'une autre... cette réussite est autant la vôtre que la mienne.

⋆ Douaa ⋆

Dédicace

« Seigneur, accrois mes connaissances »

Le parcours fut difficile, mais accompli par la grâce de Dieu. Louange à Lui qui éclaire les débuts, soutient dans les épreuves et guide jusqu'à la réussite, car toute réussite vient de Lui seul.

À ma mère

À la complice de ma vie, l'amour de mon cœur, la plus grande bénédiction que Dieu m'ait accordée. À celle qui a enveloppé mon nom de prières nuit et jour, qui a éclairé mon chemin par son amour et sa tendresse. À celle dont les invocations ont adouci les épreuves, et dont les rêves ont toujours inclus ce jour... Ce diplôme est d'abord pour toi.

À mon père

À celui qui m'a appris que la vie est un combat, et que la science en est l'arme. À celui dont je porte le nom avec fierté. Merci pour ta force, tes principes, ton soutien discret et constant.

À mon frère Bassem et son épouse

À Bassem, mon pilier, mon repère, toujours responsable et présent. Tu as été le soutien solide dans les moments difficiles. À ton épouse au cœur pur et aux prières sincères, merci à vous deux pour votre présence et votre bienveillance.

À mon frère Khalil

À l'être le plus cher à mon cœur, à celui qui a marché à mes côtés jusqu'au bout. Celui qui se rassurait de mon bien-être, qui m'a entourée de ses invocations. Tu as été mon refuge, mon frère, mon ami, mon soutien inébranlable. Tu m'as donné de la force quand j'en manquais, tu as cru en moi. Ce diplôme t'est dédié.

À celle qui est ma sœur de sang et d'âme

À ma deuxième mère, ma confidente, mon alliée, mon soutien fidèle dans chaque étape. Ta présence est un trésor, et ta joie est la mienne. Ce succès est le nôtre.

À mon binôme Douaa Mekki

À celle avec qui j'ai partagé les efforts, les doutes, les défis et les réussites. Merci pour ton sérieux, ton soutien, ta patience et ton amitié. Ce parcours n'aurait pas été le même sans toi.

Enfin...

Ce succès n'est pas uniquement le mien, mais celui de vous tous qui m'avez soutenue, encouragée, portée. Merci du fond du cœur. J'espère être votre fierté, comme vous avez toujours été ma force.

⋆ ***Hanya*** ⋆

Table des matières

Liste des figures **xiv**

Liste des tables **xv**

Abréviations **xvi**

Introduction Générale **1**

1 Généralités sur les Réseaux VANET **5**

1.1 Introduction 5

1.2 Introduction aux Réseaux VANET 5

1.2.1 Définition des Réseaux VANET 6

1.2.2 Architecture des Réseaux VANET 6

1.2.2.1 Composants Principaux 6

1.2.2.2 Topologie des Réseaux VANET 8

1.2.3 Applications des Réseaux VANET 9

1.2.3.1 Sécurité Routière 9

1.2.3.2 Gestion du Trafic 9

1.2.3.3 Divertissement et Services 10

1.3 Problématiques du Routage dans les Réseaux VANET 10

1.3.1 Protocole de Routage dans les Réseaux VANET 10

1.3.1.1 Classification des Protocoles de Routage 11

1.3.1.2 Problèmes d'Optimisation 12

1.4 Qualité de Service (QoS) dans les Réseaux VANET 13

1.4.1 Paramètres de la QoS 13

1.4.1.1 Débit 13

1.4.1.2 Latence 13
1.4.1.3 Variabilité du Délai (Jitter) et Pertes de Paquets . . . 14
1.5 Conclusion 14

2 État de l'art **15**
2.1 Introduction 15
2.2 Méthodologie de Recherche 15
2.2.1 Objectifs de l'Étude de la Littérature 16
2.2.2 Méthodes et Outils Utilisés 16
2.2.3 Critères d'Évaluation 17
2.2.3.1 Robustesse 17
2.2.3.2 Flexibilité 17
2.2.3.3 Scalabilité 17
2.2.3.4 Fiabilité des communications 17
2.2.3.5 Latence 18
2.2.3.6 Sécurité et confidentialité 18
2.3 Contexte et Défis des Réseaux VANETs 18
2.3.1 Défis du Routage dans les Réseaux VANET 18
2.3.1.1 Mobilité Élevée des Véhicules 18
2.3.1.2 Hétérogénéité des Réseaux 19
2.3.1.3 Connexions Intermittentes 19
2.3.2 Défis de la Qualité de Service dans les Réseaux VANETs 19
2.3.2.1 Variabilité des Conditions Réseau 19
2.3.2.2 Sécurisation des Communications 20
2.4 Classification des Protocoles de Routage et QoS dans les Réseaux VANETs 20
2.4.1 Protocoles de Routage dans les Réseaux VANET 20
2.4.1.1 Protocoles Basés sur la Position 21
2.4.1.2 Protocoles Basés sur la Topologie 24
2.4.1.3 Protocoles Basés sur le Clustering 26
2.4.1.4 Limites des Protocoles de Routage 28
2.4.2 Protocoles Basés sur la QoS dans les Réseaux VANET 28
2.4.2.1 Protocoles Basés sur la Latence 29
2.4.2.2 Protocoles Basés sur la Fiabilité 31

2.4.2.3 Protocoles Basés sur la Bande Passante 32
2.4.2.4 Limites des Protocoles Basés sur la QoS 34
2.5 Évolution des Techniques de Routage et de QoS dans les Réseaux VANET 35
2.5.1 Approches Conventionnelles vs Approches Modernes 35
2.5.1.1 Approches Conventionnelles 35
2.5.1.2 Approches Modernes 36
2.5.2 Impact des Technologies Émergentes sur le Routage et la QoS . 36
2.6 Intelligence Artificielle pour l'Optimisation dans les Réseaux VANET . 38
2.6.1 Introduction aux Techniques d'IA Utilisées 38
2.6.2 Techniques d'Optimisation pour le Routage dans les VANET . . 39
2.6.2.1 Apprentissage Supervisé 39
2.6.2.2 Apprentissage Non Supervisé 40
2.6.2.3 Algorithmes Génétique 40
2.6.2.4 Apprentissage Profond (Réseaux de Neurones) 41
2.6.2.5 Apprentissage par Renforcement 41
2.6.3 Avantages de l'IA pour l'Amélioration du Routage et de la QoS dans les Réseaux VANET 42
2.7 Conclusion 43

3 Protocole de Routage HRLB-SVDN 44
3.1 Introduction 44
3.2 Concepts fondamentaux 45
3.3 Modèle de réseau et hypothèses du protocole 47
3.3.1 Modèle de Réseau 47
3.3.2 Hypothèses 50
3.4 Conception du Protocole 51
3.4.1 Vue d'ensemble de la conception 51
3.5 Exemple de déploiement du Protocole HRLB-SVDN 56
3.6 Limites du Protocole HRLB-SVDN 59
3.7 Conclusion 60

4 Optimisation du routage HRLB-SVDN via un modèle d'apprentissage supervisé 61

4.1 Introduction 61
4.2 Motivations et objectifs de l'intégration du modèle CatBoostClassifier . 62
4.2.1 Motivations 62
4.2.2 Objectifs visés 63
4.3 Présentation détaillée du modèle CatBoostClassifier 64
4.3.1 Principe général du Gradient Boosting 65
4.3.2 Fonction de perte utilisée 66
4.3.3 Spécificités de CatBoost : Ordered Boosting 66
4.3.4 Traitement avancé des variables catégorielles 66
4.3.5 Structure interne du modèle 67
4.3.6 Régularisation et contrôle du surapprentissage 67
4.3.7 Adaptation au contexte des réseaux VANET 68
4.4 Description du jeu de données et processus de développement du modèle 69
4.4.1 Caractéristiques générales du dataset 69
4.4.2 Structure détaillée des enregistrements 69
4.4.3 Étapes théoriques de développement du modèle 70
4.4.3.1 Préparation et nettoyage des données 70
4.4.3.2 Sélection des caractéristiques pertinentes 70
4.4.3.3 Paramétrage du modèle `CatBoostClassifier` 71
4.4.3.4 Entraînement et validation croisée 72
4.4.3.5 Évaluation et test du modèle 73
4.5 Métriques d'évaluation du modèle 76
4.5.1 Accuracy (Précision Globale) 76
4.5.2 Log Loss (Erreur logarithmique) 77
4.5.3 ROC-AUC (Surface sous la courbe ROC) 77
4.5.4 Average Precision Score (Score de précision moyenne) 78
4.6 Intégration et Déploiement du Modèle au Protocole HRLB-SVDN . . . 78
4.6.1 Contexte de Simulation 78
4.6.2 Paramètres de Simulation 79
4.6.3 Initialisation de la Grille et des Véhicules 79
4.6.4 Intégration du Modèle CatBoost 81
4.6.5 Résultats de la Prédiction et de la Simulation 81

4.6.6 Analyse Comparative 82
4.6.6.1 Taux de Livraison 82
4.6.6.2 Débit 83
4.6.6.3 Latence Moyenne 84
4.6.7 Analyse Localisée des Performances par Véhicule 85
4.7 Conclusion 87

Conclusion Générale **89**

Bibliographie **91**

Annexe **118**

Table des figures

1.1 Architecture de Communication VANET. 7
1.2 Dispositifs de Communication Véhiculaire : OBU et RSU. 8

2.1 Classification des protocoles de routage et QoS dans les réseaux VANET. 21
2.2 Architecture d'un réseau VANET basé sur l'IA. 39

3.1 Architecture d'un SDN-VANET. 48
3.2 Division du réseau en plusieurs grilles. 49
3.3 Organigramme de la sélection des grilles dans HRLB-SVDN. 52
3.4 Organigramme de la sélection des chemins avec équilibrage de charge. . 53
3.5 Exemple de graphe de chemin utilisé pour le routage. 54
3.6 Organigramme de la sélection des nœuds relais. 55
3.7 Processus de sélection des grilles, chemins et nœuds relais. 56
3.8 Description des nœuds sans voisins dans HRLB. 57

4.1 Diagramme du processus d'apprentissage du modèle CatBoost. 64
4.2 Distribution des classes dans la variable `chemin` (0 = Secondaire, 1 = Principal). 71
4.3 Résultat de l'entraînement du modèle CatBoost. 73
4.4 Matrice de confusion. 74
4.5 Courbe ROC. 75
4.6 Courbe Précision-Rappel. 76
4.7 Distribution des prédictions. 77
4.8 Topologie en grille (3x3) représentant les zones urbaines. 80
4.9 Résumé des performances simulée. 82
4.10 Extraits de détails des performances des véhicules. 83

4.11 Taux de Livraison vs. Nombre de Paquets. 84
4.12 Débit vs. Nombre de Paquets. 85
4.13 Latence Moyenne vs. Nombre de Paquets. 86
4.14 Détails pour des Véhicules Sélectionnés aléatoirement. 87

Liste des tableaux

2.1 Comparaison des algorithmes basés sur la position. 23
2.2 Comparaison des algorithmes basés sur la topologie. 26
2.3 Comparaison des algorithmes basés sur le Clustering. 28
2.4 Comparaison des algorithmes basés sur la latence. 31
2.5 Comparaison des algorithmes basés sur la Fiabilité. 32
2.6 Comparaison des algorithmes basés sur la bande passante. 34

3.1 Données initiales et hypothèses du scénario simulé. 59

4.1 Rapport de classification détaillé 73
4.2 Paramètres de simulation 79
4.3 Densité des véhicules dans chaque grille. 80

Abréviations

— **AI / IA** : Artificial Intelligence / Intelligence Artificielle

— **CatBoost** : Categorical Boosting

— **HRLB** : Hierarchical Routing with Load Balancing

— **HRLB-SVDN** : HRLB for Software Defined Vehicular Networks

— **IoV** : Internet of Vehicles

— **ITS** : Intelligent Transport Systems

— **ML** : Machine Learning

— **OBU** : On-Board Unit

— **PDR** : Packet Delivery Ratio

— **QoS** : Quality of Service

— **RNN** : Recurrent Neural Network

— **ROC-AUC** : Receiver Operating Characteristic - Area Under Curve

— **RSU** : Road Side Unit

— **SDN** : Software Defined Networking

— **SDVN** : Software Defined Vehicular Network

— **V2I** : Vehicle-to-Infrastructure

— **V2V** : Vehicle-to-Vehicle

— **V2X** : Vehicle-to-Everything

— **VANET** : Vehicular Ad hoc Network

Introduction Générale

Les avancées technologiques dans le domaine des transports intelligents ont conduit à l'émergence des réseaux ad hoc véhiculaires (VANET), une sous-catégorie des réseaux mobiles qui permet aux véhicules de communiquer entre eux et avec les infrastructures environnantes. Ces réseaux constituent une pierre angulaire des systèmes de transport intelligents (ITS), dont l'objectif est d'assurer une circulation plus sûre, plus fluide et plus efficace. Grâce aux VANET, les véhicules peuvent échanger des informations critiques en temps réel, telles que des alertes de collision, des conditions de circulation, ou des notifications d'urgence.

Cependant, malgré leur potentiel immense, les réseaux VANET sont confrontés à des défis techniques importants, en particulier dans la gestion du routage de données et dans le maintien d'une qualité de service (QoS) satisfaisante. En effet, la mobilité élevée des véhicules, l'instabilité topologique, la variabilité des connexions, ainsi que la densité hétérogène du trafic sont autant de facteurs qui compliquent la mise en place de protocoles de communication efficaces et fiables. Ces caractéristiques particulières des VANET rendent inopérantes de nombreuses méthodes de routage traditionnelles issues des réseaux mobiles classiques.

Dans ce contexte, l'intelligence artificielle (IA), et plus précisément les techniques d'apprentissage automatique supervisé, apparaissent comme une solution prometteuse pour faire face à ces enjeux. L'intégration de l'IA dans les protocoles de routage permet une prise de décision dynamique, plus adaptée aux conditions changeantes du réseau, en exploitant des modèles prédictifs capables d'anticiper le meilleur chemin de transmission. C'est dans cette optique que s'inscrit ce mémoire, qui propose une approche innovante d'optimisation du routage basée sur un modèle de classification supervisée.

Problématique & Motivations

La complexité du routage dans les réseaux VANET provient de plusieurs facteurs interdépendants : la mobilité rapide des nœuds, la connectivité intermittente, la multiplicité des sources d'interférences, ainsi que l'imprédictibilité des trajets. Ces éléments rendent difficile le maintien de chemins de communication stables et efficaces. Les protocoles classiques, tels que ceux basés sur la topologie ou la géolocalisation, peinent à s'adapter en temps réel aux changements fréquents de l'environnement réseau. De plus, la majorité de ces protocoles ne tiennent pas compte de paramètres contextuels tels que la densité de trafic, la latence, la charge réseau ou encore l'état tampon des véhicules relais.

Par ailleurs, les besoins croissants en matière de sécurité routière, d'optimisation du trafic, et d'applications multimédia embarquées exigent des niveaux de QoS élevés, ce que les approches actuelles ne garantissent pas toujours. Ces limitations ouvrent la voie à l'exploration de nouvelles méthodes de routage plus intelligentes, capables de s'adapter dynamiquement à l'environnement réseau, tout en maximisant la performance.

L'évolution rapide des algorithmes d'apprentissage automatique, tels que les classifieurs basés sur le Gradient Boosting, offre aujourd'hui des outils performants pour modéliser et prédire le comportement optimal d'un réseau complexe. Parmi ces modèles, CatBoostClassifier, reconnu pour son efficacité à traiter des variables catégorielles et sa robustesse face au surapprentissage, se présente comme un candidat pertinent pour optimiser les décisions de routage dans un protocole existant.

Objectifs & Contributions

L'objectif principal de ce mémoire est de concevoir une méthode d'optimisation du routage dans les réseaux VANET à l'aide de l'apprentissage automatique supervisé, afin de surmonter les limites des protocoles classiques en environnement à forte mobilité. Plus précisément, le travail porte sur l'intégration du modèle CatBoostClassifier dans le protocole HRLB-SVDN pour améliorer dynamiquement le choix des chemins de transmission. Ce modèle permet, à partir de caractéristiques contextuelles telles que la densité de véhicules, la charge des tampons ou la distance entre les nœuds, de prédire

les chemins les plus efficaces en fonction de la situation réseau. L'approche proposée vise à rendre le routage plus intelligent et plus réactif, en tirant parti des capacités prédictives du classifieur pour optimiser les performances globales du réseau.

Les contributions de ce mémoire sont multiples. Tout d'abord, il s'agit d'une analyse approfondie des défis liés au routage et à la qualité de service dans les VANET, mettant en lumière les contraintes spécifiques de ce type de réseau. Ensuite, une architecture de routage optimisée est proposée en combinant les avantages du protocole HRLB-SVDN avec les capacités d'un modèle supervisé de machine learning. Cette intégration est mise en œuvre dans un environnement de simulation permettant de mesurer l'impact réel de la solution sur des métriques essentielles telles que le taux de livraison de paquets, la latence moyenne et le débit. Enfin, les résultats obtenus permettent de démontrer la pertinence de l'intelligence artificielle dans l'optimisation des protocoles de routage pour les réseaux véhiculaires de nouvelle génération.

Organisation du mémoire

Le mémoire est organisé en quatre chapitres distincts, chacun traitant d'un aspect spécifique de notre travail :

— **Chapitre 1 : Généralités sur les Réseaux VANET**

Ce chapitre introduit les concepts fondamentaux liés aux réseaux VANET, leurs architectures, les problématiques de routage et de qualité de service, ainsi que les enjeux rencontrés dans ces environnements dynamiques.

— **Chapitre 2 : État d'art**

Il présente une revue critique des travaux de recherche existants, en distinguant les approches classiques et celles basées sur l'intelligence artificielle, tout en soulignant les lacunes des protocoles actuels.

— **Chapitre 3 : Protocole de routage HRLB-SVDN**

Ce chapitre présente en détail le protocole de routage HRLB-SVDN, choisi comme étude de cas. Il en décrit les mécanismes hiérarchiques, tout en mettant en évidence ses limites, lesquelles ont motivé la proposition d'une amélioration.

— **Chapitre 4 : Optimisation du routage HRLB-SVDN via un modèle d'apprentissage supervisé**

Ce chapitre présente le développement et l'intégration du modèle CatBoostClassifier au sein du protocole HRLB-SVDN. Il détaille le processus de développement du modèle, l'évaluation de ses performances, son intégration dans le protocole, ainsi que les résultats des simulations et du déploiement réel, avant d'en analyser les performances.

— **Conclusion Générale :**

Ce dernier volet du mémoire synthétise les principaux résultats obtenus, évalue les apports du travail réalisé et en tire des enseignements généraux. Il propose également des perspectives de recherche pour approfondir et élargir les solutions proposées, notamment en intégrant d'autres techniques d'intelligence artificielle ou en explorant des scénarios de déploiement réels plus complexes.

Chapitre 1

Généralités sur les Réseaux VANET

1.1 Introduction

Les réseaux véhiculaires ad hoc (VANET) jouent un rôle fondamental dans le développement des systèmes de transport intelligents (ITS). Ils permettent aux véhicules de communiquer entre eux (V2V) et avec les infrastructures routières (V2I), facilitant ainsi la transmission d'informations critiques en temps réel, telles que les alertes de sécurité, les conditions de circulation ou les données de navigation. Cette capacité de communication vise à améliorer la sécurité routière, optimiser la gestion du trafic et offrir des services connectés aux passagers. Cependant, les VANET présentent des caractéristiques particulières, telles que la mobilité élevée des nœuds, la variabilité topologique, la connectivité intermittente et l'hétérogénéité des environnements, qui rendent la conception de protocoles de communication efficaces particulièrement complexe. Ce chapitre propose une introduction générale aux VANET, en explorant leur définition, leur architecture, leurs applications, ainsi que les problématiques liées au routage et à la qualité de service (QoS) dans un tel contexte dynamique.

1.2 Introduction aux Réseaux VANET

Les réseaux ad hoc véhiculaires (VANET) sont un élément clé des systèmes de transport intelligents (ITS), permettant aux véhicules de communiquer entre eux et avec les infrastructures environnantes. Ces réseaux visent à améliorer la sécurité routière, à fluidifier le trafic et à fournir des services de divertissement en temps réel aux

passagers. Grâce à leur topologie dynamique et à leur forte mobilité, les VANET sont un sujet de recherche actif, notamment en ce qui concerne l'optimisation du routage et de la qualité de service dans des environnements complexes et changeants.

1.2.1 Définition des Réseaux VANET

Les réseaux ad hoc véhiculaires (VANET) constituent une sous-catégorie des réseaux ad hoc mobiles (MANET), où les véhicules agissent comme capteurs mobiles ou relais de données. Ces réseaux permettent la communication entre véhicules (V2V) et entre véhicules et infrastructures fixes (V2I), telles que les unités de bord de route (RSU). La Figure 1.1 illustre un exemple typique de réseau VANET, mettant en évidence les interactions dynamiques entre véhicules et infrastructures pour la gestion des alertes et l'amélioration de la sécurité routière [1, 2].

L'objectif des VANET est de faciliter l'échange d'informations critiques, en particulier dans le cadre de la sécurité routière, telles que la détection des collisions ou les alertes de changement de voie. En plus des applications de sécurité, ces réseaux sont également utilisés pour des services de gestion du trafic, l'assistance à la conduite, et des services d'infodivertissement, contribuant ainsi à des systèmes de transport intelligents. De plus, l'utilisation de technologies de communication avancées, comme le Wi-Fi, le LTE ou le DSRC, est essentielle pour assurer une transmission efficace et sécurisée des données échangées entre les véhicules et les infrastructures [1].

1.2.2 Architecture des Réseaux VANET

L'architecture des réseaux **VANET** se compose de nombreux composants interconnectés qui fonctionnent ensemble pour fournir divers services aux conducteurs. Il est nécessaire de comprendre ces composants et leur topologie pour développer des solutions innovantes basées sur l'intelligence artificielle.

1.2.2.1 Composants Principaux

Les principaux composants pour permettre les communications dans un réseau véhiculaire VANET sont les suivants :

1. **Nœud** : Les nœuds constituent les entités principales dans ce type de réseau. Il

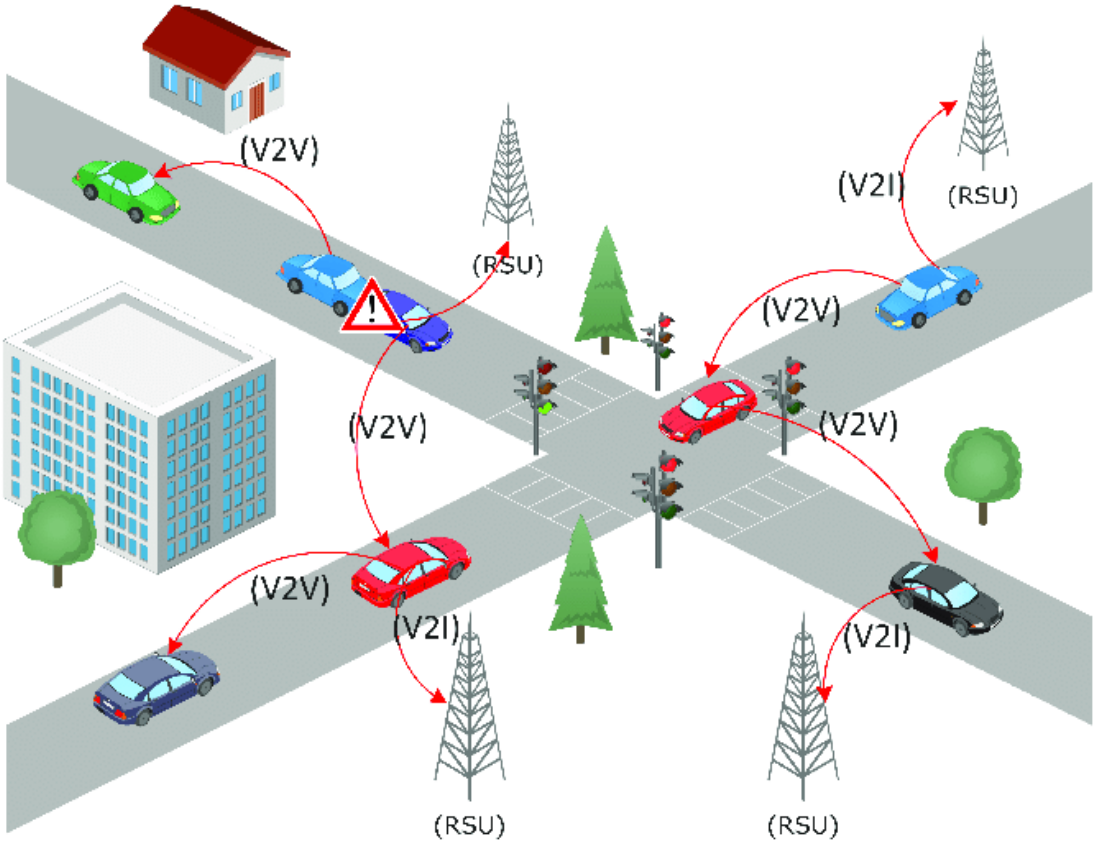


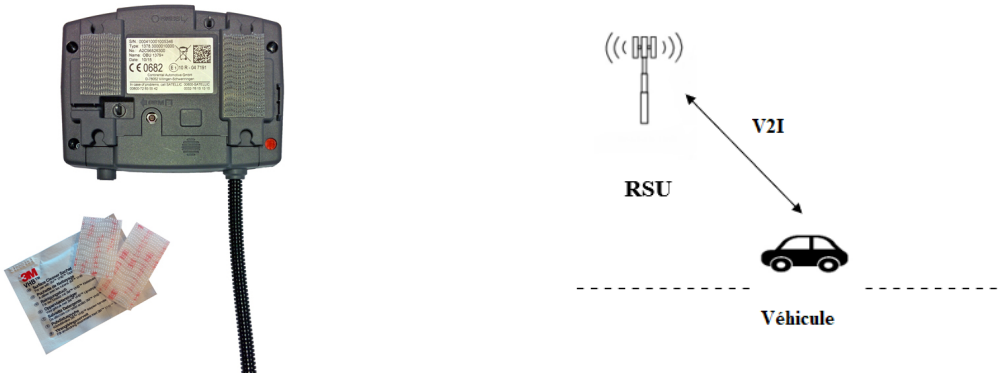
FIGURE 1.1 – Architecture de Communication VANET.

s'agit de véhicules intelligents équipés de technologies avancées telles que le GPS, des caméras et autres équipements.

2. **CA** (Certificate Authority) :La CA est un organisme chargé de sécuriser le réseau en délivrant des certificats numériques et en vérifiant l'identité des véhicules et dispositifs avant qu'ils ne rejoignent le réseau. Elle garantit des communications sûres et fiables entre les entités, minimisant ainsi les risques de fraudes et d'attaques malveillantes.
3. **OBU** (On-Board Unit) :L'OBU Figure 1.2(a) est un dispositif embarqué dans les véhicules intelligents qui collecte des données de conduite, affiche des informations de localisation, et facilite la communication avec d'autres véhicules et infrastructures. Ce dispositif joue un rôle crucial dans la sécurité routière, en permettant l'échange d'informations en temps réel sur les conditions de circulation et les dangers potentiels.
4. **AU** (Application Unit) :L'AU est un dispositif installé dans les véhicules qui assure la communication avec l'Autorité de Confiance (CA) et gère l'exécution des

applications de sécurité et de gestion des données. Elle peut également superviser le traitement des informations collectées par l'OBU, garantissant leur transmission sécurisée à la CA et aux autres véhicules.l'exécution des applications de sécurité et de données.

5. **RSU** (Road Side Unit) :Figure 1.2(b) Les RSUs sont des dispositifs situés le long des routes, jouant un rôle crucial dans la communication entre les véhicules et l'infrastructure. Ils transmettent des informations relatives au trafic, à la sécurité et à d'autres données importantes, ce qui favorise une gestion efficace des flux de circulation. En outre, les RSUs sont capables de collecter des données de trafic pour une analyse approfondie, ce qui améliore la prise de décision concernant la gestion du trafic.



((a)) Le Dispositif OBU ((b)) Road Side Unit

FIGURE 1.2 – Dispositifs de Communication Véhiculaire : OBU et RSU.

1.2.2.2 Topologie des Réseaux VANET

La topologie des réseaux VANET présente des caractéristiques spécifiques qui compliquent le routage des données en raison de la mobilité élevée et de la dynamique des nœuds. Voici les principales caractéristiques et défis des topologies de ces réseaux : [3]

1. **Mobilité élevée des nœuds** : La forte mobilité des véhicules entraîne des changements rapides de la topologie du réseau, rendant souvent les informations de routage obsolètes. Cela pose des défis pour maintenir des connexions stables entre les nœuds.
2. **Hétérogénéité de la densité des nœuds** : Dans certains scénarios comme les centres-villes ou les embouteillages, la densité des nœuds est très élevée, ce

qui augmente la probabilité de collisions de paquets et la surcharge du réseau. À l'inverse, dans des zones peu peuplées ou des autoroutes, la faible densité peut provoquer le partitionnement du réseau, empêchant la communication entre groupes isolés de véhicules.

3. **Partitionnement du réseau** : Ce phénomène se produit lorsque le réseau se divise en plusieurs sous-réseaux où les nœuds au sein d'un groupe peuvent communiquer entre eux, mais pas avec ceux des autres groupes.
4. **Intermittence de la connectivité** : les connexions entre les véhicules sont souvent instables et se coupent rapidement. Cela est causé par le mouvement rapide des voitures, ce qui rend la communication difficile.

1.2.3 Applications des Réseaux VANET

Les réseaux VANET transforment le domaine des transports en facilitant une communication directe et en temps réel entre les véhicules et les infrastructures. Ils ouvrent la voie à des progrès significatifs dans plusieurs domaines d'application, notamment : [4]

1.2.3.1 Sécurité Routière

Les réseaux VANET jouent un rôle crucial dans l'amélioration de la sécurité routière en permettant la communication entre les véhicules (V2V) et les infrastructures (V2I). Cela permet d'envoyer des alertes en temps réel sur les conditions de circulation, les dangers potentiels, ou les comportements dangereux tels que les freinages brusques. Les messages de sécurité critiques sont diffusés pour prévenir les accidents.

1.2.3.2 Gestion du Trafic

La gestion du trafic dans les réseaux VANET permet d'améliorer la fluidité de la circulation grâce à une coordination optimisée entre les véhicules et les infrastructures (comme les feux de circulation). Les informations sur les conditions de la route, les embouteillages et les itinéraires alternatifs sont partagées en temps réel, ce qui réduit la congestion et optimise l'efficacité des transports.

1.2.3.3 Divertissement et Services

Les réseaux VANET intègrent également des services de divertissement pour améliorer l'expérience des passagers. Les passagers peuvent bénéficier de services connectés tels que le streaming multimédia et d'autres applications de divertissement en voiture.

1.3 Problématiques du Routage dans les Réseaux VANET

Dans les réseaux VANET, la mobilité constante des véhicules entraîne des modifications fréquentes de la topologie, ce qui complique la maintenance de routes stables. Les connexions intermittentes, dues à des obstacles ou aux conditions environnementales, compliquent également la tâche. De plus, l'hétérogénéité des véhicules et des infrastructures contribue à la complexité du routage. Par conséquent, trouver un protocole de routage qui puisse gérer ces aspects tout en optimisant les performances est une question cruciale pour les réseaux VANET.

1.3.1 Protocole de Routage dans les Réseaux VANET

Les protocoles de routage dans les réseaux VANET jouent un rôle essentiel dans la facilitation d'une communication fluide et fiable entre les véhicules en mouvement. La dynamique de ces réseaux, caractérisée par des véhicules se déplaçant rapidement et changeant fréquemment de direction, nécessite des protocoles capables de s'adapter promptement aux variations de topologie. Cela implique l'application de techniques intelligentes pour identifier les itinéraires les plus efficaces pour la transmission des données, tout en réduisant les pertes de paquets et les temps de latence. De plus, ces protocoles doivent tenir compte de divers facteurs tels que la densité de circulation, la vitesse des véhicules et l'urgence de certains messages, ce qui représente un défi de taille. En intégrant des approches basées sur la géolocalisation et la structure topologique, ces protocoles cherchent à renforcer la fiabilité des communications et à répondre aux besoins spécifiques des applications liées à la sécurité routière et à la gestion du trafic.

1.3.1.1 Classification des Protocoles de Routage

La classification des protocoles de routage dans les réseaux VANET se base sur les méthodes utilisées pour acheminer les données entre les nœuds du réseau. Elle regroupe principalement deux grandes catégories : [5]

1. **Protocoles de routage basés sur la topologie :** Ces protocoles exploitent les informations de connectivité du réseau pour établir des chemins entre les nœuds. Ils se divisent en trois sous-catégories :

   - Protocoles réactifs : Ces protocoles, tels qu'AODV (Ad hoc On-Demand Distance Vector) et DSR (Dynamic Source Routing), établissent des routes uniquement lorsqu'elles sont nécessaires, ce qui réduit le nombre de messages de contrôle. AODV utilise des numéros de séquence pour garantir des routes récentes et nécessite moins de mémoire. En revanche, il peut générer des délais lors de la recherche de routes. DSR, quant à lui, intègre la route complète dans l'en-tête du paquet, évitant ainsi aux nœuds intermédiaires de conserver des informations de routage, mais peut entraîner une charge supplémentaire en cas de nombreuses requêtes.

   - Protocoles proactifs : Ces protocoles, comme OLSR (Optimized Link State Routing) et DSDV (Destination-Sequenced Distance Vector), maintiennent en permanence une vue à jour de la topologie du réseau. OLSR optimise la diffusion des informations de routage en utilisant des relais multipoints, réduisant ainsi le volume des messages de contrôle, mais il est vulnérable aux attaques. DSDV met à jour les routes soit périodiquement, soit lors de modifications de topologie, offrant une faible consommation de trafic de contrôle, mais souffrant d'une réactivité limitée aux changements rapides.

   - Protocoles hybrides : Ces protocoles, tels que ZRP (Zone Routing Protocol), combinent les méthodes réactives et proactives. ZRP divise le réseau en zones, utilisant un routage proactif à l'intérieur de chaque zone tout en adoptant une approche réactive pour les communications entre zones. Cela permet d'offrir des routes rapides tout en conservant la flexibilité nécessaire pour la découverte de routes externes.

2. **Protocoles de routage basés sur la géographie :** Les protocoles de routage géographiques utilisent les coordonnées GPS pour faciliter la navigation vers une

destination. Ils évitent la complexité des tables de routage en se concentrant sur la position des nœuds et de la destination. Par exemple, A-STAR (Anchor-based Street and Traffic Aware Routing) prend en compte les conditions de trafic pour déterminer les itinéraires les plus efficaces, tandis que GPSR (Greedy Perimeter Stateless Routing) relaye les paquets vers le nœud le plus proche de la destination en adoptant une approche opportuniste. Ces protocoles optimisent la communication dans des environnements dynamiques, améliorant ainsi la livraison des données tout en réduisant la signalisation.

1.3.1.2 Problèmes d'Optimisation

Les problèmes d'optimisation dans les protocoles de routage pour les réseaux VANET se concentrent principalement sur quatre grands défis : [6]

1. **Surcharge de communication :** Le réseau VANET a des capacités de communication limitées dues à la mobilité élevée des véhicules, au faible nombre de stations de base ou RSU , et à la fragmentation des connexions. Cela entraîne une surcharge qui diminue les performances globales du réseau. Une mauvaise configuration des RSU et des bases déployées peut également augmenter cette surcharge.
2. **Perte de paquets :** Une grande mobilité des véhicules peut entraîner des interruptions fréquentes de communication, ce qui augmente le risque de perte de paquets. L'absence d'une méthode efficace de sélection de relais ou de regroupement de véhicules aggrave ce problème, car elle empêche la transmission fiable des messages dans des conditions dynamiques.
3. **Délai de transmission :** Le processus de sélection du véhicule relais ou du cluster head est souvent sous-optimal, ce qui entraîne des retards dans la diffusion des messages d'urgence. Par ailleurs, les protocoles de routage classiques ne prennent pas toujours en compte des facteurs comme la vitesse du véhicule, sa direction, ou les conditions de la route, ce qui entraîne des retards de transmission importants.
4. **Diffusion inefficace des messages d'urgence :** La transmission des messages d'urgence est souvent limitée à une seule direction, sans considérer les autres zones qui pourraient également être touchées par l'événement d'urgence. Cela entraîne une diffusion incomplète ou inefficace des messages, ce qui peut avoir

des conséquences graves, notamment dans les situations critiques comme les accidents.

1.4 Qualité de Service (QoS) dans les Réseaux VANET

La Qualité de Service (QoS) dans les réseaux VANET est primordiale pour assurer des communications sûres et efficaces entre les véhicules. Elle est essentielle pour le bon fonctionnement des applications critiques, telles que les systèmes d'alerte et la gestion dynamique du trafic. En raison de l'environnement dynamique des réseaux VANET, garantir une QoS élevée représente un défi majeur pour la sécurité et la fiabilité des échanges d'informations.

1.4.1 Paramètres de la QoS

Les paramètres de qualité de service sont cruciaux pour assurer le bon fonctionnement des réseaux VANET. Parmi ces paramètres [7] :

1.4.1.1 Débit

Le débit est un facteur clé pour la QoS dans les VANETs. Il représente la quantité de données pouvant être transmises sur le réseau dans un temps donné. Les VANETs nécessitent un débit élevé pour répondre aux exigences des applications critiques, comme la transmission d'alertes de sécurité et l'accès à des services en temps réel.

1.4.1.2 Latence

La latence se réfère au temps écoulé entre l'envoi d'un message et sa réception. Dans les réseaux VANET, minimiser la latence est essentiel, surtout pour les services qui nécessitent des décisions en temps réel, tels que les avertissements de collision ou la gestion dynamique du trafic. Une latence élevée pourrait compromettre la sécurité des véhicules sur la route.

1.4.1.3 Variabilité du Délai (Jitter) et Pertes de Paquets

La variabilité du délai, aussi appelée jitter, est la fluctuation des temps de transmission de paquets dans le réseau. Un jitter important peut rendre les communications imprévisibles, affectant la performance des services en temps réel. En parallèle, les pertes de paquets se produisent lorsque des informations sont perdues en cours de transmission, ce qui peut compromettre l'intégrité des données échangées.

1.5 Conclusion

les réseaux VANET constituent une infrastructure essentielle pour les systèmes de transport intelligents, mais leur performance dépend fortement de la capacité à gérer efficacement les communications dans un environnement en constante évolution. La mobilité rapide des véhicules, les connexions instables et la diversité des scénarios routiers soulèvent de nombreuses problématiques, en particulier au niveau du routage et de la qualité de service. Une compréhension approfondie de l'architecture, des composants, des topologies et des applications des VANET est donc indispensable pour relever ces défis. Ce chapitre a permis d'établir les bases conceptuelles nécessaires à l'analyse et à la conception de solutions adaptées à ces contraintes. Les éléments abordés serviront de socle pour étudier, dans les chapitres suivants, les approches protocolaires existantes et les stratégies d'optimisation visant à améliorer la fiabilité et la performance des réseaux véhiculaires.

Le chapitre suivant est intitulé "État de l'art" et examine les recherches existantes dans le domaine des VANET, en détaillant les solutions déjà proposées pour résoudre les problèmes de routage et de qualité de service.

Chapitre 2

État de l'art

2.1 Introduction

Les réseaux VANET jouent un rôle crucial dans le développement des systèmes ITS, en permettant une communication efficace entre les véhicules et l'infrastructure routière. Cependant, en raison de la mobilité élevée des véhicules, des changements dynamiques de topologie et des contraintes de latence, le routage et l'amélioration de QoS dans ces réseaux restent des défis majeurs. Pour surmonter ces difficultés, différentes approches ont été développées, allant des protocoles de routage classiques aux solutions basées sur l'intelligence artificielle (IA). Ce chapitre explore l'état de l'art des techniques de routage et de gestion de la QoS dans les réseaux VANET, en mettant en avant les approches non-apprenantes et apprenantes. Nous présenterons les principaux protocoles utilisés, leurs avantages et leurs limites, tout en analysant comment l'intégration de l'IA peut transformer la gestion du routage et améliorer la performance globale du réseau.

2.2 Méthodologie de Recherche

L'approche méthodologique adoptée dans cette recherche repose sur une analyse rigoureuse des contributions scientifiques dans le domaine des réseaux VANET. En examinant les concepts fondamentaux et les avancées récentes, cette démarche vise à établir une compréhension approfondie des principaux défis et solutions existants. L'objectif est de fournir un cadre analytique permettant d'identifier les points faibles

des approches actuelles tout en mettant en évidence les opportunités d'amélioration. Cette méthodologie garantit que les résultats de l'étude s'appuient sur des bases solides et pertinentes.

2.2.1 Objectifs de l'Étude de la Littérature

L'étude de la littérature présentée dans ce chapitre a pour objectif d'offrir une analyse détaillée des recherches actuelles sur le routage et QoS dans les réseaux VANET. Cette exploration vise à identifier les protocoles et approches de routage les plus couramment utilisés, ainsi que les défis spécifiques inhérents à la nature dynamique et distribuée des réseaux de véhicules. En particulier, cette étude se concentre sur les solutions proposées pour faire face aux contraintes de mobilité élevée, aux connexions intermittentes et à l'hétérogénéité des réseaux, des caractéristiques propres aux VANET. Un autre objectif essentiel est d'examiner les contributions récentes de l'intelligence artificielle (IA) pour optimiser le routage et améliorer la QoS dans ces réseaux. En répertoriant et en analysant les différentes méthodes, cette revue de la littérature met en lumière les limitations des approches actuelles et identifie les opportunités où l'IA pourrait apporter des améliorations notables. Ce travail de synthèse offre ainsi une base théorique solide pour le développement de nouvelles solutions adaptées aux défis des réseaux VANET.

2.2.2 Méthodes et Outils Utilisés

Pour mener à bien cette étude de la littérature, diverses méthodes et outils ont été utilisés afin de garantir une analyse complète et fiable. Les bases de données académiques telles que ScienceDirect, SpringerLink, IEEE Xplore et ACM Digital Library ont été exploitées pour accéder à des articles récents et pertinents sur les réseaux VANET, le routage et l'intelligence artificielle. Le Service National de Documentation en Ligne (SNDL) a également été mobilisé pour consulter des ressources non accessibles au public. Pour organiser et gérer efficacement les références bibliographiques, le logiciel Zotero a été utilisé, permettant une classification systématique des sources et facilitant leur citation dans le cadre de cette recherche. L'ensemble de ces outils a permis d'assurer une couverture exhaustive et méthodique des travaux existants, en s'appuyant sur des ressources académiques fiables et à jour.

2.2.3 Critères d'Évaluation

Afin d'évaluer de manière rigoureuse les approches et solutions proposées dans les réseaux VANET, plusieurs critères essentiels ont été retenus. Ces critères permettent de structurer l'analyse et de comparer objectivement les performances des différentes stratégies étudiées. Ils englobent des aspects clés tels que la robustesse, la flexibilité, la scalabilité, la fiabilité des communications, la latence, ainsi que la sécurité et la confidentialité. Chaque critère est défini et expliqué en détail dans les sections suivantes [8].

2.2.3.1 Robustesse

La robustesse se réfère à la capacité du système à maintenir son fonctionnement malgré des conditions adverses, telles que des attaques ou des défaillances réseau. Dans les VANET, cela inclut la résilience contre les intrusions, comme les attaques par déni de service, tout en assurant une communication continue et sécurisée.

2.2.3.2 Flexibilité

La flexibilité désigne la capacité d'adaptation du système à des environnements dynamiques, comme des changements rapides de topologie réseau ou d'intensité de trafic. Un système flexible peut fonctionner efficacement dans des contextes variés, comme des zones urbaines denses ou des autoroutes.

2.2.3.3 Scalabilité

La scalabilité est la capacité du système à gérer une augmentation du nombre de nœuds ou de la densité du trafic sans dégradation des performances. Les VANET nécessitent des algorithmes capables de traiter des flux massifs de données tout en maintenant la qualité de service.

2.2.3.4 Fiabilité des communications

La fiabilité mesure la capacité à assurer un échange de données précis et sans erreurs entre les nœuds du réseau. Dans les VANET, cela inclut la vérification de l'intégrité des messages pour éviter leur modification ou corruption, même en cas d'attaques ou de conditions de réseau instables.

2.2.3.5 Latence

La latence représente le délai entre l'envoi et la réception des messages. Dans les VANET, où la rapidité est essentielle (par exemple, pour des alertes d'accidents), un système performant doit minimiser ce délai, même en présence de grandes quantités de données ou de nœuds connectés.

2.2.3.6 Sécurité et confidentialité

La sécurité et la confidentialité consistent à protéger les communications contre les accès non autorisés et la divulgation d'informations sensibles. Les VANET utilisent des méthodes comme l'authentification (ex. signatures numériques), le chiffrement des données, et des clés temporaires pour garantir que les messages ne sont ni falsifiés ni interceptés.

2.3 Contexte et Défis des Réseaux VANETs

Les réseaux VANET, bien que très prometteurs pour le développement de systèmes de transport intelligents, sont confrontés à divers défis techniques et pratiques. Ces défis découlent principalement de la nature dynamique et hétérogène des véhicules et des infrastructures routières.

2.3.1 Défis du Routage dans les Réseaux VANET

Les défis du routage dans les réseaux VANET sont liés à la dynamique unique de l'environnement, caractérisée par une forte mobilité des véhicules et une topologie en constante évolution. Ces défis exigent des solutions adaptées pour garantir des communications fiables et stables. Parmi ces défis, on trouve [9] :

2.3.1.1 Mobilité Élevée des Véhicules

La mobilité élevée des véhicules dans les réseaux VANET se réfère aux déplacements rapides et fréquents des véhicules, ce qui entraîne une modification constante de la topologie du réseau. En raison de la vitesse élevée, les connexions entre les nœuds sont souvent de courte durée, rendant difficile la mise en place de chemins de communi-

cation stables. Ces changements rapides perturbent les protocoles de routage traditionnels, qui ne sont pas conçus pour s'ajuster en temps réel à une telle dynamique. Il est donc nécessaire de développer des solutions adaptées pour maintenir une transmission continue des données malgré ces déconnexions fréquentes.

2.3.1.2 Hétérogénéité des Réseaux

Les VANET intègrent une grande diversité de dispositifs et d'infrastructures de communication, créant un réseau hétérogène. Les véhicules communiquent à la fois en V2V, en V2I et via les RSU. Cette diversité dans les types de nœuds et les protocoles de communication rend le processus de routage plus complexe. Il est nécessaire de concevoir des protocoles capables de gérer ces interactions entre différents types de réseaux et dispositifs de manière fluide et efficace.

2.3.1.3 Connexions Intermittentes

Les connexions intermittentes sont un problème majeur dans les VANET, principalement en raison de la mobilité élevée des véhicules et de la topologie changeante du réseau. Les véhicules entrent et sortent fréquemment de la portée de communication, provoquant des interruptions dans le flux de données. Ces déconnexions fréquentes nécessitent des protocoles de routage capables de rétablir rapidement les liens lorsque les connexions se rompent, afin de minimiser les perturbations dans la communication.

2.3.2 Défis de la Qualité de Service dans les Réseaux VANETs

Les réseaux VANET rencontrent plusieurs défis en matière de QoS en raison de leur nature dynamique et des environnements dans lesquels ils opèrent. Parmi ces défis, on peut citer [10] :

2.3.2.1 Variabilité des Conditions Réseau

La variabilité des conditions réseau dans les VANET constitue un défi majeur pour la QoS. En raison du dynamisme de la topologie, où les véhicules se déplacent à grande vitesse, la gestion des connexions devient complexe. Les liaisons peuvent être perdues à tout moment à cause d'obstacles physiques tels que des bâtiments ou des arbres, ainsi que des interférences électromagnétiques, entraînant des pertes de paquets. De plus, les

différentes formes de communication, qu'elles soient V2V ou V2I, présentent des défis uniques en termes de latence et de débit, rendant difficile l'établissement de connexions stables et fiables.

2.3.2.2 Sécurisation des Communications

La sécurisation des communications est un autre défi essentiel pour assurer la QoS dans les VANET. La protection des données échangées entre les véhicules est cruciale pour prévenir les interceptions et les accès non autorisés. Garantir des chemins de transmission sécurisés est indispensable pour éviter les attaques malveillantes, qui peuvent compromettre la sécurité des informations sensibles. De plus, les réseaux VANET peuvent être la cible d'attaques par déni de service (DoS), qui visent à rendre le réseau indisponible, augmentant ainsi la nécessité d'implémenter des mécanismes de sécurité robustes pour maintenir la fiabilité et la disponibilité des communications.

2.4 Classification des Protocoles de Routage et QoS dans les Réseaux VANETs

Dans les réseaux VANET, les protocoles sont classés en deux grandes catégories : les protocoles de routage et les protocoles de QoS. Les protocoles de routage visent à trouver les chemins optimaux pour la transmission des données entre les véhicules en se basant sur des mécanismes tels que la position, la topologie ou le regroupement (clustering). En revanche, les protocoles de qualité de service se concentrent sur l'amélioration des critères tels que la fiabilité, la réduction de la latence et l'utilisation efficace de la bande passante.

Cette section présente une classification détaillée des protocoles utilisés dans les réseaux VANET. Comme illustré dans la Figure 2.1 :

2.4.1 Protocoles de Routage dans les Réseaux VANET

Les protocoles de routage dans les réseaux VANET sont divisés principalement en trois catégories : basés sur la position, basés sur la topologie et basés sur le clustering. Chacun de ces protocoles vise à optimiser la communication en tenant compte de la

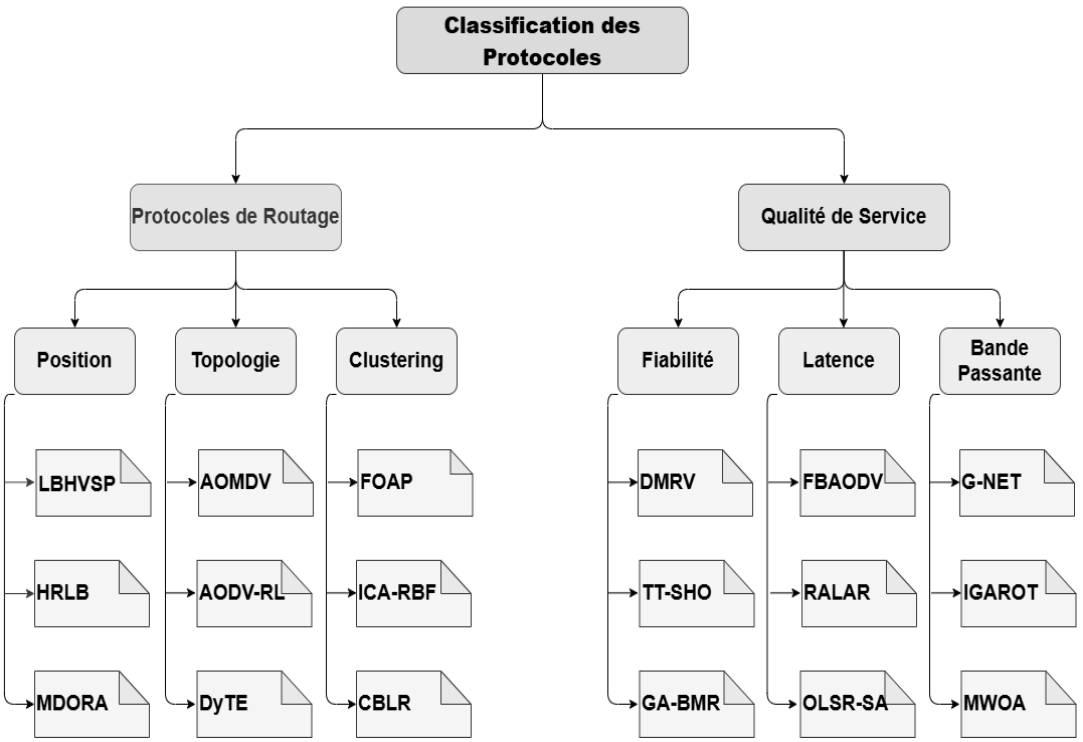


FIGURE 2.1 – Classification des protocoles de routage et QoS dans les réseaux VANET.

dynamique et de la mobilité des véhicules.

2.4.1.1 Protocoles Basés sur la Position

Ces protocoles exploitent les informations de localisation des véhicules (telles que leurs coordonnées GPS) pour établir des routes optimales. Ils permettent de minimiser les ressources utilisées pour la communication en réduisant la diffusion excessive de messages. Parmi ces protocoles, on peut citer :

1. **Location-Based Hybrid Video Streaming Protocol (LBHVSP)**

   Le LBHVSP est un protocole hybride spécialement conçu pour le streaming vidéo dans les réseaux VANET, en tenant compte des défis majeurs liés à la mobilité élevée, aux changements fréquents de topologie et à la connectivité intermittente des véhicules. Il se décompose en deux phases principales pour assurer une transmission vidéo efficace et fluide. Dans la première phase, lorsque le véhicule source et le véhicule destinataire se trouvent dans la portée de transmission directe, les données vidéo sont transmises directement vers le nœud suivant disponible (de deuxième saut), ce qui réduit le nombre de sauts nécessaires et minimise ainsi les délais de transmission. Cette méthode favorise une communication rapide en

temps réel. Dans la deuxième phase, si la destination est trop éloignée pour une transmission directe, les RSUs entrent en jeu en servant de relais grâce à leur connectivité filaire, permettant d'acheminer les paquets vidéo sur de longues distances tout en maintenant un faible délai. Le protocole intègre également une gestion dynamique de la puissance de transmission, ajustée en fonction de la distance entre les véhicules et les RSUs, afin d'optimiser l'utilisation de la bande passante, réduire la perte de paquets et améliorer le débit moyen des données. Cette approche garantit un streaming vidéo de haute qualité, même dans des environnements à forte mobilité, comme les VANET, où les véhicules se déplacent à grande vitesse. En exploitant les capacités hybrides de la communication directe et des relais via les RSUs, le LBHVSP se distingue par sa robustesse face aux défis des connexions intermittentes et à la complexité dynamique des réseaux véhiculaires, tout en assurant une latence minimale et une expérience utilisateur optimale pour le streaming en temps réel [11].

2. **Hierarchical Routing with Load Balancing(HRLB)**

   Le HRLB est un protocole de routage hiérarchique conçu pour les réseaux VANET définis par logiciel (SDVN – Software Defined Vehicular Networks). Il exploite l'architecture SDN pour bénéficier d'un contrôle centralisé et d'une vision globale du réseau, permettant ainsi une gestion efficace de la charge réseau. Le fonctionnement du protocole HRLB repose sur trois étapes hiérarchiques. D'abord, la région urbaine est divisée en plusieurs grilles géographiques, dont certaines sont sélectionnées en fonction de leur densité de véhicules et de la probabilité de transfert inter-grilles, afin d'identifier les zones à forte connectivité. Ensuite, le contrôleur central choisit deux chemins de transmission optimaux parmi les grilles sélectionnées, en utilisant une fonction de coût prenant en compte la longueur du chemin, la densité de trafic, l'espacement entre les véhicules et la charge réseau. Enfin, sur chacun des chemins choisis, des nœuds relais sont sélectionnés en fonction d'une fonction d'utilité qui combine la taille du tampon restant et la proximité avec la destination. Le protocole prévoit également un mécanisme de basculement vers un chemin de secours (NO.2 routing table) lorsque le premier (NO.1 routing table) devient congestionné. Les simulations, réalisées à l'aide de données GPS réelles de taxis à Shanghai, montrent que HRLB surpasse plusieurs

protocoles existants en termes de taux de livraison, de débit, de délai moyen et de nombre moyen de sauts, grâce à son approche centralisée et sa gestion intelligente de la charge réseau [12].

3. **Maximum Distance On-Demand Routing Algorithm (MDORA)**

   Le MDORA est un protocole de routage basé sur la localisation, conçu spécifiquement pour les réseaux VANET. Son fonctionnement repose sur deux étapes principales : la découverte de route et l'établissement de route. Lors de la découverte de route, un véhicule source envoie un message de découverte (Hello Message) à ses voisins. Les véhicules voisins qui partagent la même orientation répondent avec un message de réponse, permettant au véhicule source de calculer la distance et de construire une table des voisins. Ces voisins sont classés par ordre croissant de distance, favorisant une communication optimale. Dans l'étape de l'établissement de route, le véhicule source choisit le véhicule le plus éloigné mais toujours dans sa portée comme prochain saut, ce qui réduit le nombre total de sauts nécessaires pour atteindre la destination. MDORA se distingue par son efficacité à minimiser le délai de bout en bout (E2E delay) et sa capacité à maintenir un ratio élevé de livraison de paquets (PDR), en particulier dans des environnements à densité élevée ou avec des mouvements directionnels. Par ailleurs, son approche réduit la surcharge de communication, car il n'échange des messages de contrôle que lorsque nécessaire, contrairement à d'autres protocoles comme AODV. En résumé, MDORA optimise le routage en exploitant la position et la direction des véhicules, en garantissant des communications rapides et fiables dans les réseaux VANET [13].

Tableau2.1 présente une comparaison des algorithmes en fonction de plusieurs mesures clés.

Algorithme	**Robustesse**	**Flexibilité**	**Scalabilité**	**Fiabilité**	**Latence**	**Sécurité**
LBHVSP	Élevée	Élevée	Élevée	Élevée	Faible	Moyenne
HRLB	Élevée	Moyenne	Élevée	Élevée	Faible	Moyenne
MDORA	Élevée	Élevée	Élevée	Élevée	Faible	Élevée

TABLE 2.1 – Comparaison des algorithmes basés sur la position.

2.4.1.2 Protocoles Basés sur la Topologie

Ces protocoles utilisent les informations structurelles du réseau, telles que les liens entre les nœuds, pour le routage. Ils s'adaptent mieux aux changements rapides de la topologie, typiques des réseaux VANET. Les exemples incluent :

1. **Ad hoc On-demand Multipath Distance Vector(AOMDV)**

   Le AOMDV est une extension du protocole AODV (Ad hoc On-demand Distance Vector) qui permet de calculer plusieurs chemins entre la source et la destination lors de la demande de routage. Contrairement à AODV, qui établit un seul chemin, AOMDV permet de découvrir et d'utiliser plusieurs chemins simultanément. Lorsqu'un nœud source envoie une demande de route (RREQ), les nœuds intermédiaires répondent avec plusieurs réponses (RREP), fournissant ainsi des chemins alternatifs vers la destination. Ce mécanisme permet de renforcer la fiabilité du réseau, car en cas de défaillance d'un chemin, un autre peut immédiatement prendre le relais, réduisant ainsi les délais de reconfiguration. Cependant, l'utilisation de multiples chemins peut entraîner une surcharge de gestion des ressources, car il faut maintenir et suivre plusieurs routes simultanément. Bien que certaines implémentations puissent utiliser des informations de localisation (comme les données GPS) pour optimiser le choix des chemins, ce n'est pas une exigence inhérente au protocole. Malgré ces limitations, AOMDV est particulièrement efficace dans des réseaux où la connectivité est instable, comme dans les VANET, car il améliore la tolérance aux pannes et la fiabilité du réseau [14].

2. **AODV with Reinforcement Learning(AODV-RL)**

   Le AODV-RL est une amélioration du protocole AODV classique spécifiquement conçu pour relever les défis des VANET. Les VANET, caractérisés par leur haute mobilité, des changements fréquents de topologie et des connexions instables, nécessitent des mécanismes de routage robustes pour garantir une communication efficace et sécurisée. AODV-RL intègre un système de réputation basé sur l'apprentissage par renforcement pour évaluer la fiabilité des nœuds. Chaque véhicule conserve une base de données de réputation, mise à jour en fonction des interactions avec ses voisins, permettant d'identifier et d'isoler les nœuds malveillants ou non fiables. Grâce à ce mécanisme, les chemins de routage sont dynamiquement adaptés pour éviter les nœuds présentant un risque de sécurité. L'apprentissage

par renforcement est utilisé pour ajuster les Q-valeurs, qui représentent la qualité des chemins, en récompensant les comportements fiables et en pénalisant les comportements nuisibles. Ce processus permet au protocole de s'adapter en temps réel aux changements rapides de la topologie du réseau. Cependant, cette approche peut entraîner des délais supplémentaires, car les décisions de routage dépendent des mises à jour de la réputation et de l'ajustement des valeurs de qualité, ce qui peut être coûteux en termes de calcul et de communication dans des réseaux très dynamiques. Malgré ces limitations, AODV-RL améliore significativement la sécurité et la fiabilité du routage, en augmentant le taux de livraison des paquets et en réduisant les impacts des attaques malveillantes, tout en répondant aux exigences spécifiques des réseaux VANET où la mobilité rapide est une contrainte majeure [15].

3. **Dynamic Tri-lateral Enrollment(DyTE)**

   Le DyTE est un protocole de routage dynamique spécifiquement conçu pour les réseaux VANET, où la mobilité des véhicules entraîne des changements rapides de topologie. Ce protocole sélectionne dynamiquement les nœuds participant à la communication entre la source et la destination, en fonction de leur position et de leur disponibilité. Il s'appuie sur une méthode de sélection dite "tri-latérale", qui évalue la distance et la proximité des nœuds voisins pour garantir une communication optimale. Ce mécanisme permet de choisir les nœuds les plus proches et les plus fiables afin de maintenir une connexion stable pendant la transmission des données. Le DyTE vise à minimiser la perte de paquets et à améliorer la fiabilité des communications dans des réseaux où la densité et la mobilité des véhicules sont élevées. Cependant, cette approche peut entraîner une certaine complexité computationnelle, car les itinéraires doivent être recalculés fréquemment pour s'adapter aux changements de topologie. Bien que prometteur, le protocole DyTE pourrait nécessiter une optimisation supplémentaire pour réduire la surcharge liée aux calculs fréquents dans des environnements très dynamiques. Malgré ces défis, DyTE reste une solution prometteuse pour répondre aux défis des VANET en assurant une transmission efficace et fiable des données [11].

Tableau 2.2 illustre une comparaison des algorithmes en se basant sur des mesures clés.

Algorithme	Robustesse	Flexibilité	Scalabilité	Fiabilité	Latence	Sécurité
AOMDV	Élevée	Moyenne	Moyenne	Élevée	Faible	Moyenne
AODV-RL	Élevée	Élevée	Moyenne	Élevée	Moyenne	Élevée
DyTE	Moyenne	Élevée	Moyenne	Élevée	Faible	Élevée

TABLE 2.2 – Comparaison des algorithmes basés sur la topologie.

2.4.1.3 Protocoles Basés sur le Clustering

Les protocoles basés sur le clustering regroupent les véhicules en clusters, chaque cluster étant géré par un chef (Cluster Head). Cette approche améliore la scalabilité et réduit la surcharge de communication dans les réseaux VANET, en optimisant la gestion et la formation des clusters à l'aide d'algorithmes avancés. Nous présentons ci-après quelques exemples illustratifs :

1. **Falcon Optimization Algorithm-based Communication Protocol(FOAP)**

   Le protocole FOAP est conçu pour améliorer l'efficacité énergétique et la fiabilité du routage dans les réseaux VANET en utilisant des techniques d'optimisation avancées. Ce protocole utilise l'algorithme d'optimisation Falcon pour former des clusters de véhicules basés sur des fonctions d'adéquation, où chaque véhicule est attribué à un cluster en fonction de paramètres tels que la distance, la vitesse relative, et la qualité des liens. Le processus de formation des clusters est ensuite renforcé par l'algorithme de recherche des moineaux (Sparrow Search Algorithm), qui optimise les chemins de routage en tenant compte de la charge du réseau et des variations dynamiques de la topologie. En adoptant cette double approche d'optimisation, le protocole vise à réduire la latence et à améliorer le débit tout en minimisant la consommation énergétique des nœuds. Cependant, bien que performant, le protocole peut rencontrer des difficultés d'évolutivité dans les réseaux de grande taille en raison des ressources computationnelles importantes nécessaires pour effectuer ces calculs en temps réel [15].

2. **Imperialist Competitive Algorithm with Radial Basis Function (ICA-RBF)**

   Le ICA-RBF est un protocole de routage conçu pour optimiser les communications dans les réseaux VANET en utilisant une combinaison de l'algorithme compétitif impérialiste (ICA) et des fonctions de base radiale (RBF). Ce protocole se concentre sur le regroupement des nœuds en clusters dynamiques en

fonction de paramètres tels que la densité des véhicules, leur vitesse et leurs trajectoires. L'algorithme ICA est utilisé pour former des clusters en tenant compte de la mobilité des nœuds et des variations de la topologie du réseau. Ensuite, les fonctions de base radiale (RBF) sont appliquées pour sélectionner efficacement le Cluster Head (CH) dans chaque cluster, en se basant sur des critères comme l'espace mémoire disponible, la stabilité des liens et le nombre de voisins. Cette approche permet de maintenir des clusters stables même dans des environnements à forte mobilité, ce qui améliore la fiabilité des transmissions de données et réduit les pertes de paquets. Cependant, le protocole peut rencontrer des défis lorsqu'il s'agit d'adapter rapidement les formations de clusters en réponse aux changements rapides de la densité des véhicules et de la vitesse, en raison de la complexité computationnelle de ses algorithmes sous-jacents [15].

3. **Cluster-Based Location Routing (CBLR)**

   Le CBLR est un protocole de routage conçu pour améliorer l'efficacité et la fiabilité des communications dans les réseaux VANET en exploitant une structure basée sur des clusters. Dans ce protocole, les véhicules sont regroupés en clusters dynamiques, chaque cluster étant supervisé par un CH, qui coordonne la communication entre les membres du cluster et gère les interactions avec les autres clusters. La formation des clusters est déterminée en fonction des positions géographiques des véhicules, ce qui permet de minimiser la complexité des décisions de routage et de réduire les délais de communication. Le CBLR s'appuie également sur des critères tels que la stabilité des liens, la densité des nœuds et la distance entre les véhicules pour garantir une transmission de données fiable, même dans des environnements à mobilité élevée. Cependant, le protocole peut être limité dans des scénarios où la densité de trafic est faible ou où la topologie du réseau change rapidement, car cela peut entraîner une fragmentation des clusters et une augmentation des pertes de paquets. Néanmoins, en s'appuyant sur une gestion hiérarchique des clusters, le CBLR est particulièrement efficace pour réduire la consommation de bande passante et améliorer la qualité de service dans des applications nécessitant une transmission continue de données [15].

Le Tableau 2.3 offre une comparaison entre les algorithmes en se basant sur plusieurs mesures clés.

Algorithme	Robustesse	Flexibilité	Scalabilité	Fiabilité	Latence	Sécurité
FOAP	Moyenne	Élevée	Moyenne	Élevée	Faible	Moyenne
ICA-RBF	Élevée	Élevée	Élevée	Élevée	Faible	Élevée
CBLR	Moyenne	Moyenne	Élevée	Élevée	Faible	Moyenne

TABLE 2.3 – Comparaison des algorithmes basés sur le Clustering.

2.4.1.4 Limites des Protocoles de Routage

Les limites des protocoles de routage dans les VANET concernent principalement leur capacité à gérer la mobilité élevée, la topologie dynamique et les densités variables des réseaux. Ces protocoles font face à des défis tels que les interruptions de connexion, la complexité computationnelle, et l'adaptabilité limitée dans des environnements très dynamiques. Voici une synthèse des principales limitations identifiées [15] :

1. **Mobilité et topologie dynamique :** Les VANET se caractérisent par une architecture très mobile et dynamique, où les nœuds entrent et sortent fréquemment du réseau. Cela entraîne des interruptions fréquentes des routes, ce qui complique le routage efficace des paquets d'un véhicule émetteur à un véhicule récepteur.
2. **Efficacité limitée dans les réseaux denses :** Les solutions actuelles (comme ICA-RBF) manquent de flexibilité pour s'adapter aux variations des densités des réseaux et aux vitesses des véhicules, ce qui limite leur capacité à évoluer dans des scénarios de grande échelle ou denses.
3. **Impact de la perte de connexion et des interférences :** Les perturbations des chemins de routage peuvent entraîner des échecs d'application et une dégradation des performances du réseau, particulièrement dans des conditions de mobilité élevée et des topologies variables.
4. **Complexité des calculs :** Certaines approches comme celles utilisant la logique floue ou les algorithmes d'optimisation (par exemple, l'algorithme de Falcon) introduisent un surcoût computationnel important, réduisant leur faisabilité en temps réel, notamment dans les environnements à forte mobilité et réseau dense.

2.4.2 Protocoles Basés sur la QoS dans les Réseaux VANET

Dans les réseaux VANET, la QoS est une composante essentielle pour assurer une communication fiable et efficace, surtout pour les applications critiques telles que les

systèmes d'alerte d'urgence. Les protocoles basés sur la QoS sont conçus pour garantir que les communications répondent à des exigences strictes en matière de latence, de fiabilité et de bande passante. Ces protocoles sont classés en fonction de leur capacité à optimiser ces paramètres.

2.4.2.1 Protocoles Basés sur la Latence

La latence est un paramètre essentiel dans les réseaux VANET, en particulier pour les applications en temps réel comme les alertes de collision ou la gestion dynamique du trafic. Les protocoles axés sur la réduction de la latence optimisent les chemins de transmission pour minimiser les délais tout en maintenant la fiabilité des communications. Cela garantit des prises de décision rapides et efficaces, cruciales dans des environnements où chaque milliseconde compte.

1. **Delay-based Multipath Routing for VANET(DMRV)**

   Le protocole DMRV est conçu pour les réseaux VANET, où il se concentre sur la réduction des délais dans la transmission des données, tout en utilisant plusieurs chemins alternatifs pour garantir une communication fiable et rapide. Contrairement aux protocoles classiques de routage multipath, DMRV sélectionne les chemins non seulement en fonction de la distance ou de la qualité du lien, mais aussi en tenant compte du délai de transmission. Ce protocole est particulièrement adapté aux applications sensibles au temps, comme les alertes d'urgence ou les communications en temps réel, où le délai est crucial. Il fonctionne en analysant les différents chemins disponibles et choisit ceux qui offrent le moins de latence pour chaque paquet de données. DMRV utilise des messages d'alerte pour identifier les chemins à faible délai et gérer les défaillances de lien en temps réel, ce qui permet de maintenir la continuité de la communication. Cependant, ce protocole peut rencontrer des problèmes de surcharge de gestion des chemins lorsque le réseau devient dense, et il peut nécessiter des ressources supplémentaires pour maintenir une faible latence dans des environnements très dynamiques [14].

2. **Tent Map-Spotted Hyena Optimization(TT-SHO)**

   Le TT-SHO est un protocole de routage conçu pour améliorer la QoS et la sécurité dans les réseaux VANET. Il combine une approche d'optimisation inspirée du comportement des hyènes tachetées (Spotted Hyena Optimization, SHO) et une

méthode de réglage basée sur la distribution chaotique Tent Map. Ce protocole vise à résoudre les problèmes de routage liés à la dynamique élevée des VANET, en optimisant les chemins de routage tout en garantissant la sécurité des données transmises. L'algorithme SHO est ajusté par la carte Tent pour améliorer sa convergence et explorer efficacement l'espace des solutions, ce qui permet de choisir les routes optimales. En parallèle, le protocole intègre une méthode de chiffrement hybride chaotique pour protéger les données pendant leur transmission, répondant ainsi aux exigences de sécurité des applications critiques. Les principaux objectifs du TT-SHO incluent la réduction des délais, l'augmentation du débit et la fiabilité dans des environnements à mobilité élevée et à densité variable. Cependant, en raison de la complexité computationnelle des algorithmes d'optimisation et des méthodes de chiffrement utilisées, le TT-SHO peut présenter des défis en termes de scalabilité et de charge de calcul dans des réseaux de grande taille ou à très forte densité de véhicules [15].

3. **GA-based Multipath Routing(GA-BMR)**

   Le protocole GA-BMR utilise des algorithmes génétiques pour optimiser la sélection de plusieurs chemins de communication dans les réseaux, y compris les VANET. Les VANET, caractérisés par une mobilité rapide et des changements fréquents de topologie, nécessitent des mécanismes robustes pour maintenir des communications fiables. Les algorithmes génétiques sont des techniques d'optimisation inspirées des processus de sélection naturelle, où les solutions (chemins de routage) sont évaluées en fonction de critères tels que la longueur du chemin, la qualité du lien, la consommation d'énergie et la fiabilité du chemin. Dans un environnement VANET, ce protocole génère une population de solutions candidates et les évalue à travers des générations successives, en appliquant des opérateurs génétiques comme la sélection, le croisement et la mutation pour améliorer les chemins au fil du temps. En sélectionnant plusieurs chemins parallèles, le GA-based Multipath Routing améliore la résilience du réseau en permettant la redondance et la tolérance aux pannes, ce qui est essentiel dans les réseaux véhiculaires. Cependant, l'utilisation des algorithmes génétiques peut être coûteuse en termes de calcul, surtout dans les VANET à grande échelle où la mobilité élevée exige des recalculs fréquents, entraînant une latence accrue et une consommation d'énergie

plus élevée [14].

Le Tableau 2.4 évalue les algorithmes en fonction de diverses mesures clés.

Algorithme	**Robustesse**	**Flexibilité**	**Scalabilité**	**Fiabilité**	**Latence**	**Sécurité**
DMRV	Moyenne	Faible	Faible	Élevée	Faible	Faible
TT-SHO	Élevée	Élevée	Moyenne	Élevée	Faible	Élevée
GA-BMR	Élevée	Élevée	Élevée	Élevée	Faible	Moyenne

TABLE 2.4 – Comparaison des algorithmes basés sur la latence.

2.4.2.2 Protocoles Basés sur la Fiabilité

Ces protocoles garantissent la livraison des données même dans des environnements à forte mobilité et à faible connectivité. Ils jouent un rôle essentiel pour les applications sensibles. Parmi eux, on trouve [16] :

1. **Fitness-based AODV (FBAODV)**

   Le protocole FBAODV est une amélioration du protocole AODV qui intègre une fonction de fitness pour évaluer et sélectionner les chemins de communication dans les réseaux VANET. Cette fonction prend en compte des critères tels que la latence, la bande passante, et la fiabilité du lien, permettant ainsi de choisir les meilleurs chemins pour la transmission des données. Le principal avantage de FBAODV est sa capacité à s'adapter aux environnements mobiles et dynamiques des VANET, où la topologie du réseau change fréquemment. Ce protocole est particulièrement adapté aux applications de sécurité, comme les messages urgents, où la latence doit être réduite au minimum tout en garantissant la fiabilité de la communication. Cependant, dans des environnements très denses avec une forte mobilité, FBAODV peut rencontrer des problèmes de surcharge computationnelle, ce qui peut entraîner une augmentation de la latence et un surcoût énergétique.d'énergie accrue.

2. **Genetic Optimized Location Aided Routing(RALAR)**

   Le RALAR est un protocole de routage dans les réseaux VANET qui utilise des données GPS et un algorithme génétique pour optimiser le choix des chemins. Ce protocole permet de réduire la latence en rejetant les données GPS incorrectes et en sélectionnant les itinéraires les plus efficaces en fonction de la topologie

dynamique du réseau. RALAR est conçu pour être particulièrement efficace dans des environnements où la mobilité des véhicules et la densité du réseau varient rapidement. En utilisant l'algorithme génétique, le protocole peut trouver des chemins optimaux qui maximisent la fiabilité et la latence tout en conservant une faible consommation d'énergie. Cependant, comme tous les algorithmes génétiques, RALAR peut rencontrer des problèmes de scalabilité dans des réseaux très vastes, où les calculs nécessaires pour ajuster les paramètres du routage deviennent coûteux.

3. **Optimized Link State Routing with Simulated Annealing (OLSR-SA)**

   Le protocole OLSR-SA intègre Simulated Annealing dans le protocole OLSR pour améliorer l'optimisation des paramètres de routage, réduisant ainsi la latence tout en maintenant une bande passante efficace. L'algorithme de Simulated Annealing permet de trouver des configurations de routage optimales, même dans des environnements à topologie dynamique. Il est particulièrement adapté aux réseaux VANET, où les véhicules se déplacent rapidement et où la latence est un facteur crucial pour la transmission de messages de sécurité. Cependant, cette approche d'optimisation peut entraîner un coût computationnel élevé, ce qui limite son efficacité dans des réseaux très denses et à haute mobilité.

Le Tableau 2.5 présente une comparaison des algorithmes en fonction de plusieurs mesures clés.

Algorithme	**Robustesse**	**Flexibilité**	**Scalabilité**	**Fiabilité**	**Latence**	**Sécurité**
FBAODV	Moyenne	Faible	Faible	Moyenne	Faible	Faible
RALAR	Élevée	Élevée	Moyenne	Élevée	Faible	Faible
OLSR-SA	Élevée	Élevée	Moyenne	Élevée	Faible	Moyenne

TABLE 2.5 – Comparaison des algorithmes basés sur la Fiabilité.

2.4.2.3 Protocoles Basés sur la Bande Passante

Ces protocoles optimisent l'utilisation de la bande passante en réduisant les congestions et en maximisant le débit, essentiels pour des applications exigeantes comme la diffusion vidéo en temps réel. [16]

1. **G-NET (Genetic Network Protocol)**

Le protocole G-NET repose sur l'utilisation d'algorithmes génétiques pour optimiser la sélection des chemins de communication dans un réseau VANET. Ce protocole utilise des chemins parallèles pour améliorer la bande passante et garantir une transmission efficace des données, tout en maintenant une faible latence. En évaluant les solutions de routage à travers plusieurs générations et en appliquant des opérateurs génétiques comme la sélection, le croisement et la mutation, G-NET est capable de sélectionner les meilleures routes disponibles en fonction de la qualité du lien et de la vitesse des véhicules. Cependant, les calculs fréquents dans un environnement dynamique peuvent augmenter la latence et la consommation d'énergie.

2. **IGAROT (Improved Genetic Algorithm-based Route Optimization Technique)**

   Le protocole IGAROT utilise un algorithme génétique amélioré pour sélectionner les meilleurs chemins dans les réseaux VANET. En combinant la clusterisation K-means et la sélection génétique, ce protocole optimise la bande passante en sélectionnant les routes les plus fiables et les plus efficaces. IGAROT permet une gestion dynamique des ressources du réseau, tout en réduisant la latence et en améliorant la fiabilité. Cependant, la complexité des calculs dans de grands réseaux peut entraîner des coûts de traitement élevés et une latence accrue, surtout dans des environnements à haute mobilité.

3. **MWOA (Moth Whale Optimization Algorithm)**

   Le MWOA est un protocole hybride combinant Moth Search (MS) et Whale Optimization Algorithm (WOA) pour optimiser les chemins multipoints dans les réseaux VANET. Ce protocole est particulièrement adapté pour les applications de vidéo en temps réel, où la bande passante et la latence sont des facteurs clés. MWOA permet de sélectionner les meilleurs chemins de transmission tout en maintenant une bande passante élevée et en réduisant la latence. Cependant, il peut présenter des défis en termes de complexité computationnelle, en particulier dans des réseaux très denses ou à haute mobilité.

Le Tableau 2.6 propose une comparaison des algorithmes en se basant sur des mesures clés.

Algorithme	Robustesse	Flexibilité	Scalabilité	Fiabilité	Latence	Sécurité
G-NET	Moyenne	Moyenne	Moyenne	Élevée	Faible	Moyenne
IGAROT	Élevée	Moyenne	Moyenne	Élevée	Faible	Faible
MWOA	Élevée	Moyenne	Moyenne	Élevée	Faible	Moyenne

TABLE 2.6 – Comparaison des algorithmes basés sur la bande passante.

2.4.2.4 Limites des Protocoles Basés sur la QoS

Les protocoles QoS dans les réseaux VANET sont conçus pour optimiser la performance du réseau, mais plusieurs facteurs peuvent nuire à leur efficacité. Il est important de prendre en compte les limites suivantes [14] :

1. **Augmentation du temps de transmission de bout en bout :** Certains protocoles, malgré leur objectif d'optimisation, entraînent une augmentation du délai global de transmission en raison des traitements complexes nécessaires pour sélectionner les meilleures routes. Les algorithmes d'optimisation avancés, tels que ceux basés sur les algorithmes génétiques ou l'optimisation multi-objectifs, peuvent générer des surcharges computationnelles qui ralentissent la transmission des données.
2. **Surcharge des messages de contrôle :** La mise à jour constante des chemins de routage exige l'envoi fréquent de messages de contrôle, ce qui surcharge le réseau et augmente le temps de traitement. Cette surcharge est particulièrement problématique dans les réseaux denses, où un grand nombre de véhicules doit échanger des informations en continu pour maintenir une communication stable et fiable.
3. **Problèmes de latence et de retard :** Dans les applications critiques telles que les alertes d'urgence, une faible latence est essentielle. Cependant, certains protocoles QoS échouent à garantir des délais acceptables en raison de leur complexité de calcul et du nombre élevé de décisions à prendre en temps réel. Cette situation peut entraîner des retards dans la livraison des messages critiques et affecter la réactivité du réseau.
4. **Scalabilité limitée :** Les protocoles QoS conçus pour optimiser la fiabilité ou la bande passante peuvent rencontrer des difficultés d'adaptation dans des environnements très denses. Lorsque le nombre de véhicules augmente, la congestion du

réseau s'accentue et la gestion des ressources devient plus complexe, ce qui peut dégrader la performance du routage et augmenter le risque de saturation.

2.5 Évolution des Techniques de Routage et de QoS dans les Réseaux VANET

Les réseaux VANET nécessitent des protocoles de routage efficaces pour garantir une communication rapide et fiable entre les véhicules. Le routage dans ces réseaux doit non seulement s'adapter à la forte mobilité et aux changements de topologie, mais aussi assurer une QoS pour les applications sensibles au délai. L'évolution des techniques de routage et de QoS reflète la transition des approches conventionnelles aux méthodes modernes, l'intégration des techniques d'optimisation et des algorithmes heuristiques, et l'impact des technologies émergentes.

2.5.1 Approches Conventionnelles vs Approches Modernes

Dans les réseaux VANET, le routage est un enjeu central pour assurer une communication efficace entre les véhicules. Différentes stratégies ont été développées pour répondre aux défis liés à la mobilité et à la dynamique du réseau.Ces stratégies peuvent être regroupées en approches conventionnelles et approches modernes, chacune présentant des caractéristiques et des mécanismes spécifiques [17] [18].

2.5.1.1 Approches Conventionnelles

Les approches conventionnelles de routage dans les réseaux VANET se basent généralement sur des protocoles décentralisés, où chaque véhicule agit comme un nœud qui communique directement avec ses voisins. Parmi les protocoles les plus utilisés figurent l'AODV et le DSR . Ces protocoles permettent la découverte et la gestion des routes dans un réseau où les positions des véhicules changent fréquemment.

Cependant, les protocoles conventionnels rencontrent plusieurs défis, tels que la scalabilité du réseau, la réactivité en temps réel et un contrôle limité de la QoS. Ces systèmes peuvent souffrir de latence élevée, de pertes de paquets et de communications peu fiables, particulièrement dans des environnements urbains densément peuplés.

2.5.1.2 Approches Modernes

Les approches modernes du routage dans les réseaux VANET intègrent des techniques centralisées, souvent basées sur des architectures comme le Software-Defined Networking (SDN), qui sépare les plans de contrôle et de données. Cela permet une gestion plus flexible et en temps réel du réseau, améliorant considérablement la QoS en adaptant dynamiquement le comportement du réseau en fonction des conditions de circulation et de l'environnement.

L'IA et l'apprentissage automatique, lorsqu'ils sont intégrés dans ces réseaux, permettent de prédire les modèles de circulation et d'ajuster les routes en temps réel pour réduire la congestion et les retards. Cela répond aux besoins de fiabilité des systèmes de transport intelligents, en assurant une meilleure continuité de service dans des conditions de trafic variées. De plus, la 5G et le edge computing renforcent ces capacités en augmentant la bande passante et en réduisant la latence, offrant ainsi des échanges de données plus rapides et stables entre les véhicules. Ces innovations créent une infrastructure plus réactive et évolutive, essentielle pour la mobilité connectée de demain.

2.5.2 Impact des Technologies Émergentes sur le Routage et la QoS

Les technologies émergentes, notamment la 5G et l'IA et l'edge computing, transforment le routage et la QoS dans les réseaux VANET. Voici quelques impacts majeurs [19] :

1. **Amélioration du Routage Dynamique :** Les nouvelles technologies permettent une gestion plus dynamique et réactive du trafic dans les réseaux VANET, rendant le routage plus intelligent et réactif aux conditions changeantes de la route.
   - **Routage adaptatif et intelligent :** L'IA et le Machine Learning permettent d'optimiser les protocoles de routage en analysant en temps réel les conditions de trafic et en ajustant les itinéraires pour minimiser la latence.
   - **Routage basé sur la localisation :** L'intégration de technologies telles que le GPS haute précision et la communication V2X (Vehicle-to-Everything)

améliore l'efficacité du routage, en évitant les congestions et les zones à haut risque d'accident.

2. **Optimisation de la QoS :** Les avancées technologiques favorisent l'amélioration de la QoS, en particulier dans les environnements nécessitant des communications critiques, comme celles liées à la sécurité.
   - **Réduction de la latence :** L'adoption des réseaux 5G et futurs 6G améliore la connectivité avec une latence ultra-faible, ce qui est crucial pour les applications de sécurité comme l'évitement de collision.
   - **Gestion adaptative des ressources :** Edge Computing permet un traitement des données plus proche des véhicules, réduisant le temps de transmission et améliorant la QoS pour les services critiques.
3. **Sécurité et Fiabilité du Réseau :** La sécurité et la fiabilité des réseaux VANET sont cruciales, surtout avec l'augmentation des menaces de cybersécurité et des risques de pannes.
   - **Protection contre les cyberattaques :** Les nouvelles technologies de cryptographie et l'authentification basée sur la blockchain renforcent la sécurité des communications dans les VANET.
   - **Correction d'erreurs et tolérance aux pannes :** L'IA permet de détecter et corriger dynamiquement les erreurs de transmission, assurant ainsi une meilleure fiabilité du réseau.
4. **Standardisation et Interopérabilité :** Avec l'intégration de diverses technologies et infrastructures, il est crucial de garantir que les différents systèmes et appareils puissent interagir de manière fluide et efficace.
   - L'intégration des VANET avec les infrastructures existantes nécessite des efforts de standardisation pour assurer une interopérabilité fluide entre différentes technologies et fabricants.

2.6 Intelligence Artificielle pour l'Optimisation dans les Réseaux VANET

L'intelligence artificielle (IA) a révolutionné les systèmes de communication en permettant d'optimiser le routage et la gestion des réseaux VANET. En appliquant des techniques avancées d'IA, il devient possible de réduire la latence, améliorer la QoS et garantir une transmission rapide des données entre les véhicules.

2.6.1 Introduction aux Techniques d'IA Utilisées

L'intelligence artificielle (IA) utilisée pour l'optimisation dans les réseaux VANET permet d'améliorer significativement les services de transport en intégrant des capacités de traitement local et en temps réel. Grâce à l'IA, les réseaux VANET peuvent collecter et analyser des données en temps réel issues des véhicules, telles que les habitudes de conduite, les conditions de circulation ou les incidents, pour optimiser les services proposés. Par exemple, l'IA peut améliorer la gestion du trafic, anticiper les accidents ou optimiser les itinéraires en fonction des conditions locales. En intégrant l'IA aux points d'accès périphériques (Edge Nodes), il est possible de développer des modèles d'apprentissage spécifiques à chaque région, réduisant ainsi la latence et améliorant la précision des prédictions en temps réel. Cette approche optimise également l'efficacité du réseau en réduisant la dépendance à un serveur central pour le traitement des données, un facteur crucial dans les environnements à haut débit de données comme les VANET. De plus, ces techniques d'optimisation s'appuient sur des algorithmes d'apprentissage automatique et des systèmes distribués, garantissant que les décisions prises par l'IA soient basées sur des données locales pertinentes, tout en réduisant les temps de réponse et en assurant la sécurité des informations transmises [20].

Comme illustré dans la Figure 2.2 ci-dessous, l'architecture VANET basée sur l'IA met en œuvre une authentification mutuelle et une collecte de données via des nœuds périphériques (Edge Nodes), assurant la sécurité et l'efficacité du réseau [20].

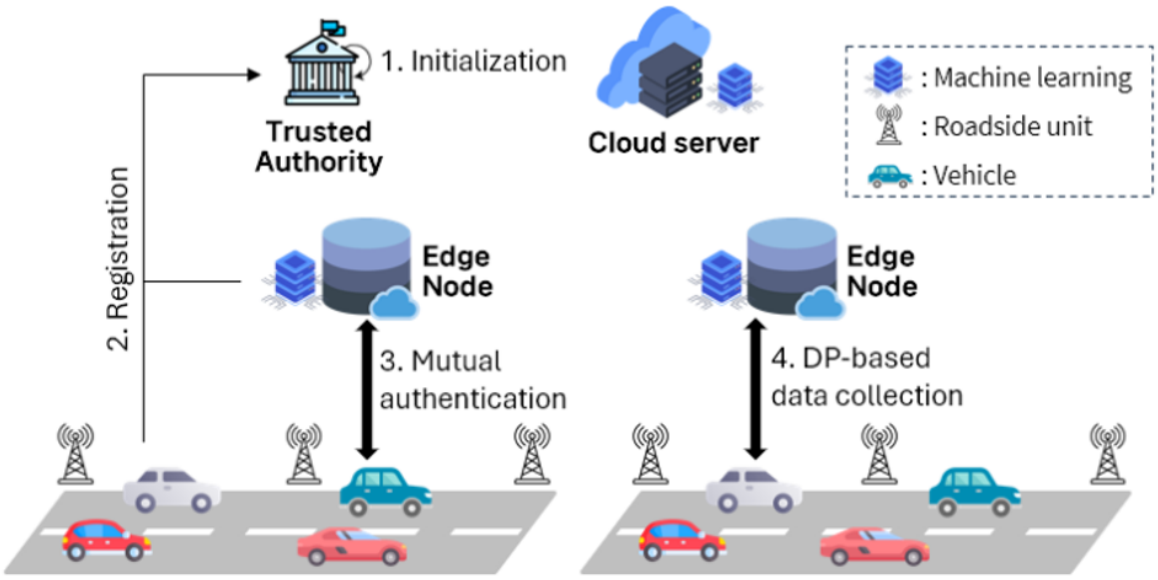


FIGURE 2.2 – Architecture d'un réseau VANET basé sur l'IA.

2.6.2 Techniques d'Optimisation pour le Routage dans les VANET

Dans les réseaux de véhicules intelligents VANET, le routage des données représente un défi de taille en raison des conditions variables et dynamiques des environnements urbains. Assurer la fiabilité et la rapidité de la transmission de l'information entre les véhicules est essentiel pour une gestion efficace des ressources de communication. Avec l'essor des techniques d'IA, il est désormais possible d'optimiser ces réseaux en prenant en compte divers facteurs, comme les conditions de trafic en temps réel, la mobilité des véhicules et les fluctuations de la qualité des signaux. Ces avancées permettent de développer des approches adaptatives qui améliorent la performance du routage et la QoS pour l'ensemble des utilisateurs.

2.6.2.1 Apprentissage Supervisé

L'apprentissage supervisé est une technique clé pour optimiser le routage dans les réseaux VANET. Il repose sur l'utilisation de données étiquetées, où les caractéristiques des environnements urbains, tels que les obstacles physiques comme les bâtiments, sont associées à des résultats connus, tels que le taux d'erreur binaire (BER) lors de la transmission des signaux. En entraînant un modèle à comprendre ces relations, il devient possible de prédire les zones de forte dégradation du signal et de choisir

les routes de communication optimales en fonction des conditions rencontrées. Cela améliore significativement la fiabilité et l'efficacité des communications, en particulier dans des scénarios critiques où des décisions rapides et précises sont nécessaires, comme pour les véhicules d'urgence [21].

2.6.2.2 Apprentissage Non Supervisé

L'apprentissage non supervisé est un paradigme de l'IA qui permet d'identifier des motifs, des structures, ou des relations au sein de données non étiquetées. Dans le cadre des réseaux VANET, l'apprentissage non supervisé est utilisé pour optimiser des aspects tels que le routage des données et la gestion des ressources, sans nécessiter de données préalablement étiquetées. L'un des principaux algorithmes utilisés dans ce domaine est le K-means clustering, qui regroupe les véhicules en fonction de caractéristiques telles que leur proximité géographique et la qualité des signaux radio (SNR). Cette méthode permet de minimiser l'utilisation des ressources tout en optimisant l'efficacité des communications entre véhicules. L'apprentissage non supervisé dans les réseaux VANET contribue également à réduire la latence et à améliorer la répartition des ressources, en optimisant le comportement des véhicules dans des conditions dynamiques [22].

2.6.2.3 Algorithmes Génétique

Les algorithmes génétiques (AG) jouent un rôle crucial dans l'optimisation des protocoles de routage pour les réseaux VANET. Ils s'inspirent des principes de la sélection naturelle et de l'évolution pour résoudre des problèmes complexes en explorant un large espace de solutions potentielles. Dans le contexte de la recherche présentée, l'AG est combiné avec l'optimisation par colonie de fourmis (ACO) pour former une approche appelée EGAACO(Enhanced Genetic Algorithm and Ant Colony Optimization). Cette méthode permet d'améliorer l'efficacité du routage en ajustant dynamiquement les chemins de communication entre les véhicules selon les conditions de trafic. L'AG intervient principalement dans les processus de sélection, de croisement et de mutation des solutions, permettant de trouver des trajectoires optimisées pour le transfert des données. Les résultats montrent que cette approche dépasse les méthodes traditionnelles en termes de débit moyen, de ratio de livraison des paquets, et de réduction du délai de bout en bout, ce qui la rend particulièrement adaptée aux

environnements de trafic urbain complexe [23].

2.6.2.4 Apprentissage Profond (Réseaux de Neurones)

Dans le contexte des VANET, les réseaux de neurones profonds, tels que les réseaux neuronaux convolutifs (CNN) et récurrents (RNN), jouent un rôle crucial pour l'analyse des données de trafic en temps réel. Les CNN, principalement utilisés pour la reconnaissance d'image, permettent de traiter les cartes de trafic et de détecter les obstacles ou anomalies de circulation. Les RNN, de leur côté, sont spécialisés dans le traitement des séquences temporelles et permettent de prévoir l'évolution des conditions de trafic en fonction des tendances actuelles et historiques. Cela est particulièrement utile pour identifier les itinéraires les plus fluides et anticiper les ralentissements à l'avance. En appliquant l'apprentissage profond, les VANET peuvent gérer une grande quantité de données en continu, ajuster les trajets selon des analyses précises, et réagir rapidement aux changements dans l'environnement urbain, comme les embouteillages soudains, améliorant ainsi l'efficacité globale du réseau [24].

2.6.2.5 Apprentissage par Renforcement

L'apprentissage par renforcement est une méthode de plus en plus prisée dans les réseaux VANET pour son efficacité dans la gestion dynamique des routes. Dans ce cadre, les agents d'apprentissage (c'est-à-dire les véhicules ou unités de contrôle) apprennent à choisir les meilleures actions (itinéraires) pour maximiser une récompense cumulative, telle qu'une faible latence de communication ou un taux élevé de livraison des paquets. Par exemple, des algorithmes de type Q-learning permettent aux agents d'optimiser le routage en évaluant chaque action en fonction des récompenses reçues, telles que la fluidité du trafic ou la minimisation des temps de parcours. Une variante plus avancée, l'apprentissage par renforcement profond (DRL), combine cette approche avec des réseaux de neurones profonds pour traiter des environnements encore plus complexes. Les DRL dans les VANET améliorent significativement la réactivité et l'adaptabilité du réseau en permettant une prise de décision plus intelligente, même dans des situations de trafic changeant rapidement, contribuant ainsi à une réduction de la congestion et à l'amélioration de la QoS pour l'ensemble du réseau [24].

2.6.3 Avantages de l'IA pour l'Amélioration du Routage et de la QoS dans les Réseaux VANET

L'intégration des techniques d'IA dans les protocoles de routage et les stratégies de gestion de QoS offre plusieurs avantages [17] :

— **Amélioration du routage grâce à l'IA :** L'IA et l'apprentissage automatique (ML) permettent d'optimiser la sélection des routes dans les réseaux VANET en s'adaptant dynamiquement aux changements de topologie et à la densité variable du trafic. Contrairement aux protocoles classiques, ces approches basées sur l'IA analysent en temps réel les conditions du réseau et ajustent les itinéraires en conséquence. Par exemple, des algorithmes de renforcement comme le Q-learning permettent aux véhicules de choisir les routes les plus optimales en apprenant des décisions passées. De plus, les techniques de vision par ordinateur et les réseaux neuronaux peuvent être utilisées pour détecter les obstacles urbains et améliorer les performances du routage géographique en sélectionnant les relais les plus fiables.

— **Optimisation de la QoS :** L'IA joue un rôle clé dans l'amélioration des critères de QoS en réduisant la latence, en augmentant le débit et en diminuant le taux de perte de paquets. Grâce aux modèles prédictifs, les protocoles intelligents peuvent anticiper les ruptures de connexion et ajuster les itinéraires avant que des dégradations ne surviennent. L'apprentissage par renforcement permet de minimiser le délai de transmission en sélectionnant des routes offrant un bon compromis entre fiabilité et rapidité. De plus, les algorithmes basés sur l'IA optimisent la gestion de la bande passante en répartissant dynamiquement la charge de transmission en fonction des besoins du réseau.

— **Gestion avancée du trafic et des ressources réseau :** L'IA permet d'optimiser la gestion des ressources réseau en améliorant l'allocation du spectre radio et en optimisant la gestion des handovers, c'est-à-dire le changement de point d'accès des véhicules en mouvement. Par exemple, les réseaux neuronaux convolutifs (CNNs) permettent une meilleure attribution des fréquences pour réduire les interférences. De plus, les modèles de prévision du trafic, basés sur l'apprentissage supervisé, analysent les tendances et proposent des itinéraires alternatifs

en cas de congestion. Cette gestion intelligente contribue à une meilleure fluidité du trafic et une réduction des embouteillages.

— **Sécurité et protection contre les attaques :** Les réseaux VANET sont exposés à diverses cyberattaques telles que les attaques Sybil, les attaques par déni de service (DDoS) ou la falsification de messages. L'IA est un outil puissant pour détecter ces menaces et améliorer la sécurité du routage. Grâce aux algorithmes de détection d'anomalies comme le SVM (Support Vector Machine) ou le deep learning, il est possible d'identifier les comportements suspects et de bloquer les véhicules malveillants. De plus, des protocoles basés sur l'IA permettent d'authentifier les communications et de chiffrer dynamiquement les échanges afin de garantir la confidentialité des données transmises.

2.7 Conclusion

L'étude des approches de routage et de gestion de la QoS dans les réseaux VANET met en évidence l'évolution des techniques utilisées pour améliorer la fiabilité et l'efficacité des communications. Les protocoles non-apprenants, bien qu'efficaces dans certaines conditions, présentent des limites face aux défis liés à la dynamique des VANET. L'intégration de l'intelligence artificielle et de l'apprentissage automatique apparaît ainsi comme une solution prometteuse, permettant une prise de décision optimisée, une meilleure adaptation aux conditions du réseau et une réduction des délais de transmission. Toutefois, malgré leurs nombreux avantages, les approches basées sur l'IA posent encore des défis en termes de complexité computationnelle et de sécurité. Ces constats soulignent la nécessité de poursuivre la recherche afin de développer des solutions hybrides et intelligentes capables d'améliorer durablement la performance des VANET. La conception et la description détaillée de cet algorithme seront discutées dans le prochain chapitre de notre mémoire.

Chapitre 3

Protocole de Routage HRLB-SVDN

3.1 Introduction

L'émergence des réseaux définis par logiciel (SDN) a ouvert de nouvelles perspectives pour surmonter les défis des réseaux VANET. En offrant une gestion centralisée et une visibilité globale du réseau, les SDN permettent une prise de décision plus efficace en matière de routage, réduisant ainsi les problèmes de connectivité intermittente et de congestion du réseau. L'intégration du SDN dans les VANET, connue sous le nom de Software-Defined Vehicular Network (SDVN), améliore la flexibilité et l'efficacité du routage en optimisant la gestion du trafic et l'utilisation des ressources réseau.

Dans ce contexte, le protocole "A Hierarchical Routing Scheme With Load Balancing in Software Defined Vehicular Ad Hoc Networks" [12] propose une approche de routage hiérarchique basée sur la géographie et l'équilibrage de charge pour les SDVN. Il divise la zone de couverture en plusieurs petites grilles géographiques et sélectionne les chemins optimaux en fonction de la densité des véhicules et des probabilités historiques de transfert entre les grilles. Une fonction de coût est introduite pour équilibrer la charge et maintenir deux chemins optimaux avec un coût minimal. Ensuite, des nœuds relais sont sélectionnés sur ces chemins en fonction de leur utilité pour le routage.

Ce chapitre analyse en détail ce protocole, en mettant en évidence ses mécanismes de sélection des chemins, sa gestion de l'équilibrage de charge et son impact sur la performance globale des SDVN. De plus, il examine ses limitations et ses points faibles, notamment en ce qui concerne la complexité de mise en œuvre, la dépendance à l'infrastructure SDN et les éventuelles latences induites par la gestion centralisée du réseau.

3.2 Concepts fondamentaux

1. **L'architecture SDN (Software-Defined Networking)**

   L'architecture SDN est un modèle de gestion des réseaux où le contrôle du réseau est centralisé dans un contrôleur SDN au lieu d'être réparti entre les équipements réseau comme les routeurs et les commutateurs. Cette séparation permet une programmabilité accrue, une meilleure flexibilité et une gestion plus efficace du réseau. Dans les réseaux VANET, la forte mobilité des véhicules entraîne des défis en matière de routage et de gestion du trafic. L'intégration du SDN dans les VANET améliore considérablement la gestion du réseau grâce aux avantages suivants [25] :

   — **Vision globale du réseau :** Contrairement aux protocoles traditionnels où chaque véhicule prend des décisions de routage localement, le contrôleur SDN dispose d'une vue centralisée du réseau, ce qui permet une prise de décision optimisée.

   — **Meilleure gestion de la charge du réseau :** Le SDN permet un équilibrage de charge intelligent, réduisant la congestion en sélectionnant dynamiquement les meilleures routes. Il surveille en permanence l'état du réseau et adapte le routage en temps réel.

   — **Flexibilité et adaptation rapide :** Grâce à sa capacité de programmabilité, le SDN peut adapter dynamiquement les routes en fonction des mouvements des véhicules, contrairement aux protocoles classiques qui peinent à suivre ces changements rapides.

   — **Séparation du Plan de Contrôle et du Plan de Données**

      — **Plan de contrôle (Control Plane) :** Assure la gestion et la prise de décisions, localisé dans le contrôleur SDN.

      — **Plan de données (Data Plane) :** Responsable du transfert des paquets entre véhicules et infrastructures, exécuté par des équipements réseau intelligents.

2. **LTE (Long Term Evolution)**

   Le LTE (est une technologie de communication sans fil utilisée pour les réseaux mobiles 4G. Elle permet des connexions Internet rapides, avec un débit élevé et

une faible latence, ce qui signifie que les données sont envoyées et reçues très rapidement. Dans les réseaux VANET, LTE est essentiel pour permettre aux véhicules de communiquer entre eux (V2V) ou avec les infrastructures comme les feux de signalisation (V2I). Grâce à LTE, les véhicules peuvent échanger des informations en temps réel, comme les alertes de sécurité ou les conditions de circulation, même sur de longues distances [26].

3. **Protocole OpenFlow**

   OpenFlow est un protocole de communication qui permet la séparation du plan de contrôle et du plan de données dans un réseau, facilitant ainsi la gestion centralisée des commutateurs (switches) par un contrôleur SDN. Dans un environnement VANET, il joue un rôle clé en optimisant la communication entre les véhicules et l'infrastructure en permettant un routage intelligent des paquets. Contrairement aux protocoles traditionnels comme AODV, qui découvrent les routes de manière distribuée, OpenFlow permet au contrôleur SDN de définir les règles de transmission en fonction des adresses MAC et d'autres paramètres réseau. Cette approche réduit la surcharge du réseau, améliore la latence et garantit une meilleure gestion des flux de données, notamment pour les applications critiques comme la communication entre les leaders de pelotons de véhicules [27].

4. **routage hiérarchique**

   Le routage hiérarchique est une méthode de routage qui divise un réseau en plusieurs niveaux ou partitions afin d'améliorer l'efficacité du routage et la gestion des ressources. Dans l'article, cette approche est utilisée dans le cadre du Hierarchical Privacy-Preserving Route Planning (HPRoP), où le réseau routier est divisé en sous-réseaux appelés partitions, organisées de manière hiérarchique. Chaque partition contient des routes pré-calculées et est gérée indépendamment, ce qui réduit le temps de calcul et améliore la protection de la vie privée des utilisateurs. Cette organisation hiérarchique permet d'optimiser le processus de planification d'itinéraires tout en minimisant la surcharge de calcul et de communication [28].

3.3 Modèle de réseau et hypothèses du protocole

Dans cette section, nous présentons le modèle de réseau et les hypothèses sur lesquels repose le protocole HRLB-SVDN. Cette démarche vise à établir un cadre clair pour la conception et l'analyse de notre approche de routage. Le réseau est conçu selon une approche en grilles, permettant une meilleure gestion de la mobilité et de la densité. Ces hypothèses permettent de simplifier la prise de décision au niveau du contrôleur SDN.

3.3.1 Modèle de Réseau

Le protocole HRLB-SVDN est un protocole de routage hiérarchique pour les réseaux VANET définis par logiciel (SDVN). Il utilise une architecture centralisée où la zone urbaine est divisée en petites grilles géographiques. Le protocole choisit les chemins optimaux en se basant sur la densité des véhicules, la probabilité de transfert entre les grilles et la charge réseau. Grâce à cette approche, HRLB-SVDN améliore le taux de livraison des paquets, réduit la congestion et équilibre la charge réseau.

Considérons une architecture SDVN (Software-Defined Vehicular Network) illustrée dans la Figure 3.1, où le réseau est divisé à :

1. **Plan de contrôle (Contrôleur SDN) :**Gère le routage en surveillant la topologie du réseau en temps réel.
2. **Plan de données (Véhicules et Stations de Base) :** Transmet les paquets selon les instructions du contrôleur SDN.
3. **Modes de communication :** on a deux types :
   - **V2V (Véhicule-à-Véhicule) :**Les véhicules échangent directement des paquets et servent de relais.
   - **V2BS (Véhicule-à-Station de Base) :**V2BS (Véhicule-à-Station de Base) Les véhicules transmettent des données au contrôleur SDN via des stations de base LTE.
4. **Station de base :** On a utilisé :
   - **Stations de base macrocellulaires (MBSs) :** Couvrent de grandes zones et assurent une connectivité stable.

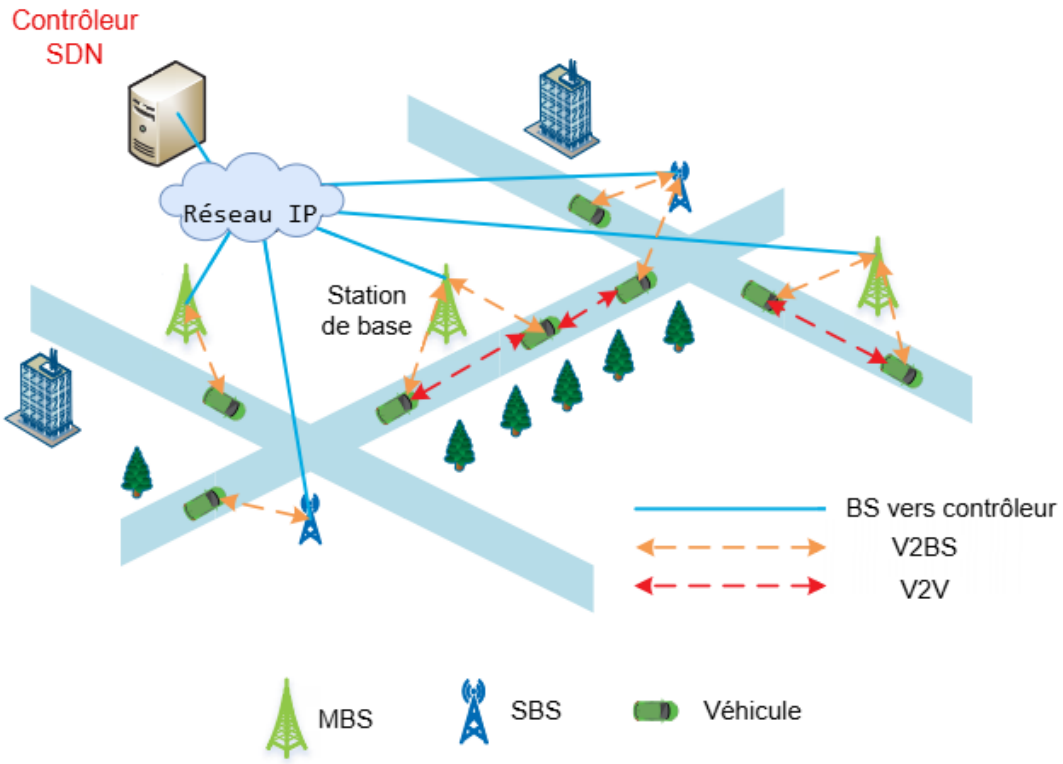


FIGURE 3.1 – Architecture d'un SDN-VANET.

— **Stations de base petites cellules (SBSs) :** Optimisent la couverture en zones urbaines denses, améliorant la communication à courte portée.

Tous les véhicules envoient périodiquement des messages de balise au contrôleur central via les stations de base. Ces messages permettent au contrôleur de surveiller en temps réel l'état du réseau et d'optimiser le routage. Chaque message contient les informations suivantes :

— **Position :** Coordonnées géographiques actuelles du véhicule.

— **Vitesse :** Vitesse actuelle du véhicule en déplacement.

— **Direction :** Orientation du véhicule sur la route.

— **Taille du tampon restant :** Espace mémoire disponible pour le stockage temporaire des paquets.

Le contrôleur central divise une grande région de la ville en grilles (Figure 3.2). Chaque grille est définie comme suit :

— **Grille Source :** La grille où se trouve le véhicule source.

— **Grille Destination :** La grille où se trouve le véhicule de destination.

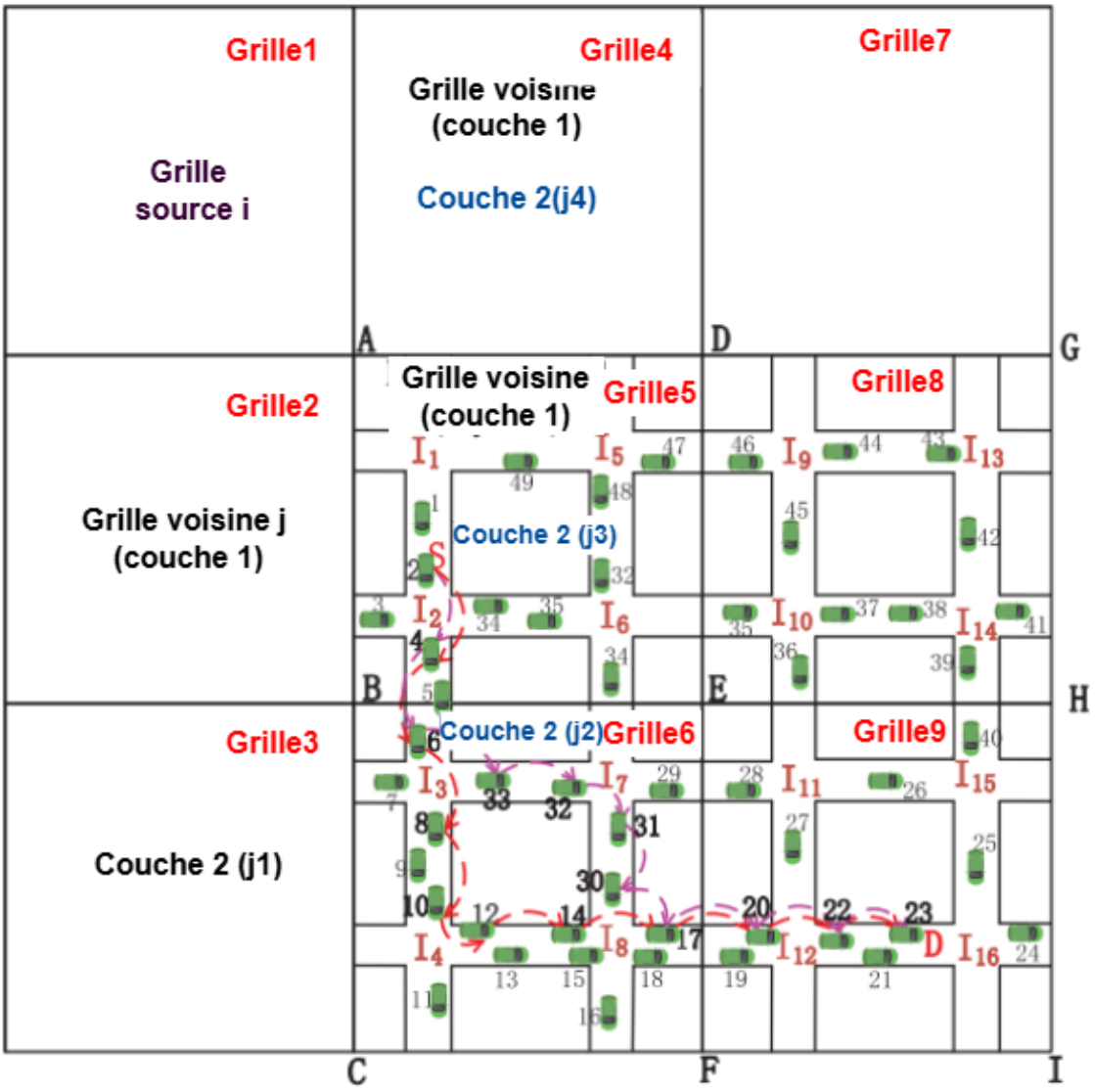


FIGURE 3.2 – Division du réseau en plusieurs grilles.

— **Densité des Véhicules d'une Grille :** Nombre moyen de véhicules dans une grille donnée, calculé par l'Équation 3.1 :

$$\rho_j = \frac{V_j}{S_j} \tag{3.1}$$

où V_j est le nombre de véhicules et S_j la surface de la grille.

— **Densité moyenne des véhicules dans la grille voisine j :**

$$\rho_ESj = \frac{1}{n}\sum_{k=1}^{n}\rho_{jk} \tag{3.2}$$

où : - n est le nombre total de grilles adjacentes à la grille j, excluant la grille source i.

- ρ_{jk} est la densité des véhicules en temps réel dans la grille jk (couche 2).

— **Probabilité moyenne de transfert dans la grille voisine j :**

$$P_ESj = \frac{1}{n} \sum_{k=1}^{n} p_{i,jk}^{(2)}, \tag{3.3}$$

où : - $p_{i,jk}^{(2)}$ est la probabilité de transition d'ordre 2 de la grille source i à la grille de couche 2 jk.

— **Probabilité de Transfert entre Grilles :** La probabilité qu'un véhicule passe d'une grille i à une grille j est donnée par :

$$p_{i,jk}^{(2)} = \sum_{\varsigma \in I} p_{i,\varsigma}\, p_{\varsigma,jk} \tag{3.4}$$

où I représente l'ensemble des grilles.

3.3.2 Hypothèses

Dans le cadre de notre modèle, nous formulons les hypothèses suivantes afin de simplifier l'analyse et de structurer le fonctionnement du système :

— **Tous les véhicules sont équipés d'un GPS** pour connaître leur position et leur vitesse.

— **Le contrôleur SDN dispose d'une carte numérique** pour suivre la topologie des routes.

— **Le contrôleur SDN reçoit des messages de balise de chaque véhicule** (position, vitesse, charge du tampon).

— **Chaque véhicule est connecté à une station de base LTE** pour communiquer avec le contrôleur.

— **Les décisions de routage sont prises de manière centralisée par le contrôleur SDN**.

— **Le protocole OpenFlow est utilisé** pour la communication entre le contrôleur SDN et les dispositifs du réseau.

3.4 Conception du Protocole

Dans cette section, nous décrivons l'architecture générale du protocole proposé, en détaillant ses composants clés et leurs interactions.

3.4.1 Vue d'ensemble de la conception

Le protocole HRLB-SVDN repose sur trois principes :

1. **Sélection des grilles**
   - Divise la ville en plusieurs grilles de taille égale.
   - Sélectionne les grilles avec la meilleure connectivité en fonction de la densité des véhicules et de la probabilité de transfert entre les grilles.

   Le contrôleur SDN sélectionne les grilles les plus connectées à l'aide d'une fonction d'état globale (OS) :

$$OS_j = \mu_1 \cdot \frac{\rho_{\mathrm{ES}j}}{\rho_{\max}} + \mu_2 \cdot \frac{\rho_j}{\rho_{\max}} + \mu_3 \cdot p_{i,j} + \mu_4 \cdot P_{\mathrm{ES}j} \tag{3.5}$$

   tel que :

   - ρ_ESj : Moyenne des densités des véhicules dans les grilles voisines.
   - P_ESj : Moyenne des probabilités de transfert entre la grille actuelle et les autres grilles voisines.

   Le protocole HRLB-SVDN repose sur une sélection dynamique des grilles afin d'assurer un routage optimal des paquets dans un réseau SDVN. L'organigramme de la Figure 3.3 détaille les différentes étapes du processus.

2. **Sélection des chemins**
   - Sélectionne deux chemins optimaux en minimisant un coût de routage basé sur :
     - La longueur du chemin
     - La densité des véhicules sur le chemin
     - La distance entre véhicules adjacents
     - La charge du réseau

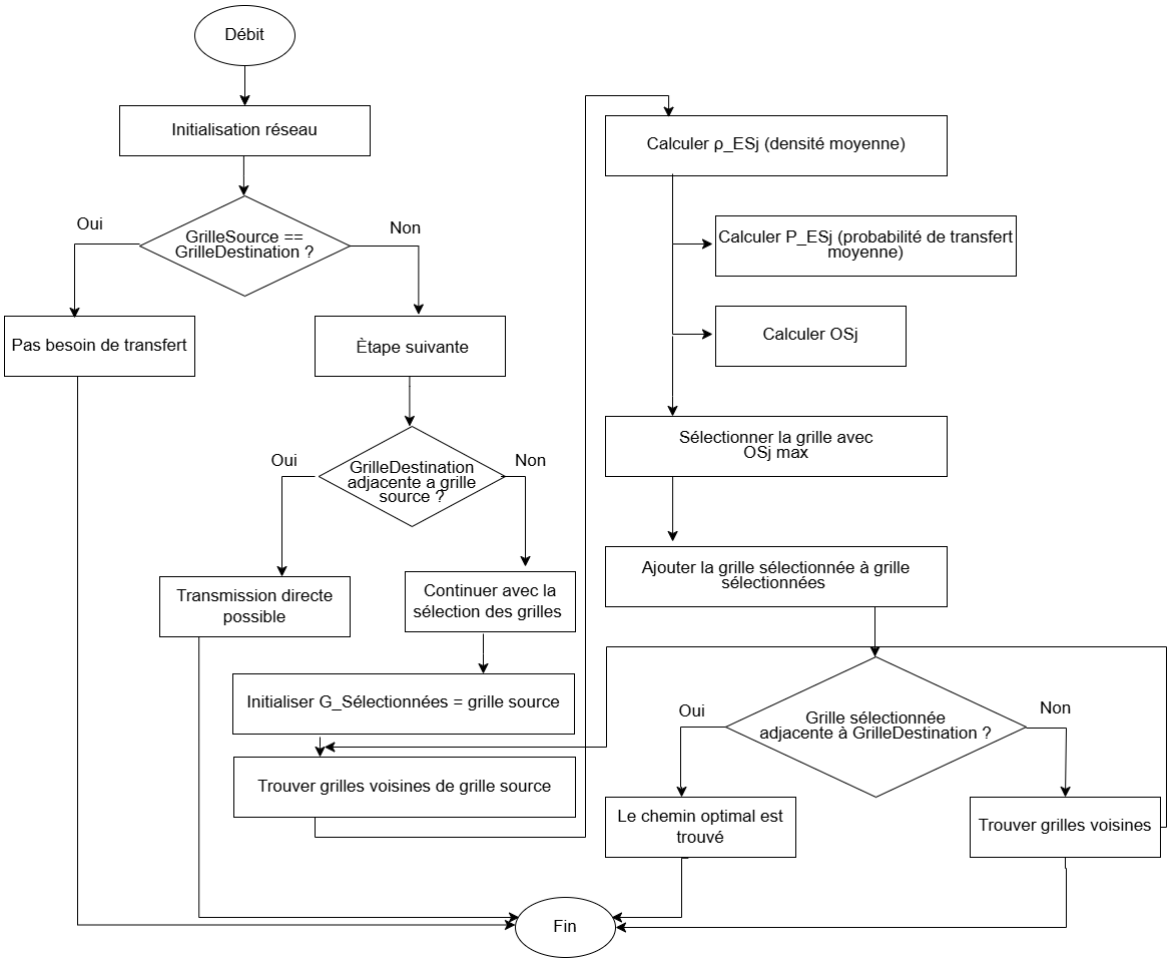


FIGURE 3.3 – Organigramme de la sélection des grilles dans HRLB-SVDN.

Le contrôleur SDN sélectionne deux chemins en fonction d'une fonction de coût de chemin (PC) :

$$PC_i = \alpha f(l_i) + \beta \left[1 - g(v_i)\right] + \gamma h(\sigma_i) - \delta \left[1 - \phi(L^i_{\text{avg}})\right] \tag{3.6}$$

tel que :

- $f(l_i) = \frac{l_i}{l_{\max}}$: Longueur du chemin
- $g(v_i) = \frac{v_i}{v_{\max}}$: Densité des véhicules sur le chemin
- $h(\sigma_i) = \frac{\sigma_i}{\sigma_{\max}}$: Écart-type de la distance entre les véhicules (meilleure connectivité si faible)
- $\left(L^i_{\text{avg}}\right) = \frac{L^i_{\text{avg}}}{R_{tr}}$: Charge du réseau sur le chemin

  La distance entre véhicules adjacents est donnée par :

$$\sigma_i = \sqrt{\frac{(r^i_{a,b} - \mu_i)^2 + (r^i_{b,c} - \mu_i)^2 + \cdots + (r^i_{n,m} - \mu_i)^2}{N^i_{\text{inval}}}} \tag{3.7}$$

où :

$$\mu_i = \frac{r^i_{a,b} + r^i_{b,c} + \cdots + r^i_{n,m}}{N^i_{\text{inval}}} \tag{3.8}$$

est la moyenne arithmétique de la distance entre véhicules adjacents.

Le protocole HRLB-SVDN utilise une stratégie d'optimisation pour sélectionner les meilleurs chemins entre la source et la destination. La Figure 3.4 illustre les différentes étapes du processus.

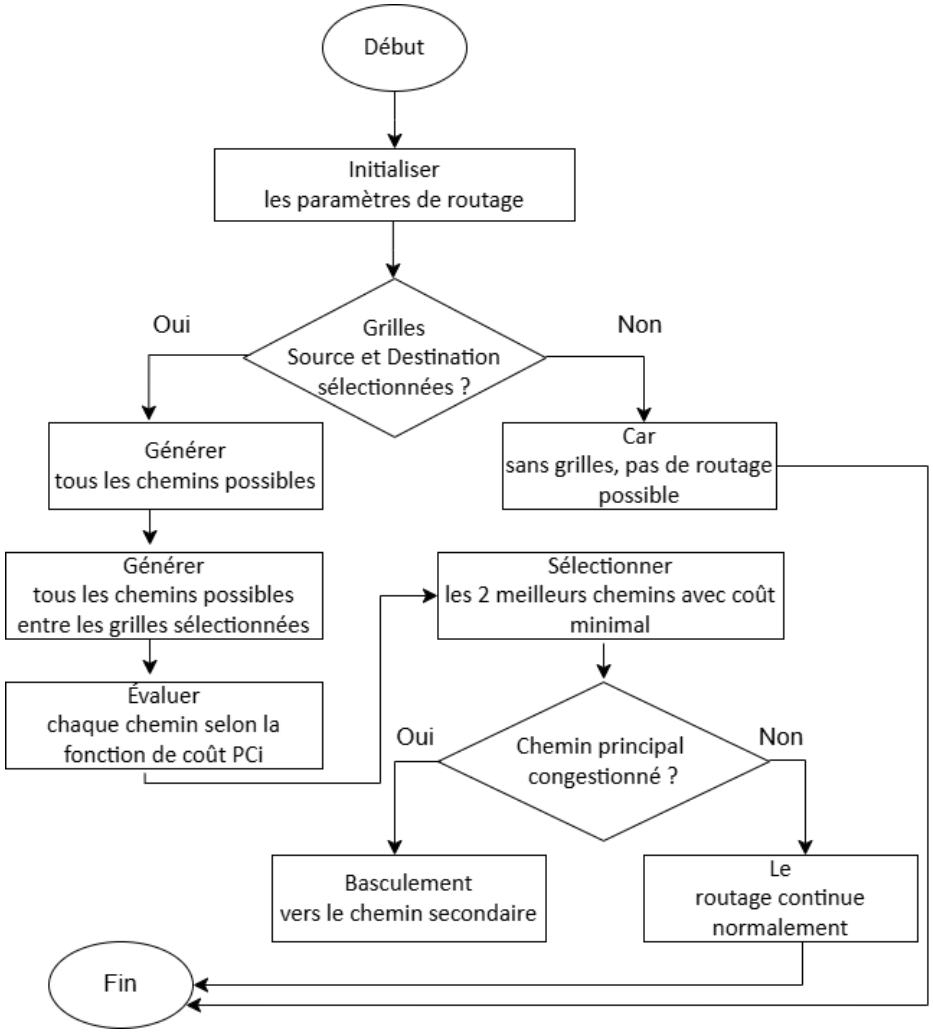


FIGURE 3.4 – Organigramme de la sélection des chemins avec équilibrage de charge.

Le protocole HRLB-SVDN utilise une stratégie d'optimisation pour sélectionner les meilleurs chemins entre la source et la destination. La Figure 3.5 illustre un exemple de représentation graphique de la sélection des chemins, où plusieurs

routes alternatives sont considérées en fonction des critères définis.

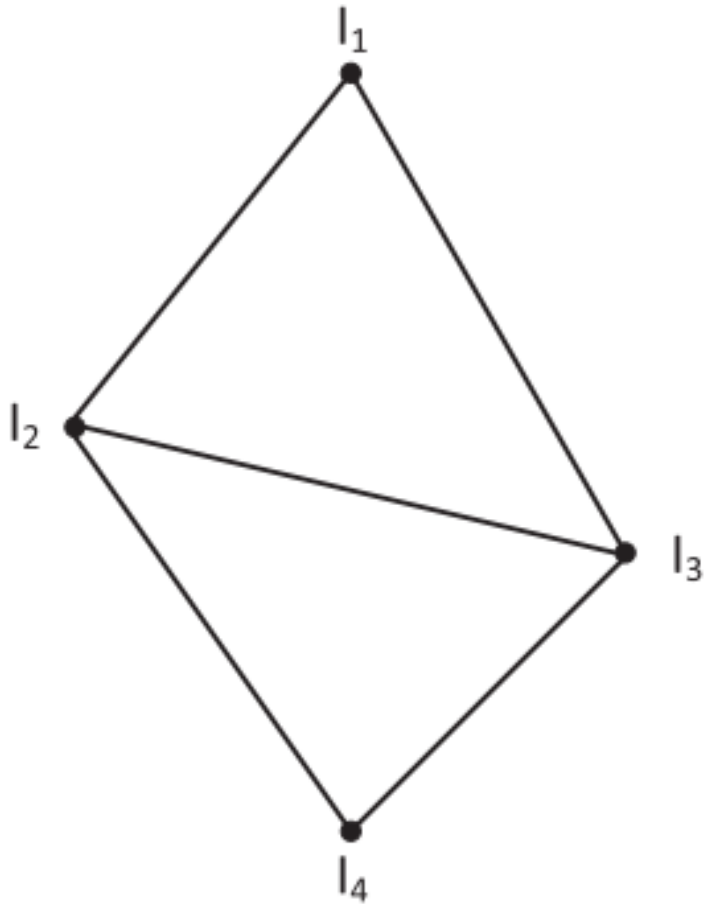


FIGURE 3.5 – Exemple de graphe de chemin utilisé pour le routage.

3. **Sélection des nœuds relais**
   - Filtre les nœuds relais selon un score d'utilité qui dépend :
     - De l'espace tampon restant
     - De la distance à la destination

Chaque véhicule intermédiaire est sélectionné en fonction d'une fonction d'utilité (U_i) :

$$U_i = \frac{B_i}{S_i} \tag{3.9}$$

tel que :

- B_i : Espace tampon restant du véhicule
- S_i : Distance du véhicule à la destination

Après avoir sélectionné le meilleur chemin, le protocole HRLB-SVDN identifie les véhicules les plus appropriés pour relayer les paquets. La Figure 3.6 détaille ce processus.

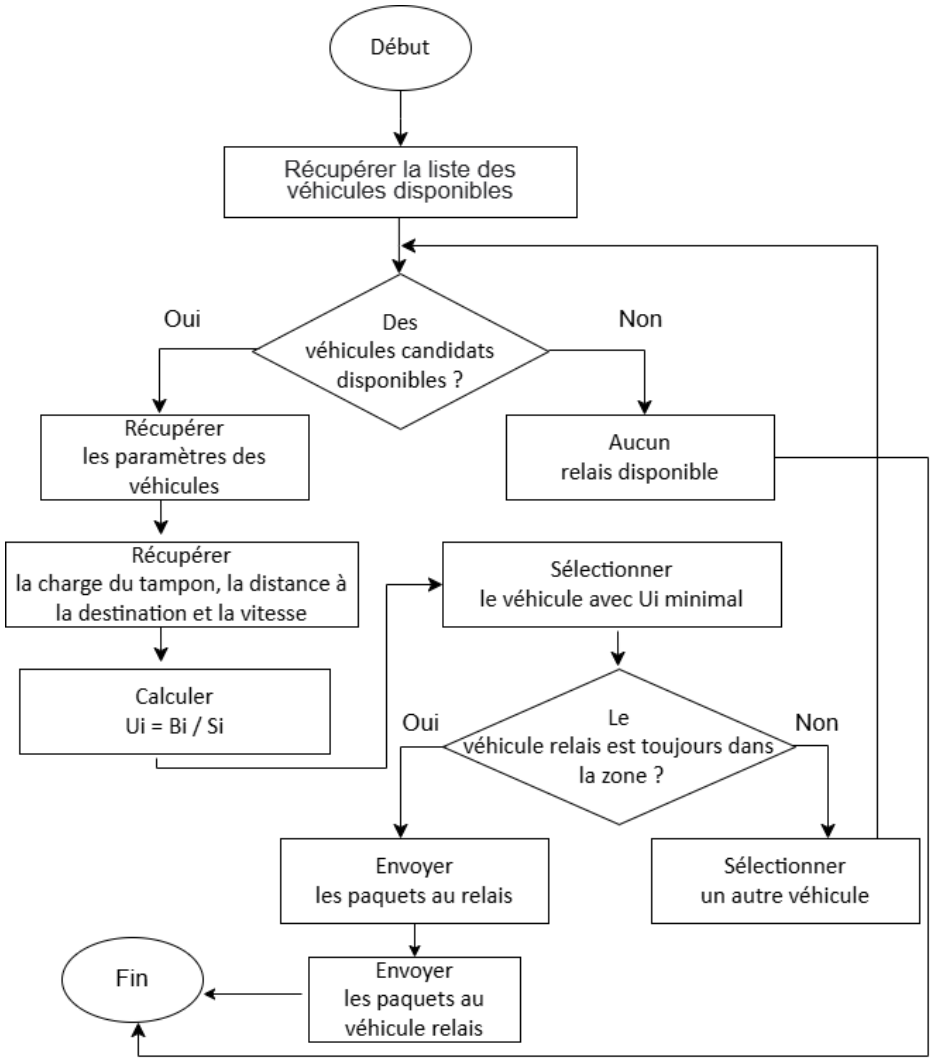


FIGURE 3.6 – Organigramme de la sélection des nœuds relais.

La Figure 3.7 illustre le processus de routage hiérarchique en trois étapes. Tout d'abord, le contrôleur SDN sélectionne les grilles optimales en fonction de la connectivité des véhicules. Ensuite, deux chemins optimaux sont choisis en minimisant un coût basé sur la distance, la densité des véhicules et la charge du réseau. Enfin, les nœuds relais sont sélectionnés selon leur espace tampon disponible et leur proximité avec la destination. Ce processus permet d'améliorer l'efficacité du routage et de réduire la congestion dans un réseau VANET.

La Figure 3.8 illustre ce processus. Si un véhicule source A ne possède aucun voisin à portée, le contrôleur SDN sélectionne le véhicule B comme relais. Ce processus est

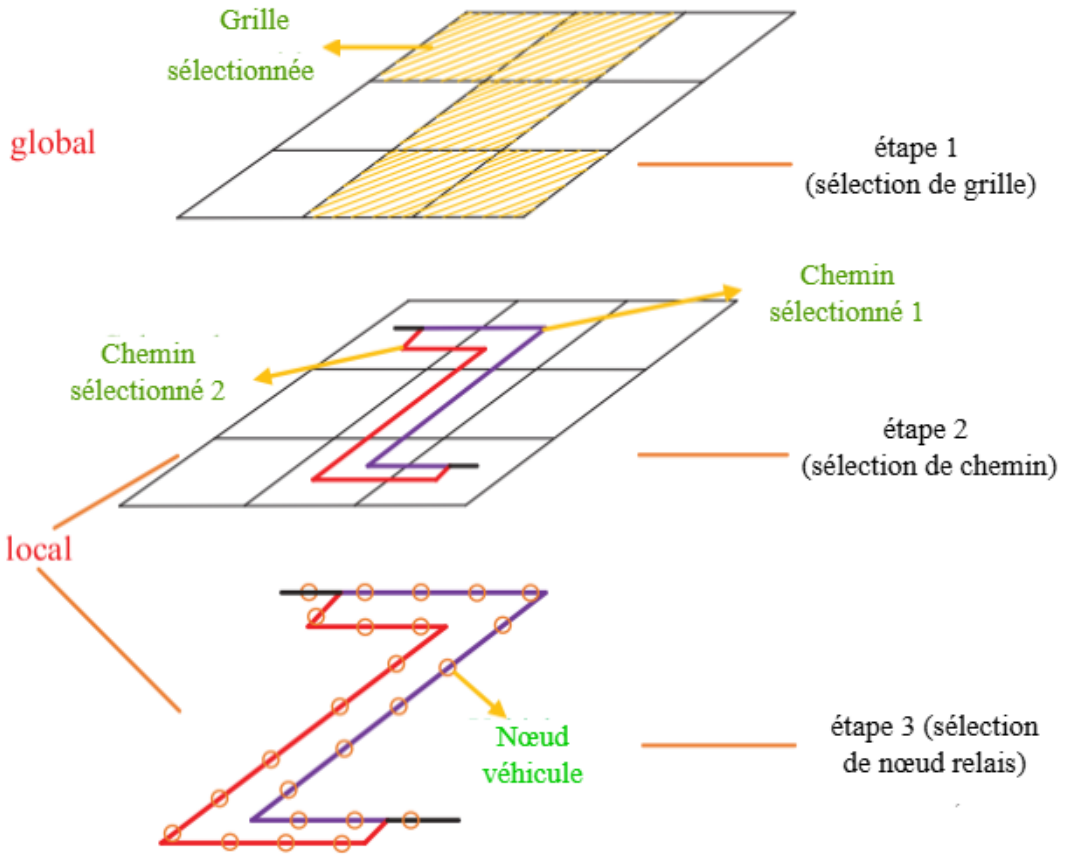


FIGURE 3.7 – Processus de sélection des grilles, chemins et nœuds relais.

répété jusqu'à ce qu'un véhicule atteigne la destination.

3.5 Exemple de déploiement du Protocole HRLB-SVDN

Pour illustrer le fonctionnement du protocole HRLB-SVDN, considérons un scénario simulé dans une zone urbaine dont les dimensions sont de 3000 mètres sur 3000 mètres. Cette zone est subdivisée en 36 grilles carrées, chacune mesurant 500 mètres de côté. On suppose que 540 véhicules sont répartis aléatoirement dans ces grilles, soit une moyenne de 15 véhicules par grille. Nous considérons quatre grilles représentées dans la carte numérique centralisée (ACIG) :

— Grille 5 : ABED

— Grille 6 : BCFE

— Grille 9 : EFIH

— Grille 8 : DEHG

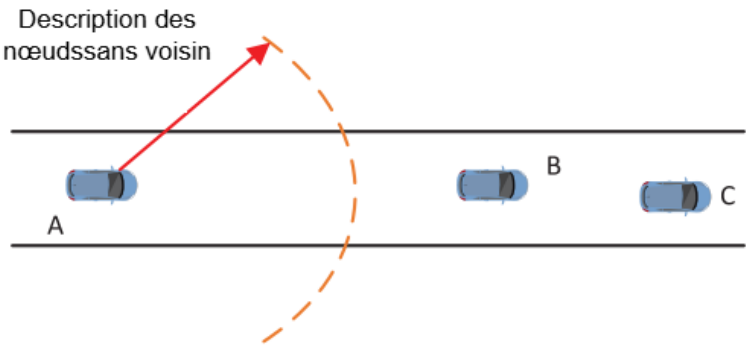


FIGURE 3.8 – Description des nœuds sans voisins dans HRLB.

L'objectif est de transmettre les paquets du nœud source 2 (situé dans la Grille 5) vers le nœud destination 23 (situé dans la Grille 9).

3.5.1 Sélection des Grilles

Le nœud 2 commence par rechercher une table de routage existante. En l'absence de celle-ci, une *requête de routage* est envoyée au contrôleur SDN.

Le contrôleur identifie que :

— Le nœud source 2 appartient à la grille 5

— Le nœud de destination 23 appartient à la grille 9

Ces deux grilles n'étant pas adjacentes, le processus de sélection de grilles s'active conformément à l'équation d'état global (3.5).

Les grilles 6 et 8 étant voisines de la grille 5, le contrôleur calcule leur score de connectivité. Supposons que la valeur de la grille 6 soit supérieure à celle de la grille 8, alors la grille 6 est sélectionnée. La grille 9 étant adjacente à la grille 6, la sélection de grilles se termine ici.

3.5.2 Sélection des Chemins de Routage

Le contrôleur SDN identifie ensuite deux chemins potentiels entre la source et la destination dans la région formée par les grilles 5, 6 et 9. Le coût de chaque chemin est évalué à l'aide de la fonction de coût (3.6).

Les chemins sélectionnés sont :

1. Chemin 1 : I2 → I3 → I4 → I8 → I12
2. Chemin 2 : I2 → I3 → I7 → I8 → I12

Le **chemin 1** est préféré car son coût est plus faible.

3.5.3 Sélection des Nœuds Relais

Les nœuds relais sont choisis grâce à la fonction d'utilité (3.9), permettant d'optimiser le routage.

Nœuds sélectionnés :

— Chemin 1 : 4, 6, 8, 10, 12, 14, 17, 20, 22

— Chemin 2 : 4, 6, 33, 32, 31, 30, 17, 20, 22

Deux tables de routage sont générées :

— NO :1 → pour le chemin 1

— NO :2 → pour le chemin 2

3.5.4 Surveillance de la Charge et Commutation de Chemin

Durant la phase de transmission des données, le contrôleur SDN assure une surveillance continue et en temps réel de la charge du réseau, notamment sur le chemin primaire (chemin 1), afin de garantir la stabilité et la fluidité du trafic.

— Dès que le taux d'occupation du chemin 1 dépasse un seuil critique fixé à 70%, le contrôleur déclenche automatiquement une stratégie de commutation proactive en activant la table de routage NO :2.

— Tous les nouveaux paquets entrants sont alors redirigés vers un chemin alternatif (chemin 2) préconfiguré dans la table NO :2, assurant ainsi un rééquilibrage dynamique de la charge sans interruption du service.

— Afin d'éviter toute surcharge du réseau ou perte de performance, les paquets déjà en transit ne sont pas réacheminés. Cela minimise les délais de transmission et évite des traitements redondants ou des risques de duplication.

Cette approche repose sur une gestion intelligente du trafic pilotée par le contrôleur SDN, qui adapte le routage en fonction de l'évolution de la charge en temps réel. Le Tableau 3.1 présente une synthèse des paramètres initiaux utilisés dans l'exemple d'application du protocole HRLB-SVDN, illustrant ainsi les conditions de déclenchement de cette logique de commutation.

Paramètre	**Valeur**
Taille de la zone simulée	3000 m 3000 m (9 km^2)
Taille d'une grille	500 m 500 m
Nombre total de grilles	36 grilles
Nombre total de véhicules	540 véhicules
Nombre de véhicules par grille	15 véhicules
Rayon de communication	250 m

TABLE 3.1 – Données initiales et hypothèses du scénario simulé.

3.6 Limites du Protocole HRLB-SVDN

Bien que le protocole HRLB-SVDN apporte une contribution significative à l'optimisation du routage dans les réseaux véhiculaires, plusieurs limites subsistent, notamment en termes de robustesse, de scalabilité et de fiabilité dans des conditions extrêmes :

1. **Latence accrue en environnement à haute mobilité** : Le protocole repose sur la réception fréquente de messages de balise (beacon) pour suivre en temps réel la position, la vitesse et la charge tampon des véhicules. Toutefois, dans des environnements fortement dynamiques où les véhicules se déplacent rapidement et changent fréquemment de position, les délais de mise à jour peuvent entraîner une désynchronisation entre l'état réel du réseau et les décisions prises par le contrôleur. Cela peut conduire à des choix de chemin sous-optimaux et à une dégradation de la performance globale.
2. **Risque de surcharge du contrôleur SDN** : Le modèle centralisé implique que le contrôleur SDN collecte, traite et analyse continuellement les informations provenant de l'ensemble des véhicules connectés. Dans des scénarios à grande échelle, avec plusieurs milliers de nœuds mobiles, cette centralisation peut engendrer une surcharge computationnelle importante, limitant ainsi la réactivité du contrôleur et sa capacité à maintenir une vision globale cohérente du réseau.
3. **Congestion possible au niveau des stations de base** : Le bon fonctionnement du protocole repose également sur une connectivité fiable entre les véhicules et les stations de base (SBS et MBS). En zones urbaines densément peuplées ou lors de pics de trafic, les ressources radio peuvent devenir insuffisantes, engendrant une congestion des stations, une augmentation du taux de perte de paquets et une réduction notable du débit disponible.

4. **Dépendance à la précision du GPS** : Le protocole suppose une localisation précise des véhicules, ce qui est indispensable pour un routage efficace. Or, en milieux urbains denses (effet canyon urbain), la précision du GPS peut être altérée par des phénomènes de multi-trajets ou d'obstruction satellite, ce qui nuit à la fiabilité des données de positionnement et peut perturber les mécanismes de prise de décision du contrôleur SDN.

Ces limitations mettent en lumière les défis à relever pour le déploiement à grande échelle du protocole HRLB-SVDN et suggèrent des pistes d'amélioration futures, notamment en matière de décentralisation, de tolérance aux erreurs et d'optimisation des ressources radio.

3.7 Conclusion

Ce chapitre a présenté le protocole HRLB-SVDN, en détaillant ses mécanismes de sélection des grilles, des chemins et des nœuds relais afin d'optimiser le routage dans les réseaux VANET. L'approche proposée vise à améliorer l'efficacité globale du réseau en équilibrant dynamiquement la charge et en réduisant les phénomènes de congestion, notamment dans des environnements urbains denses.

Toutefois, plusieurs limitations ont été identifiées, telles que la latence accrue en cas de forte mobilité, la surcharge potentielle du contrôleur SDN et la dépendance à la précision du GPS. Ces contraintes soulignent la nécessité d'adopter des approches plus adaptatives et intelligentes pour renforcer la résilience du protocole.

Ainsi, afin de pallier ces limites, nous proposons une amélioration du protocole HRLB-SVDN en y intégrant des techniques d'apprentissage automatique (Machine Learning). Cette approche permettra d'anticiper les variations du trafic, de prendre des décisions de routage plus réactives, et d'alléger la charge du contrôleur via des mécanismes prédictifs. Ces travaux d'amélioration feront l'objet du chapitre suivant, qui portera également sur la simulation et l'évaluation des performances de la solution proposée à l'aide d'outils et de scénarios adaptés.

Chapitre 4

Optimisation du routage HRLB-SVDN via un modèle d'apprentissage supervisé

4.1 Introduction

Dans un contexte où les réseaux de type SDVN doivent répondre à des exigences élevées en termes de fiabilité, de rapidité de prise de décision et d'adaptabilité, l'efficacité des mécanismes de routage devient un enjeu stratégique majeur. Le protocole HRLB-SVDN, en apportant une structuration hiérarchique et une gestion optimisée de la charge, a contribué de manière significative à l'amélioration de la performance globale des réseaux VANET. Toutefois, malgré ses atouts indéniables, HRLB-SVDN présente certaines limites inhérentes à son mode de fonctionnement centralisé, notamment en matière de latence de décision, de surcharge potentielle du contrôleur, et de rigidité face aux variations rapides de topologie.

À mesure que l'environnement réseau devient plus dynamique et imprévisible, une approche plus intelligente et prédictive s'impose pour assurer une continuité de service optimale. C'est dans cette optique que ce chapitre propose l'intégration d'un modèle d'apprentissage supervisé de nouvelle génération, le **CatBoostClassifier**, directement appliqué à la problématique de sélection de chemin de routage.

Le modèle CatBoostClassifier, grâce à sa capacité à apprendre efficacement à partir de données tabulaires complexes et massives, permet d'anticiper avec précision le che-

min optimal en s'appuyant sur des caractéristiques locales du réseau. Cette approche vise à alléger la charge du contrôleur, à réduire la latence des décisions, et à doter les véhicules d'une capacité locale de prédiction proactive, renforçant ainsi l'agilité, la robustesse et la scalabilité du protocole HRLB-SVDN face aux défis posés par les VANET modernes. L'ensemble de l'expérimentation a été réalisé à l'aide du langage Python, en utilisant la plateforme Google Colab, offrant un environnement de développement flexible, collaboratif et adapté aux traitements de données à grande échelle.

4.2 Motivations et objectifs de l'intégration du modèle CatBoostClassifier

Dans la dynamique des réseaux VANET modernes, où rapidité, robustesse et adaptabilité sont devenues des impératifs, l'architecture actuelle du protocole HRLB-SVDN montre ses limites. Pour renforcer ses capacités décisionnelles tout en réduisant la dépendance au contrôleur central, nous proposons l'introduction d'un modèle d'apprentissage supervisé avancé : **CatBoostClassifier**. Cette section expose d'abord les motivations principales qui sous-tendent ce choix, puis précise les objectifs d'amélioration visés par cette intégration.

4.2.1 Motivations

Plusieurs facteurs motivent l'introduction d'un modèle d'apprentissage machine dans l'optimisation du routage pour les VANET :

1. **Réduction de la latence** : Dans HRLB-SVDN, chaque prise de décision nécessite un aller-retour au contrôleur, induisant une latence sensible surtout dans des environnements à forte mobilité. Un modèle local de prédiction permet de s'affranchir de ce délai.
2. **Allègement de la charge du contrôleur** : En confiant aux véhicules une partie de l'intelligence de décision, on réduit la surcharge computationnelle pesant sur le contrôleur central.
3. **Meilleure adaptabilité aux variations topologiques** : Les VANET sont par essence hautement dynamiques. Un modèle supervisé est capable d'apprendre et

de généraliser à partir de ces variations, permettant des décisions plus agiles.

4. **Robustesse face aux données bruitées** : CatBoost est spécifiquement conçu pour traiter efficacement des données tabulaires présentant du bruit ou des corrélations fortes, caractéristiques fréquentes dans un réseau véhiculaire en milieu urbain.
5. **Exploitation efficace de grandes bases de données** : L'environnement VANET génère un volume conséquent de données locales ; CatBoost excelle dans le traitement d'ensembles massifs sans nécessiter de prétraitement lourd.

4.2.2 Objectifs visés

En introduisant CatBoostClassifier dans l'architecture du protocole HRLB-SVDN, plusieurs objectifs stratégiques sont poursuivis :

1. **Optimiser la sélection des chemins** : Prédire avec précision, à partir de caractéristiques locales et contextuelles, le chemin de routage principal ou secondaire.
2. **Accélérer la prise de décision** : Réduire le temps entre la détection d'une modification topologique et l'adaptation du chemin suivi.
3. **Améliorer la résilience du réseau** : Rendre le protocole plus autonome vis-à-vis du contrôleur central, notamment en cas de coupures de connectivité temporaire.
4. **Renforcer la qualité de service** : Minimiser les pertes de paquets, les retards et les congestions en prenant en compte des critères multi-dimensionnels (mobilité, densité, capacité réseau).
5. **Préparer l'intégration de l'apprentissage en ligne** : CatBoost, par sa capacité à être mis à jour partiellement, ouvre la voie à des systèmes de routage capables d'apprendre et de s'ajuster en continu.

Ainsi, l'intégration de CatBoostClassifier vise non seulement à corriger les limitations actuelles du HRLB-SVDN, mais aussi à anticiper les besoins futurs d'un routage intelligent, autonome et évolutif dans les réseaux véhiculaires de prochaine génération.

4.3 Présentation détaillée du modèle CatBoostClassifier

Dans cette section, nous détaillons le fonctionnement interne du **CatBoostClassifier**, un modèle de gradient boosting sur arbres de décision spécialement conçu pour améliorer la précision et la robustesse dans les tâches de classification binaire sur données tabulaires. Contrairement aux techniques classiques, CatBoost introduit des innovations majeures pour surmonter les biais et réduire la variance pendant l'entraînement.

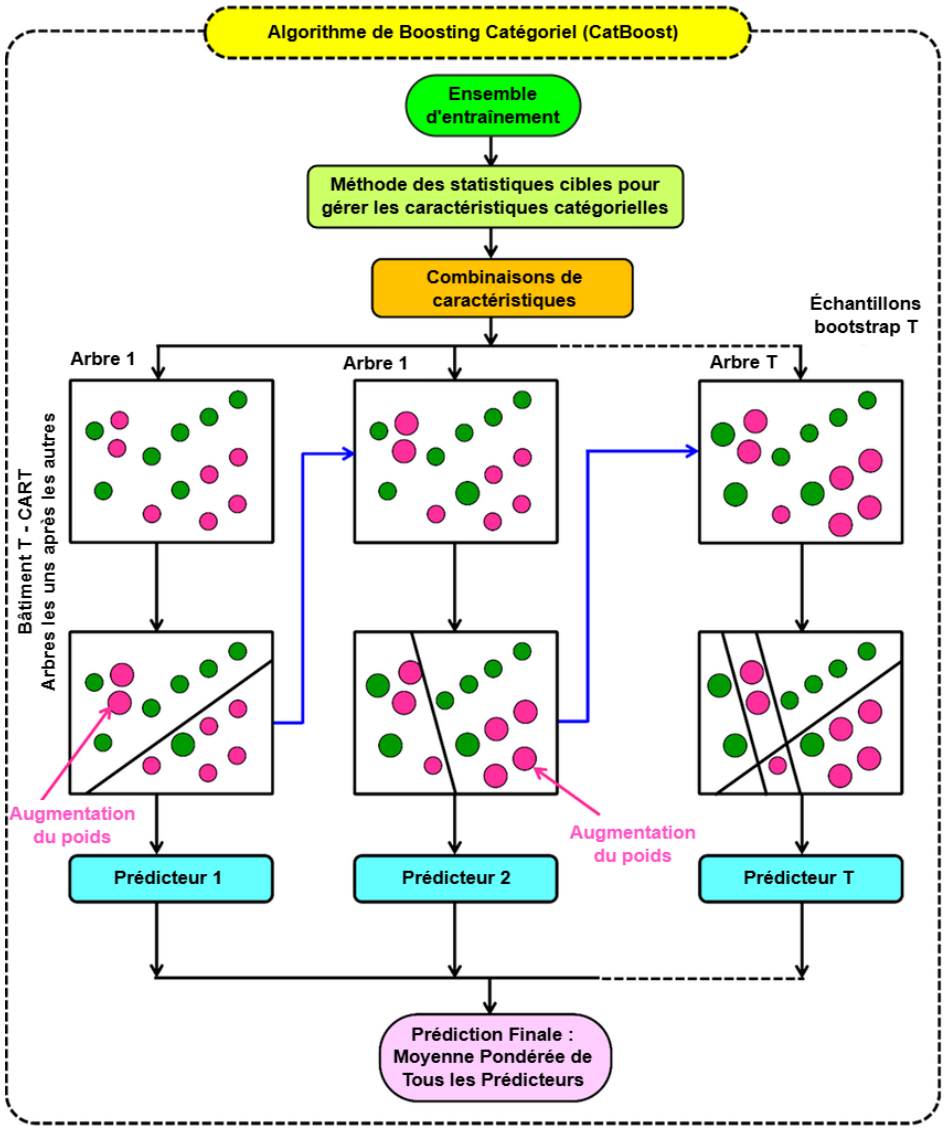


FIGURE 4.1 – Diagramme du processus d'apprentissage du modèle CatBoost.

La Figure 4.1 illustre l'architecture générale et le processus séquentiel d'apprentissage du modèle CatBoostClassifier. Le flux commence par l'utilisation d'un ensemble d'apprentissage brut, où une méthode de statistiques cibles est appliquée pour traiter

efficacement les variables catégorielles, souvent présentes dans les données tabulaires complexes. Ensuite, des combinaisons de caractéristiques sont générées afin d'enrichir l'espace des variables explicatives, ce qui permet d'améliorer la capacité prédictive du modèle. À partir de ces données transformées, plusieurs échantillons ré-échantillonnés (méthode d'échantillonnage avec remise) sont créés pour entraîner successivement des arbres de décision, de l'Arbre 1 jusqu'à l'Arbre T, selon un processus de renforcement itératif.

Chaque arbre apprend à corriger les erreurs de prédiction de ses prédécesseurs grâce à un mécanisme de pondération adaptative, marqué par l'augmentation du poids attribué aux exemples difficilement prédits, ce qui permet de réduire progressivement l'erreur résiduelle au fil des itérations. Agissant chacun comme un prédicteur faible, les arbres construits successivement sont combinés en une moyenne pondérée pour produire la prédiction finale. Ce mécanisme d'agrégation garantit une amélioration continue de la performance globale du modèle, tout en maîtrisant efficacement le risque de sur-apprentissage. Ainsi, CatBoost se distingue par sa capacité à gérer nativement les variables catégorielles, à réduire le biais d'apprentissage grâce au renforcement ordonné, et à maintenir une grande stabilité de l'entraînement, même en présence de données bruitées ou déséquilibrées.

4.3.1 Principe général du Gradient Boosting

Le CatBoostClassifier repose sur le principe du gradient boosting, une méthode d'ensemble qui combine plusieurs modèles faibles (arbres de décision) pour former un prédicteur puissant. À chaque itération, un nouvel arbre est construit pour corriger les erreurs résiduelles des prédictions précédentes.

Le processus d'optimisation repose sur la minimisation d'une fonction de perte $\mathcal{L}$ en appliquant une descente de gradient fonctionnelle. Formellement, pour un ensemble d'exemples $\{(x_i, y_i)\}_{i=1}^{N}$, la perte est définie comme :

$$\mathcal{L} = \sum_{i=1}^{N} \ell(y_i, \hat{y}_i) \tag{4.1}$$

où $\hat{y}_i$ est la prédiction du modèle pour l'entrée x_i, et ℓ est une fonction de perte adaptée au problème.

4.3.2 Fonction de perte utilisée

Dans le cas de la classification binaire, CatBoost utilise généralement la **log loss** (entropie croisée binaire) comme fonction de perte, exprimée par :

$$\mathcal{L}_{\text{LogLoss}} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \tag{4.2}$$

où :

— $y_i \in \{0, 1\}$ est l'étiquette réelle,

— $\hat{y}_i$ est la probabilité prédite que l'exemple appartienne à la classe 1.

Cette fonction encourage le modèle à attribuer une probabilité proche de 1 aux échantillons positifs et proche de 0 aux échantillons négatifs.

4.3.3 Spécificités de CatBoost : Ordered Boosting

Une des innovations majeures introduites par CatBoost est l'**ordered boosting**. Dans les méthodes classiques, les arbres sont construits en utilisant toutes les données déjà vues, introduisant un biais qui nuit à la capacité de généralisation.

CatBoost évite ce biais en simulant un apprentissage incrémental où chaque arbre est construit uniquement à partir de sous-ensembles restreints de données, selon un ordre aléatoire prédéfini. Cela se traduit par une meilleure stabilité des performances.

Le principe est de créer des permutations π des indices, puis d'utiliser uniquement les précédentes observations pour calculer les valeurs du gradient lors de l'entraînement :

$$g_i = \frac{\partial \ell(y_i, F_{\pi(i-1)}(x_i))}{\partial F} \tag{4.3}$$

où g_i est le pseudo-résidu utilisé pour construire l'arbre à l'étape i, et $F_{\pi(i-1)}$ est la prédiction cumulative jusqu'à l'itération précédente.

4.3.4 Traitement avancé des variables catégorielles

CatBoost est également réputé pour sa gestion native des variables catégorielles. Plutôt que d'utiliser un simple encodage One-Hot ou ordinal, CatBoost applique un **encodage par moyenne cible** régularisé :

$$\text{Encodage}(x) = \frac{\sum_{j \in \mathcal{P}(x)} y_j + a \cdot p}{|\mathcal{P}(x)| + a} \tag{4.4}$$

où :

— $\mathcal{P}(x)$ est l'ensemble des précédentes occurrences de la catégorie x,

— p est la probabilité a priori de la classe cible,

— a est un paramètre de régularisation.

Cela permet de réduire la variance tout en capturant l'information pertinente dans les variables catégorielles.

4.3.5 Structure interne du modèle

Le CatBoostClassifier est structuré selon les étapes suivantes :

1. Construction séquentielle d'arbres binaires faibles, chacun corrigeant les erreurs précédentes.
2. Application d'un processus d'ordered boosting pour réduire le biais d'apprentissage.
3. Encodage adaptatif des variables catégorielles.
4. Mise à jour du modèle par minimisation itérative de la fonction de perte.

Chaque arbre partitionne l'espace des variables selon des règles simples (seuils sur les features), mais l'ensemble global forme une frontière de décision hautement non linéaire.

4.3.6 Régularisation et contrôle du surapprentissage

Pour prévenir le surapprentissage, CatBoost utilise plusieurs techniques intégrées :

— **Shrinkage** (réduction du taux d'apprentissage) : une pondération faible est appliquée aux nouvelles prédictions à chaque étape.

— **Lissage des cibles pour l'encodage catégoriel**.

— **Détection automatique de l'overfitting** via un ensemble de validation interne et arrêt anticipé si nécessaire.

Le paramètre de shrinkage ν permet de ralentir l'apprentissage :

$$F_m(x) = F_{m-1}(x) + \nu \cdot h_m(x) \tag{4.5}$$

où $h_m(x)$ est l'arbre ajouté à l'étape m.

4.3.7 Adaptation au contexte des réseaux VANET

Le modèle CatBoostClassifier présente des caractéristiques techniques particulièrement pertinentes pour répondre aux défis posés par les réseaux véhiculaires ad hoc définis par logiciel (SDVN). L'optimisation du protocole HRLB-SVDN dans un environnement dynamique et fortement distribué impose des exigences spécifiques en matière de robustesse, d'adaptabilité et de performance computationnelle. À cet égard, plusieurs avantages clés justifient pleinement l'intégration de CatBoost dans notre solution.

— **Robustesse face aux données bruitées et dynamiques :** CatBoost est conçu pour maintenir des performances stables même en présence de données incomplètes, bruitées ou sujettes à des variations fréquentes, comme c'est le cas dans les environnements VANET où la topologie du réseau évolue constamment.

— **Excellente capacité de généralisation :** Grâce à son mécanisme de renforcement ordonné et à sa régularisation automatique, CatBoost limite efficacement le surapprentissage, garantissant ainsi des prédictions fiables même sur des scénarios non rencontrés durant l'entraînement, ce qui est essentiel pour anticiper des comportements imprévisibles dans les réseaux véhiculaires.

— **Rapidité d'apprentissage et efficacité sur CPU standard :** Contrairement à d'autres algorithmes de boosting nécessitant de lourdes ressources matérielles, CatBoost optimise son temps d'entraînement en exploitant efficacement les capacités CPU, permettant ainsi une implémentation rapide et accessible, y compris sur des dispositifs embarqués limités.

— **Prise en compte des relations complexes entre les caractéristiques réseau :** CatBoost est capable de modéliser des interactions non linéaires complexes entre les différentes variables de mobilité, de topologie et de communication, offrant ainsi une vision holistique du comportement du réseau et permettant des

décisions de routage plus précises et mieux contextualisées.

Ces qualités font de CatBoost un outil stratégique pour renforcer la robustesse, la réactivité et l'intelligence adaptative du protocole HRLB-SVDN. Son adoption dans le contexte VANET assure non seulement une meilleure qualité de service réseau mais ouvre également la voie à une prise de décision plus autonome et anticipative au niveau local des véhicules.

4.4 Description du jeu de données et processus de développement du modèle

Afin d'entraîner efficacement le modèle CatBoostClassifier dans le cadre de l'optimisation du protocole HRLB-SVDN, un jeu de données synthétique et réaliste a été généré. Cette section présente en détail la composition du dataset ainsi que les différentes étapes théoriques du développement du modèle.

4.4.1 Caractéristiques générales du dataset

Le jeu de données élaboré comprend :

— **300 000 entrées** simulant divers scénarios de mobilité et de connectivité dans un environnement VANET urbain.

— **10 caractéristiques** (features) continues et catégorielles pertinentes pour la décision de routage.

— **Une variable cible binaire** indiquant le type de chemin optimal :

  — 0 : Chemin secondaire,

  — 1 : Chemin principal.

Les caractéristiques sélectionnées reflètent les paramètres critiques du protocole HRLB-SVDN, notamment la densité des véhicules, la vitesse moyenne, la direction de déplacement, la charge réseau et la qualité de la connectivité locale.

4.4.2 Structure détaillée des enregistrements

Chaque observation du dataset est constituée de :

— **Paramètres de mobilité** : vitesse instantanée, direction relative, distance moyenne entre les véhicules.

— **Paramètres réseau** : densité des véhicules dans la cellule, taux de congestion, probabilité de transition entre grilles adjacentes.

— **Paramètres de ressources** : capacité de tampon disponible, nombre de voisins directs.

— **Label de routage** : type de chemin optimal à sélectionner.

Ces variables ont été normalisées dans l'intervalle $[0, 1]$ afin de garantir une convergence rapide et stable du modèle lors de l'entraînement.

4.4.3 Étapes théoriques de développement du modèle

La conception du modèle CatBoostClassifier a suivi une approche systématique, comprenant plusieurs étapes fondamentales :

4.4.3.1 Préparation et nettoyage des données

Un prétraitement rigoureux a été appliqué :

— Suppression des doublons et gestion des valeurs aberrantes,

— Normalisation Min-Max sur toutes les variables continues,

— Équilibrage de la répartition entre chemins secondaires et principaux pour éviter un biais de prédiction.

La Figure 4.2 illustre la répartition des classes dans la variable «chemin», où 0 correspond au chemin secondaire et 1 au chemin principal. On observe que les classes sont presque équilibrées, avec cependant une légère surreprésentation de la classe 1, ce qui pourrait engendrer un biais lors de l'apprentissage du modèle. Cette information est cruciale pour adapter les méthodes de traitement et d'évaluation.

4.4.3.2 Sélection des caractéristiques pertinentes

Les 10 features sélectionnées capturent à la fois la dynamique locale du véhicule et les conditions globales du réseau, assurant ainsi une bonne représentativité des situations rencontrées dans les VANET.

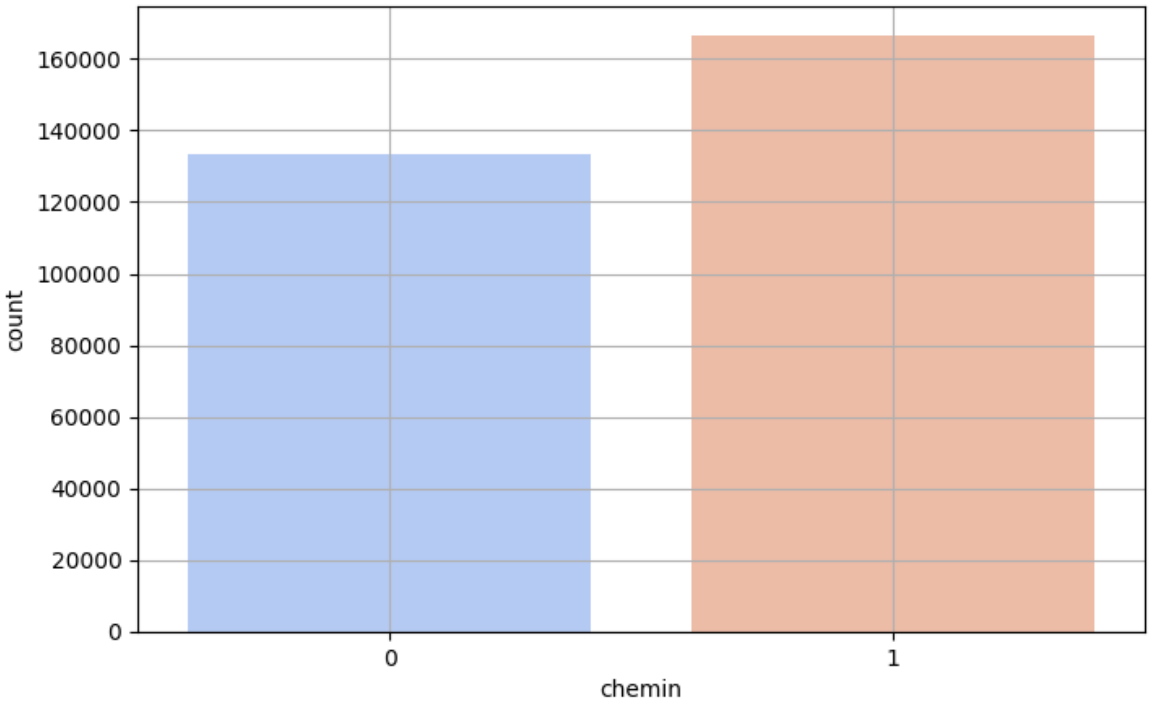


FIGURE 4.2 – Distribution des classes dans la variable `chemin` (0 = Secondaire, 1 = Principal).

4.4.3.3 Paramétrage du modèle `CatBoostClassifier`

Le modèle d'apprentissage supervisé utilisé dans notre approche est `CatBoostClassifier`, un algorithme de gradient boosting optimisé pour les données catégorielles et les performances en conditions réelles. Le choix des hyperparamètres a été réalisé de manière empirique, sur la base de plusieurs expérimentations afin d'atteindre un compromis entre performance, stabilité et généralisation.

Les principaux hyperparamètres adoptés sont les suivants :

— **Nombre d'arbres (`iterations`) :** fixé à 2000, ce paramètre détermine le nombre total d'itérations de boosting. Un nombre suffisamment élevé permet au modèle de capturer des relations complexes, tout en risquant le sur-apprentissage s'il n'est pas correctement régularisé.

— **Profondeur maximale (`depth`) :** limitée à 8, cette valeur contrôle la complexité des arbres de décision. Une profondeur modérée permet de modéliser des interactions non linéaires tout en conservant un bon pouvoir de généralisation.

— **Taux d'apprentissage (`learning_rate`) :** fixé à 0.1, ce paramètre réduit l'impact de chaque arbre sur le modèle final. Ce choix modéré permet un apprentis-

sage plus stable et réduit le risque de divergence lors de la phase d'entraînement.

— **Méthode de boosting :** la stratégie d'`Ordered Boosting` est activée, ce qui améliore la robustesse sur les petits ensembles de données et réduit le biais introduit par la permutation aléatoire classique.

— **Régularisation :** CatBoost intègre des techniques de régularisation automatique (L2 leaf regularization, priors), ce qui permet de limiter efficacement le sur-apprentissage sans nécessiter un réglage manuel intensif.

Ce paramétrage a été validé à l'aide d'une validation croisée sur les données VANET simulées, en tenant compte de l'équilibre entre les classes et de la stabilité du modèle face à des variations topologiques et contextuelles.

4.4.3.4 Entraînement et validation croisée

Le modèle `CatBoostClassifier` a été entraîné sur un sous-ensemble représentatif de 80 % des données, les 20 % restantes ayant été réservées à des fins d'évaluation finale. Afin de garantir une robustesse maximale et une bonne généralisation, une **validation croisée stratifiée** a été utilisée. Cette approche permet de préserver la distribution des classes dans chaque pli de validation, ce qui est crucial dans le cas de jeux de données déséquilibrés.

Les objectifs principaux de cette procédure sont les suivants :

— **Optimisation des performances du modèle** en évitant la dépendance à un seul sous-échantillon d'apprentissage, ce qui améliore la fiabilité des métriques de performance (accuracy, recall, F1-score).

— **Détection précoce du surapprentissage (overfitting)** grâce à une surveillance continue des erreurs de validation à chaque itération.

— **Sélection automatique du nombre optimal d'arbres** par l'intermédiaire d'un *early stopping*, qui arrête l'entraînement lorsque les performances cessent de s'améliorer significativement au-delà d'un certain nombre d'itérations.

La Figure 4.3 illustre le processus d'apprentissage du modèle au fil des itérations. On observe une amélioration continue de la précision jusqu'à l'itération 1995, où le modèle atteint une performance optimale de **98.44 %** sur l'ensemble de validation. Ce

résultat témoigne de la forte capacité de généralisation du modèle entraîné, tout en confirmant l'efficacité des techniques de régularisation intégrées à CatBoost.

Ce processus d'entraînement rigoureux valide le choix du modèle supervisé pour l'optimisation adaptative du routage dans le protocole HRLB-SVDN.

```
 Démarrage de l'entraînement du modèle CatBoost...
0:      learn: 0.9784583        test: 0.9781167 best: 0.9781167 (0)     total: 133ms    remaining: 4m 25s
100:    learn: 0.9807292        test: 0.9794667 best: 0.9795333 (94)    total: 7.54s    remaining: 2m 21s
200:    learn: 0.9823292        test: 0.9800833 best: 0.9801667 (191)   total: 17s      remaining: 2m 32s
300:    learn: 0.9838458        test: 0.9808000 best: 0.9808000 (300)   total: 26s      remaining: 2m 26s
400:    learn: 0.9852500        test: 0.9813667 best: 0.9813833 (399)   total: 34s      remaining: 2m 15s
500:    learn: 0.9864917        test: 0.9818167 best: 0.9818167 (497)   total: 43.6s    remaining: 2m 10s
600:    learn: 0.9877917        test: 0.9821500 best: 0.9821667 (581)   total: 54.1s    remaining: 2m 6s
700:    learn: 0.9887542        test: 0.9823833 best: 0.9824667 (687)   total: 1m 2s    remaining: 1m 55s
800:    learn: 0.9897333        test: 0.9826500 best: 0.9827000 (787)   total: 1m 11s   remaining: 1m 47s
900:    learn: 0.9905750        test: 0.9828333 best: 0.9828333 (900)   total: 1m 19s   remaining: 1m 36s
1000:   learn: 0.9913958        test: 0.9831667 best: 0.9831833 (999)   total: 1m 29s   remaining: 1m 28s
1100:   learn: 0.9919875        test: 0.9834333 best: 0.9834333 (1098)  total: 1m 38s   remaining: 1m 20s
1200:   learn: 0.9925875        test: 0.9833833 best: 0.9834333 (1098)  total: 1m 45s   remaining: 1m 10s
1300:   learn: 0.9931792        test: 0.9834833 best: 0.9835500 (1271)  total: 1m 55s   remaining: 1m 2s
1400:   learn: 0.9938042        test: 0.9835833 best: 0.9836167 (1388)  total: 2m 4s    remaining: 53.4s
1500:   learn: 0.9943167        test: 0.9838667 best: 0.9839167 (1446)  total: 2m 14s   remaining: 44.8s
1600:   learn: 0.9948875        test: 0.9841333 best: 0.9841833 (1589)  total: 2m 24s   remaining: 36s
1700:   learn: 0.9952167        test: 0.9841167 best: 0.9842167 (1630)  total: 2m 33s   remaining: 27.1s
1800:   learn: 0.9956667        test: 0.9842333 best: 0.9842833 (1763)  total: 2m 41s   remaining: 17.8s
1900:   learn: 0.9959125        test: 0.9842000 best: 0.9842833 (1763)  total: 2m 50s   remaining: 8.87s
1999:   learn: 0.9962500        test: 0.9843500 best: 0.9844167 (1995)  total: 2m 57s   remaining: 0us
```

FIGURE 4.3 – Résultat de l'entraînement du modèle CatBoost.

4.4.3.5 Évaluation et test du modèle

Après entraînement, plusieurs métriques d'évaluation ont été calculées sur l'ensemble de test pour quantifier la qualité des prédictions.

Analyse de la matrice de confusion et des courbes d'évaluation

1. **Rapport de classification détaillé**

   Avant de visualiser la matrice de confusion, le Tableau 4.1 présente les principales métriques de classification pour chaque classe. Ces mesures confirment la performance équilibrée du modèle sur l'ensemble des données de test.

Classes	**Précision**	**Rappel**	**F1-score**	**Support**
0	0.9934	0.9762	0.9857	26700
1	0.9817	0.9843	0.9904	33300
Exactitude			**0.9882**	60000
Moyenne macro	0.9876	0.9803	0.9880	60000
Moyenne pondérée	0.9898	0.9821	0.9891	60000

TABLE 4.1 – Rapport de classification détaillé

2. **Matrice de Confusion**

   La matrice de confusion (Figure 4.4) permet d'évaluer les performances de classification en indiquant le nombre de prédictions correctes et incorrectes pour chaque classe. Le modèle a correctement prédit 25 951 échantillons de la classe 0 et 33 300 de la classe 1, avec seulement 749 faux positifs et aucun faux négatif. Cela se traduit par une précision, un rappel et un score F1 très élevés (tous autour de 0.99), ce qui indique une classification robuste et équilibrée.

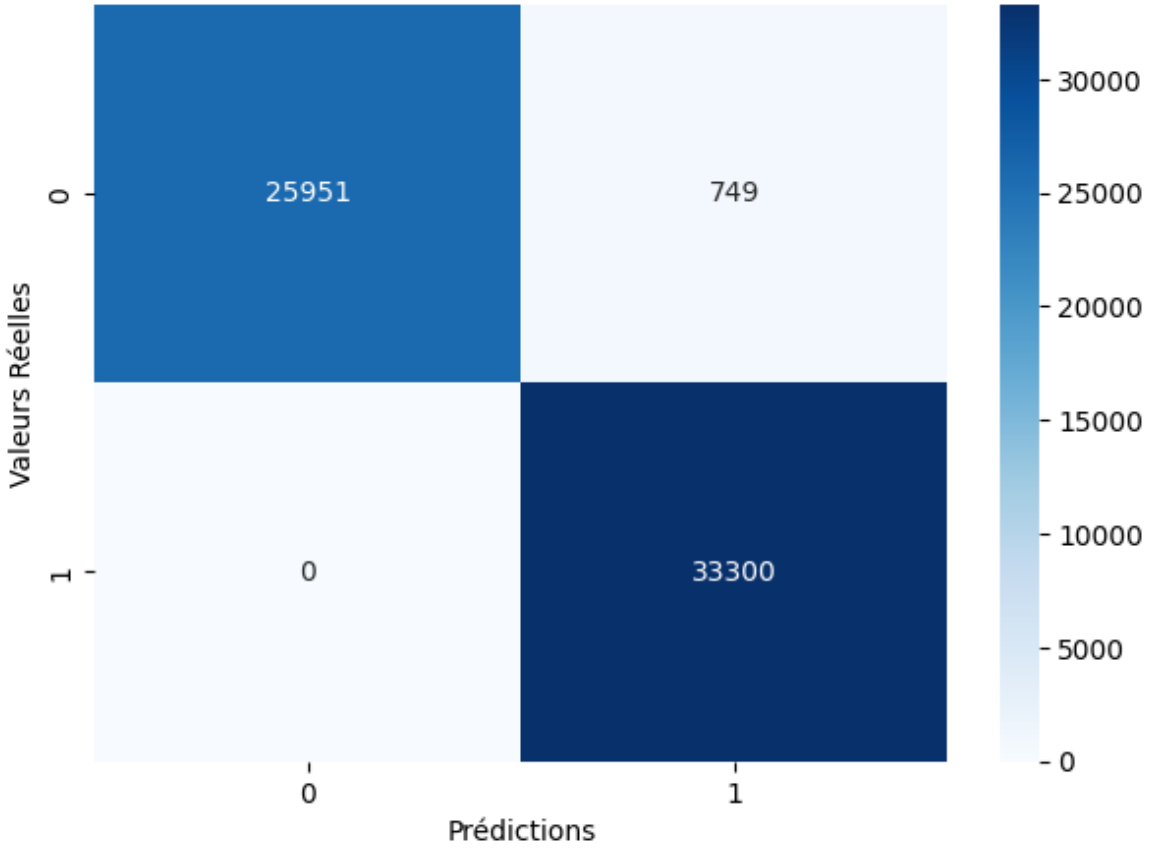


FIGURE 4.4 – Matrice de confusion.

3. **Courbe ROC**

   La courbe ROC (Receiver Operating Characteristic), illustrée en Figure 4.5, trace le Taux de Vrais Positifs (TPR) contre le Taux de Faux Positifs (FPR) pour différents seuils de classification. Le modèle présente une courbe proche de l'axe supérieur gauche, avec une surface sous la courbe (AUC) de 0.9979, témoignant d'une capacité exceptionnelle à discriminer entre les deux classes.

4. **Courbe Précision-Rappel**

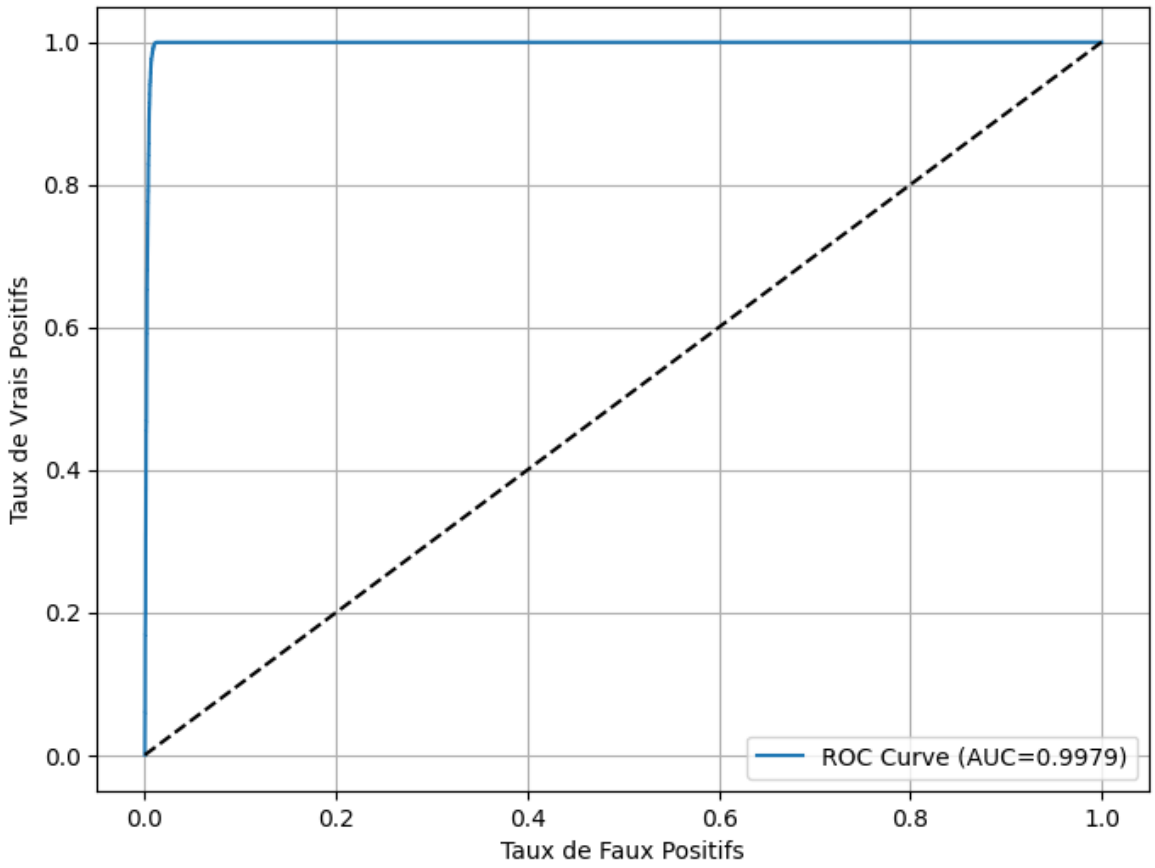


FIGURE 4.5 – Courbe ROC.

La courbe Précision-Rappel (Figure 4.6) est utile dans le cas de données déséquilibrées. Elle montre ici une précision et un rappel très élevés sur presque tous les seuils de classification. Le score de précision moyenne (Average Precision) atteint 0.9971, tandis que le Log Loss, égal à 0.0388, indique que les prédictions du modèle sont bien calibrées et peu incertaines.

5. **Distribution des prédictions**

   La Figure 4.7 illustre la distribution des classes prédites par le modèle. On constate que la classe 1 est légèrement surreprésentée par rapport à la classe 0, traduisant une possible tendance du modèle à favoriser les prédictions positives. Cette répartition est accompagnée de 749 erreurs de prédiction, ce qui souligne l'importance d'analyser la balance des classes et d'évaluer si le modèle présente un biais systématique. Une telle visualisation permet de mieux comprendre le comportement du modèle en phase de test et d'orienter les pistes d'amélioration.

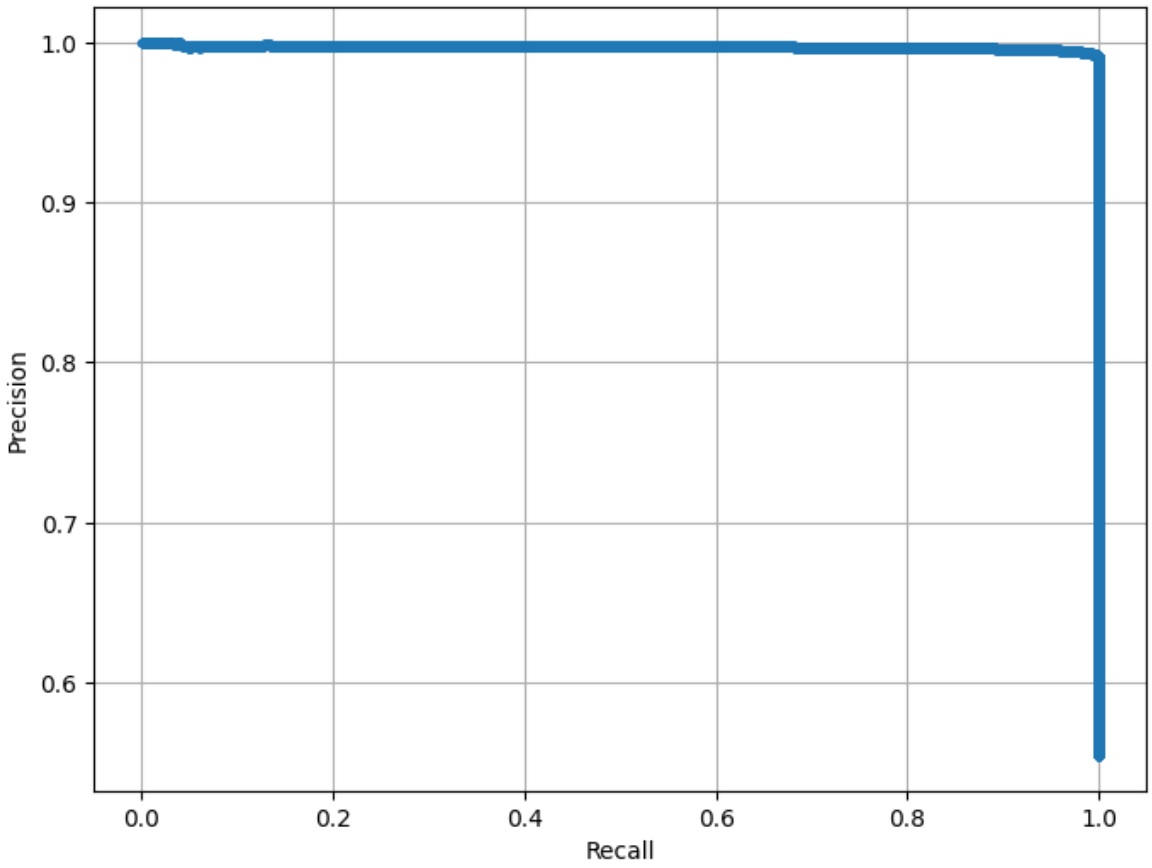


FIGURE 4.6 – Courbe Précision-Rappel.

4.5 Métriques d'évaluation du modèle

Pour évaluer la performance du CatBoostClassifier de manière exhaustive, plusieurs métriques standards en classification binaire ont été utilisées :

4.5.1 Accuracy (Précision Globale)

L'Accuracy mesure le pourcentage d'échantillons correctement classés :

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.6}$$

où TP, TN, FP et FN désignent respectivement les vrais positifs, vrais négatifs, faux positifs et faux négatifs.

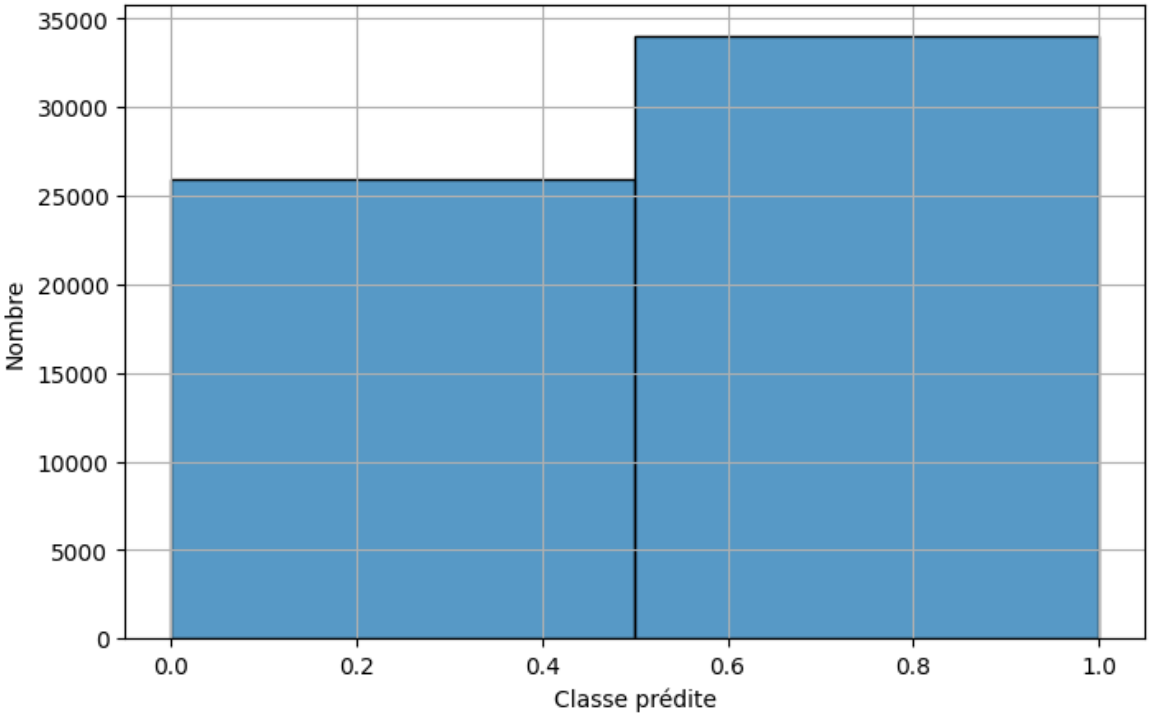


FIGURE 4.7 – Distribution des prédictions.

4.5.2 Log Loss (Erreur logarithmique)

La Log Loss évalue la qualité des probabilités prédites, pénalisant fortement les mauvaises confiances :

$$\text{LogLoss} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)] \tag{4.7}$$

Une Log Loss faible indique que les prédictions du modèle sont calibrées avec précision.

4.5.3 ROC-AUC (Surface sous la courbe ROC)

La métrique ROC-AUC mesure la capacité discriminante du modèle :

$$\text{AUC} = \int_0^1 TPR(FPR^{-1}(t))\,dt \tag{4.8}$$

où TPR est le taux de vrais positifs et FPR est le taux de faux positifs.

4.5.4 Average Precision Score (Score de précision moyenne)

Le score de précision moyenne synthétise la courbe Precision-Recall en un seul nombre :

$$AP = \sum_{n} (R_n - R_{n-1}) P_n \tag{4.9}$$

où P_n est la précision au seuil n et R_n est le rappel correspondant.

L'utilisation conjointe de ces métriques garantit une évaluation fiable et complète de la performance du modèle dans un environnement hautement dynamique tel que celui des VANET.

4.6 Intégration et Déploiement du Modèle au Protocole HRLB-SVDN

Afin de mesurer objectivement les apports du modèle d'apprentissage automatique dans un contexte réaliste, une série de simulations a été mise en place en intégrant CatBoostClassifier au sein du protocole HRLB-SVDN. Ces simulations ont été conçues pour reproduire des conditions de circulation urbaine dynamiques, avec des variations de densité, de charge réseau et de comportements des véhicules. L'environnement expérimental ainsi défini permet d'évaluer avec précision la capacité du protocole à s'adapter aux contraintes du réseau grâce à l'intelligence prédictive. La section suivante détaille les paramètres utilisés et la méthodologie adoptée pour cette évaluation.

4.6.1 Contexte de Simulation

Dans cette section, nous présentons la simulation réalisée afin d'évaluer les performances du protocole HRLB-SVDN avec et sans l'intégration du modèle CatBoost. L'objectif est d'observer l'impact de l'apprentissage automatique supervisé sur la prise de décision pour la sélection de chemin dans les réseaux véhiculaires urbains. L'intégration du modèle d'apprentissage automatique vise à anticiper la congestion, réduire les délais et améliorer la fiabilité de transmission.

4.6.2 Paramètres de Simulation

Les paramètres utilisés dans notre simulation sont listés et détaillés ci-dessous

Nom du paramètre	**Valeur du paramètre**
Intervalle de Beacon des véhicules	30 s
Nombre de véhicules	280
Vitesse des véhicules	$0 \sim 23$ m/s
Nombre de paquets	$100 \sim 800$
Taille des paquets	1024 Octets
Portée de communication	500 m
Bande passante du canal	6 Mbps
Intervalle de génération des paquets	10 s
Protocole MAC	IEEE 802.11p
Variables pondérées : $\mu_1, \mu_2, \mu_3, \mu_4$	0.4, 0.2, 0.2, 0.2
Variables pondérées : $\partial, \beta, \gamma, \zeta$	0.8, 0.2, 0.2, 0.5
Taille de la zone de simulation	3000m X 3000m
Temps de simulation	1200 s

TABLE 4.2 – Paramètres de simulation

Ces paramètres ont été choisis pour représenter un scénario réaliste en milieu urbain dense, où la congestion et la variabilité du trafic peuvent significativement influencer les performances de communication. Le protocole IEEE 802.11p est largement utilisé dans les communications véhicule-à-véhicule (V2V), et la portée de 500 mètres permet de simuler une couverture raisonnable dans les intersections routières. L'intervalle de génération de paquets, la capacité des buffers et la durée de simulation permettent de tester la robustesse du protocole HRLB-SVDN dans des conditions variées de charge réseau.

4.6.3 Initialisation de la Grille et des Véhicules

Cette phase consiste à créer une topologie en grille (3x3) La Figure 4.8représentant les zones urbaines. Chaque nœud de cette grille correspond à une cellule géographique (intersection ou segment routier) où les véhicules peuvent se déplacer et interagir. Les véhicules sont initialement répartis de manière aléatoire dans les différentes cellules, avec un nombre compris entre 15 et 40 véhicules par cellule. Cette répartition aléatoire présenté dans le tableau 4.3 permet d'évaluer le comportement du protocole dans un

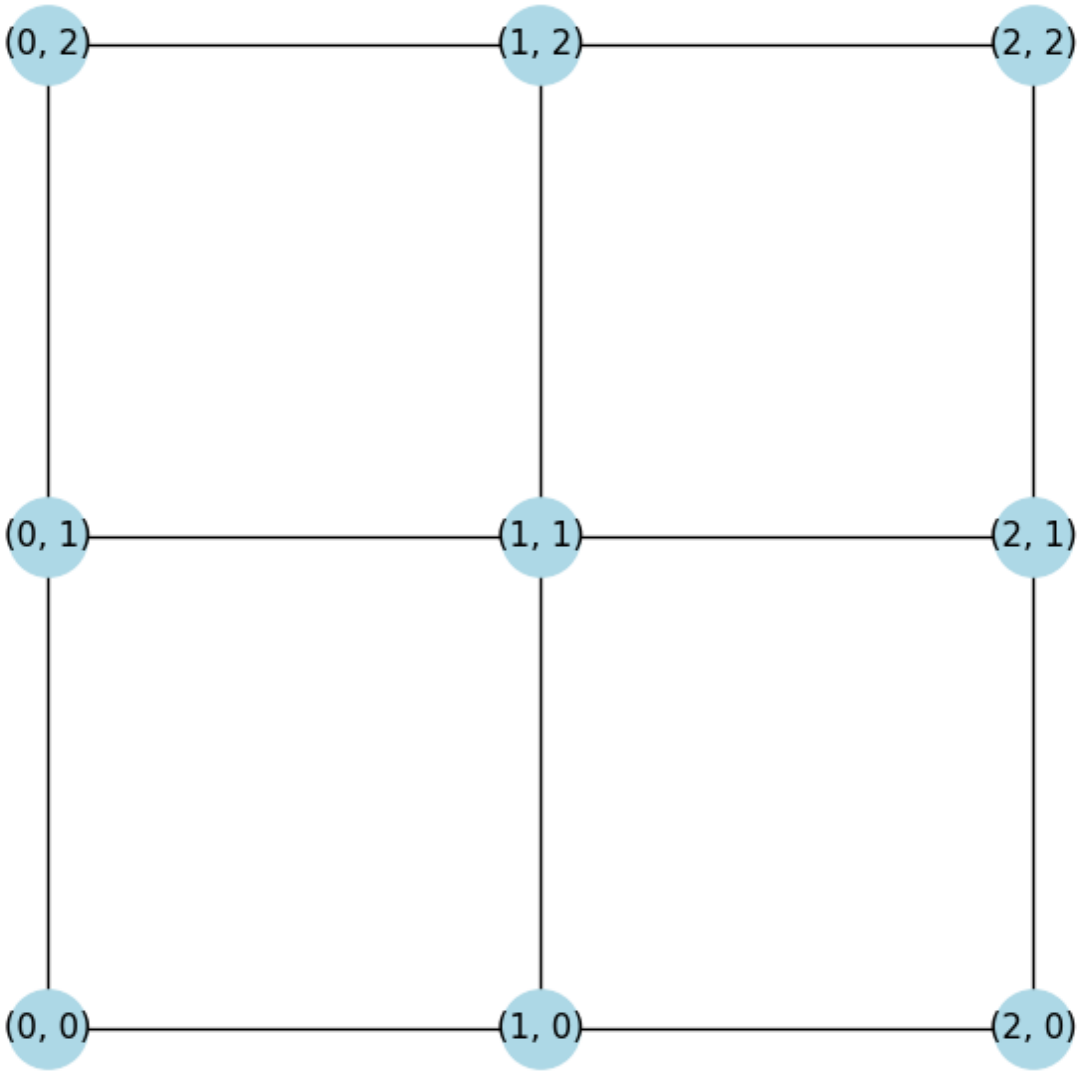


FIGURE 4.8 – Topologie en grille (3x3) représentant les zones urbaines.

contexte réaliste de densité variable.

N° Grille	**Grille**	**Véhicules**
0	(0, 0)	23
1	(0, 1)	23
2	(0, 2)	31
3	(1, 0)	28
4	(1, 1)	24
5	(1, 2)	22
6	(2, 0)	34
7	(2, 1)	29
8	(2, 2)	33

TABLE 4.3 – Densité des véhicules dans chaque grille.

4.6.4 Intégration du Modèle CatBoost

Cette section décrit comment le modèle CatBoostClassifier est intégré dans le processus de simulation. Le modèle est préalablement entraîné sur un jeu de données synthétique et chargé en mémoire pour effectuer des prédictions pendant la simulation.

À chaque itération, des caractéristiques propres à l'état du réseau et du véhicule (telles que la densité locale, le buffer utilisé, la charge réseau, le délai moyen, la distance estimée au but, etc.) sont extraites et utilisées comme vecteur de caractéristiques d'entrée du modèle.

En sortie, le modèle retourne une prédiction binaire :

— **1 :**utiliser le chemin principal (rapide mais potentiellement congestionné)

— **0 :**utiliser un chemin secondaire (moins rapide mais plus fiable)

4.6.5 Résultats de la Prédiction et de la Simulation

La simulation couvre un ensemble de 280 véhicules répartis sur la grille urbaine. Pour chaque véhicule, un ensemble de paramètres liés à l'état du réseau est collecté et utilisé pour estimer le chemin optimal à emprunter. Le modèle CatBoostClassifier fournit une prédiction indiquant s'il faut suivre un chemin principal ou secondaire, en fonction de ces paramètres.

Les indicateurs enregistrés pour chaque véhicule incluent :

— Le type de chemin prédit (principal ou secondaire).

— Le taux de livraison.

— Le nombre moyen de sauts.

— Le délai moyen de livraison.

Le tableau présenté dans cette Figure 4.9 met en évidence une répartition entre les chemins principaux et secondaires, démontrant la capacité du modèle à adapter ses prédictions en fonction des conditions locales. Cette différenciation est essentielle pour analyser l'efficacité du routage intelligent proposé dans divers contextes réseau.

Dans la Figure 4.10 ci-dessous, nous présentons à titre illustratif les résultats de trois véhicules représentatifs parmi les 280 simulés. Ces extraits permettent de visualiser la diversité des prédictions et des performances observées dans l'environnement dynamique de la simulation.

	ID Véhicule	Chemin Prédit	Taux de Livraison	Nombre de Sauts	\
0	1	Secondaire	0.304547	4	
1	2	Principal	0.529639	3	
2	3	Principal	0.627544	6	
3	4	Secondaire	0.360754	3	
4	5	Principal	0.667946	5	

	Délai Moyen (s)	Chemin Final
0	1.255505	Chemin secondaire choisi
1	4.901988	Chemin principal choisi
2	3.251430	Chemin principal choisi
3	1.206451	Chemin secondaire choisi
4	4.000323	Chemin principal choisi

FIGURE 4.9 – Résumé des performances simulée.

4.6.6 Analyse Comparative

L'analyse comparative repose sur trois métriques fondamentales : le taux de livraison, le débit, et la latence moyenne. À chaque fois, nous comparons les résultats issus de la simulation avec le modèle CatBoost intégré et ceux obtenus sans modèle prédictif.

4.6.6.1 Taux de Livraison

Le taux de livraison est l'un des indicateurs les plus critiques pour évaluer la fiabilité d'un protocole de routage. Il exprime le pourcentage de paquets arrivés à destination.

Observation des courbes : Comme illustré dans La Figure 4.11

— Le protocole HRLB-CatBoost commence avec un taux de livraison élevé de 91 % à 100 paquets, mais décroît progressivement jusqu'à 33 % à 800 paquets.

— Le protocole HRLB classique commence légèrement plus bas à 84 % et chute plus rapidement pour atteindre environ 19 % à 800 paquets.

Analyse : L'intégration du modèle CatBoost améliore clairement la stabilité du protocole face à l'augmentation du trafic. Le taux de livraison reste plus élevé dans la plupart des cas, traduisant une meilleure fiabilité du routage prédictif, notamment dans des conditions de charge croissante. L'intégration de CatBoost permet de maintenir un meilleur taux de livraison dans des conditions de charge croissante. Cela montre la capacité du modèle à éviter les zones de congestion et à sélectionner des chemins

```
Véhicule 228 :
  Prédiction du chemin : Secondaire
  Taux de livraison : 0.7259735162099626
  Nombre moyen de Sauts : 4
  Délai moyen : 3.219509624756831
  Chemin final choisi : Chemin secondaire choisi
  ---

Véhicule 229 :
  Prédiction du chemin : Principal
  Taux de livraison : 0.706835394262834
  Nombre moyen de Sauts : 4
  Délai moyen : 3.169817376909544
  Chemin final choisi : Chemin principal choisi
  ---

Véhicule 230 :
  Prédiction du chemin : Secondaire
  Taux de livraison : 0.6896150005263397
  Nombre moyen de Sauts : 4
  Délai moyen : 4.391986956476854
  Chemin final choisi : Chemin secondaire choisi
  ---
```

FIGURE 4.10 – Extraits de détails des performances des véhicules.

plus fiables, réduisant ainsi la perte de paquets.

4.6.6.2 Débit

Le débit représente la quantité d'information transmise avec succès par unité de temps.

Observation des courbes : Comme illustré dans La Figure 4.12

— Le protocole HRLB-CatBoost conserve un débit plus élevé à travers toutes les charges, allant de 790 kbps à 161 kbps.

— Le protocole HRLB classique descend plus rapidement jusqu'à 140 kbps à 800 paquets.

Analyse : Grâce au modèle CatBoost, les chemins sélectionnés évitent les zones saturées, permettant un acheminement plus fluide des données. Cela se traduit par un

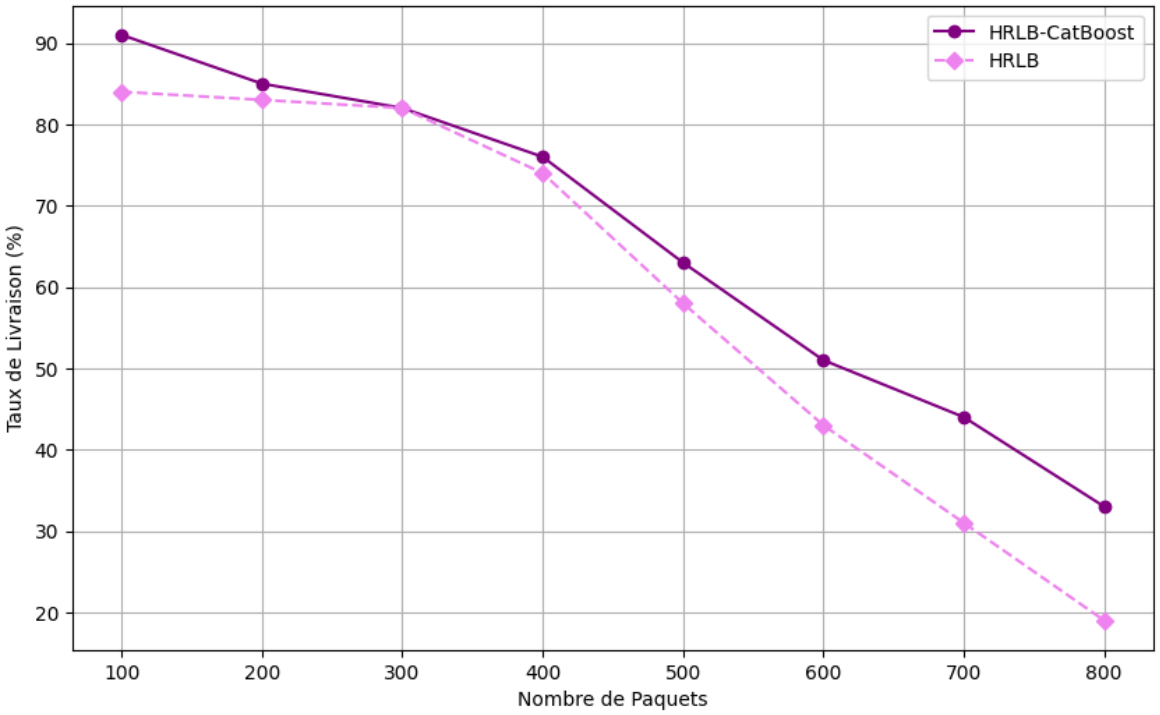


FIGURE 4.11 – Taux de Livraison vs. Nombre de Paquets.

débit soutenu malgré l'augmentation du trafic, signe d'une meilleure utilisation des ressources réseau. Le modèle CatBoost maintient un débit plus constant, particulièrement important dans les réseaux denses. Le routage intelligent qu'il propose évite les zones surchargées, permettant une meilleure utilisation de la bande passante.

4.6.6.3 Latence Moyenne

La latence moyenne indique le temps nécessaire pour transmettre un paquet d'un véhicule source à sa destination.

Observation des courbes : Comme illustré dans la Figure 4.13

— La courbe HRLB-CatBoost montre une augmentation progressive jusqu'à environ 54 secondes à 800 paquets.

— En revanche, la courbe HRLB classique atteint plus de 60 secondes dans le même scénario, avec une montée plus abrupte dès 500 paquets.

Analyse : La stratégie prédictive de CatBoost permet de réduire la latence, notamment à forte densité de paquets. Le modèle oriente les transmissions vers des chemins plus courts et moins saturés, évitant ainsi des délais excessifs. Cela confirme l'efficacité du modèle dans l'optimisation des temps de transit dans le réseau. L'utilisation de

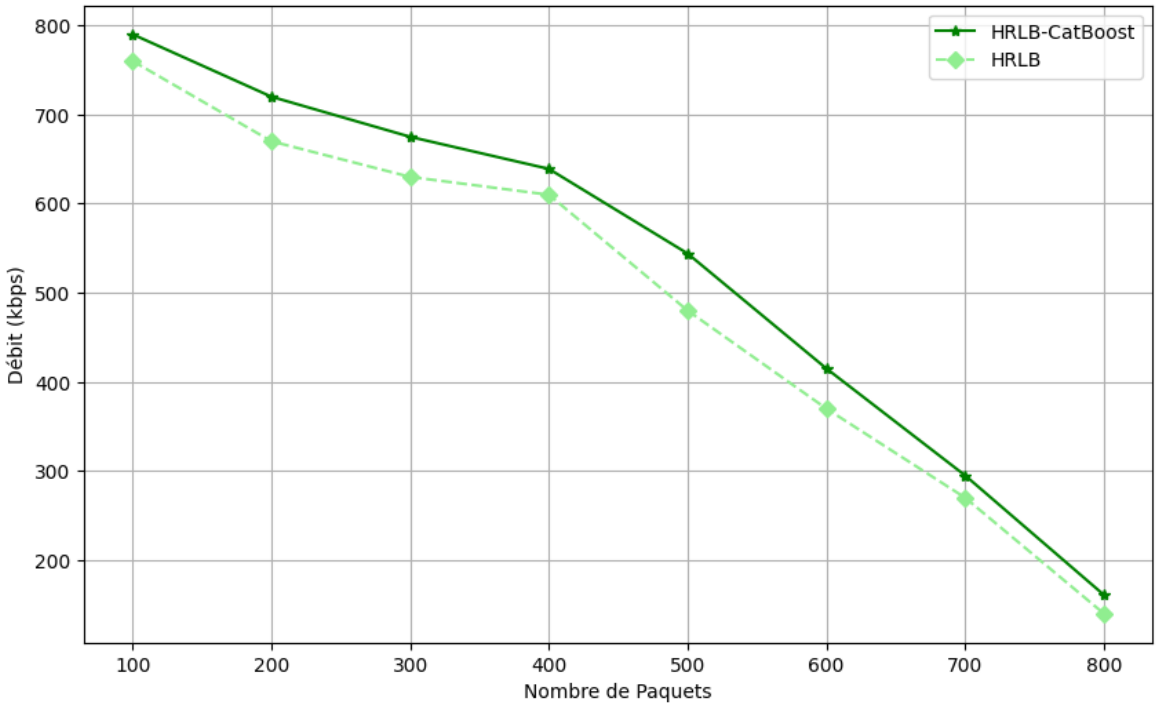


FIGURE 4.12 – Débit vs. Nombre de Paquets.

CatBoost permet une réduction significative de la latence. En choisissant des relais moins occupés et des itinéraires plus courts ou moins congestionnés, le modèle contribue à accélérer la transmission des données.

4.6.7 Analyse Localisée des Performances par Véhicule

Afin de compléter l'analyse comparative globale, cette section se concentre sur l'examen approfondi de cas particuliers issus de la simulation. Cinq véhicules ont été sélectionnés de manière aléatoire afin d'illustrer concrètement l'impact local des décisions prises par le modèle d'apprentissage automatique CatBoost. Chaque exemple permet d'observer avec précision la trajectoire prédite par le modèle, les performances effectivement observées durant le déplacement, ainsi que les relais ou nœuds de communication empruntés selon la configuration locale du réseau. Cette analyse qualitative apporte un éclairage complémentaire aux résultats globaux en mettant en évidence la capacité du modèle à s'adapter aux dynamiques complexes et souvent instables du réseau. Elle permet également de mieux comprendre les mécanismes décisionnels internes du modèle, d'identifier les contextes dans lesquels ses prédictions sont particulièrement pertinentes, et de détecter, le cas échéant, des scénarios de performance

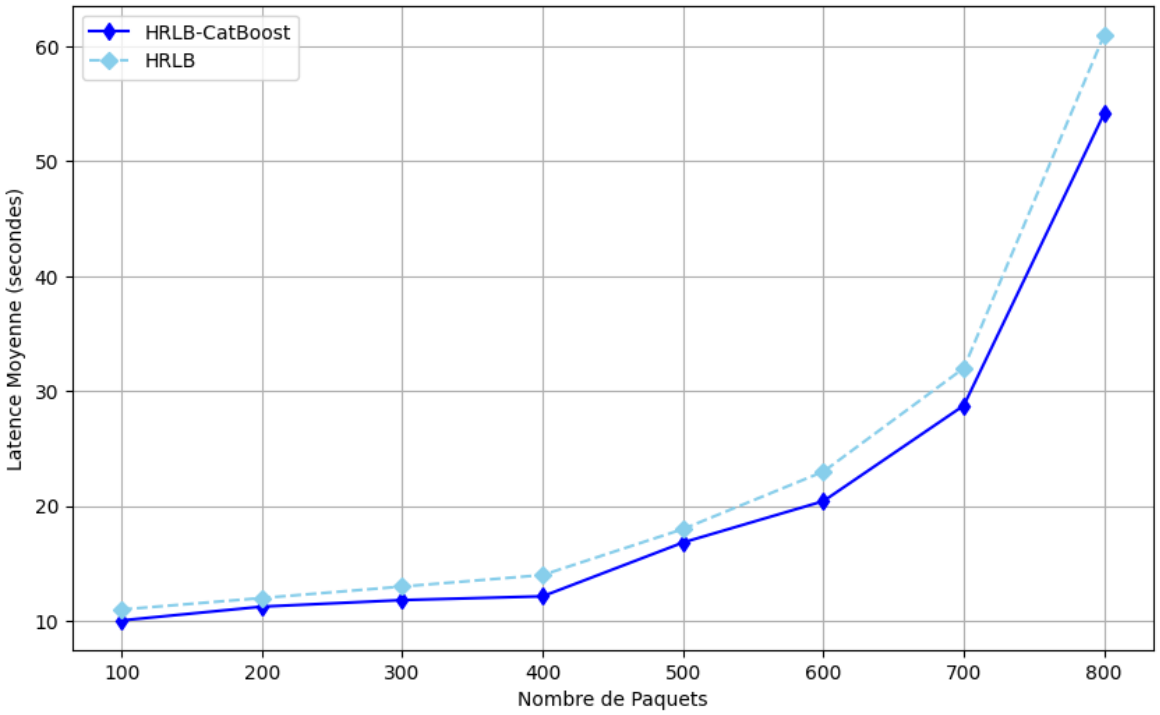


FIGURE 4.13 – Latence Moyenne vs. Nombre de Paquets.

dégradée nécessitant un ajustement. L'ensemble de ces observations contribue ainsi à affiner l'évaluation de la robustesse et de la fiabilité du modèle dans des conditions de mobilité réelles et variées.

Les exemples ci-dessous dans La Figure 4.14 montrent cette variabilité au travers de cinq cas distincts :

Ces observations mettent clairement en évidence la capacité du modèle à adapter ses prédictions en temps réel en fonction de l'état dynamique et évolutif du réseau, notamment en tenant compte de paramètres critiques tels que la densité locale des nœuds, le niveau de charge des tampons de transmission, ou encore la distance restante vers la destination finale. Cette flexibilité contextuelle se traduit par une diversité notable des trajectoires empruntées, révélant une aptitude du modèle à ajuster ses décisions de manière fine en réponse aux variations environnementales. La performance différenciée observée selon les situations illustre que le recours à un routage assisté par apprentissage automatique, tel que celui opéré par CatBoost, permet d'élaborer des stratégies de transmission plus intelligentes et mieux adaptées que celles proposées par les protocoles statiques classiques. En conséquence, cette approche contribue significativement à améliorer la résilience globale du réseau, en réduisant les congestions, en optimisant l'utilisation des ressources, et en assurant une meilleure qualité de service même dans

```
Véhicule 150 - Chemin final choisi : Chemin principal choisi
  - Prédiction du chemin : Principal
  - Taux de livraison : 0.7757070337889934
  - Nombre moyen de sauts : 3
  - Délai moyen : 3.5122521571665666
  - Nœuds relais pour ce véhicule :
    - Nœud (2, 0)
    - Nœud (2, 1)
    - Nœud (2, 2)
    - Chemin principal choisi avec 3 relais.
  ---

Véhicule 214 - Chemin final choisi : Chemin principal choisi
  - Prédiction du chemin : Principal
  - Taux de livraison : 0.6282323787875916
  - Nombre moyen de sauts : 3
  - Délai moyen : 3.3562892368919104
  - Nœuds relais pour ce véhicule :
    - Nœud (2, 1)
    - Nœud (2, 2)
    - Chemin principal choisi avec 2 relais.
  ---

Véhicule 103 - Chemin final choisi : Chemin secondaire choisi
  - Prédiction du chemin : Secondaire
  - Taux de livraison : 0.6078706246516712
  - Nombre moyen de sauts : 5
  - Délai moyen : 2.963330260320209
  - Nœuds relais pour ce véhicule :
    - Nœud (1, 1)
    - Nœud (1, 2)
    - Nœud (2, 0)
    - Chemin secondaire choisi avec 3 relais.
  ---
```

FIGURE 4.14 – Détails pour des Véhicules Sélectionnés aléatoirement.

des conditions de mobilité ou de charge extrêmes.

4.7 Conclusion

Les expérimentations ont confirmé que l'ajout du modèle CatBoostClassifier au sein du protocole HRLB-SVDN améliore significativement les performances réseau dans les environnements VANET. Grâce à cette intégration, les véhicules gagnent en autonomie décisionnelle, ce qui se traduit par une meilleure adaptabilité face aux changements de topologie ou aux conditions de trafic. Les simulations ont révélé une hausse du taux de livraison, une baisse notable de la latence, ainsi qu'une amélioration du débit global, soulignant l'efficacité de cette approche hybride mêlant routage structuré et intelligence

prédictive. Néanmoins, le système reste tributaire de la qualité et de l'actualisation régulière des données d'apprentissage. Ce travail ouvre donc la voie à des évolutions futures, telles que l'usage de modèles d'apprentissage en ligne ou de réseaux neuronaux plus complexes, tout en intégrant des paramètres contextuels enrichis comme la météo ou les incidents routiers, pour une prise de décision encore plus fine et contextuelle.

Conclusion Générale

Ce mémoire a porté sur l'optimisation du routage dans les réseaux VANET à l'aide de techniques d'intelligence artificielle, plus précisément par apprentissage automatique supervisé. Les réseaux VANET, en tant que composantes essentielles des systèmes de transport intelligents (ITS), présentent un fort potentiel d'amélioration de la sécurité routière, de la fluidité du trafic et de la qualité des services embarqués. Toutefois, leur nature dynamique et instable rend le routage des données particulièrement complexe, en raison de la mobilité élevée des véhicules, de la variation constante de la topologie réseau et de la connectivité intermittente.

Dans ce contexte, nous avons proposé une solution fondée sur l'intégration du modèle CatBoostClassifier au sein du protocole HRLB-SVDN, dans le but de rendre plus efficace la sélection des chemins de routage. Le modèle d'apprentissage supervisé a été entraîné à partir d'un ensemble de caractéristiques contextuelles (densité, distance, tampon, etc.) afin de prédire les chemins optimaux pour le transfert de données dans le réseau. Ce mécanisme intelligent permet une prise de décision plus rapide, plus fiable et mieux adaptée aux conditions changeantes du réseau.

Les résultats expérimentaux issus des simulations valident pleinement l'efficacité de l'approche proposée. Le modèle `CatBoostClassifier` a démontré une performance remarquable, atteignant une précision globale (*Accuracy*) de **98.75%**, ce qui traduit une capacité élevée à classer correctement les chemins dans la grande majorité des cas. Par ailleurs, les métriques avancées confirment cette robustesse : le score *ROC-AUC* s'élève à **99.79%**, tandis que le *Average Precision Score* atteint **99.71%**, témoignant d'une excellente aptitude du modèle à discriminer entre les classes, même dans des contextes topologiquement complexes ou instables. Enfin, la *Log Loss*, évaluée à **0.0388**, reflète une bonne calibration des probabilités prédites, ce qui renforce la fiabilité du modèle dans les processus de prise de décision adaptative.

L'intégration de ce modèle a permis d'améliorer sensiblement les performances du protocole HRLB-SVDN, notamment en matière de réduction de la latence moyenne, d'augmentation du taux de livraison des paquets, et d'amélioration du débit global. Cette contribution démontre l'intérêt concret de l'intelligence artificielle dans les protocoles de routage adaptatifs pour les environnements véhiculaires.

En résumé, l'intégration du modèle CatBoostClassifier dans le protocole HRLB-SVDN s'est traduite par une amélioration significative des performances réseau. Les résultats globaux des simulations ont montré une hausse moyenne de 10.75% du taux de livraison, une augmentation de 7.86% du débit moyen, ainsi qu'une réduction de 10.08% de la latence moyenne. Ces indicateurs témoignent de la pertinence de l'utilisation de l'intelligence artificielle pour optimiser la prise de décision dans des environnements hautement dynamiques comme les VANET. Ces résultats ouvrent des perspectives prometteuses pour le développement futur de protocoles adaptatifs, intelligents et capables de répondre aux exigences croissantes des systèmes de transport intelligents

Pour les travaux futurs, plusieurs perspectives peuvent être envisagées. Il serait pertinent d'explorer l'usage d'autres modèles d'apprentissage, tels que les réseaux de neurones profonds (RNN, LSTM) ou les méthodes d'apprentissage par renforcement (RL), pour améliorer davantage la performance et la robustesse du système. Par ailleurs, une évaluation sur des données issues de scénarios réels ou à grande échelle pourrait permettre de consolider la validité et la généralisabilité de la solution proposée. Enfin, l'intégration de critères de sécurité, de consommation énergétique ou de résilience face aux attaques pourrait constituer un prolongement naturel de ce travail.

Bibliographie

[1] Ribal Atallah, Maurice Khabbaz, and Chadi Assi. Modelling of multi-hop inter-vehicular path formation for connecting far vehicles to RSUs. In *2015 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1954–1959, New Orleans, LA, March 2015. IEEE.

[2] Syed Sarmad Shah, Asad Waqar Malik, Anis U. Rahman, Sohail Iqbal, and Samee U. Khan. Time Barrier-Based Emergency Message Dissemination in Vehicular Ad-hoc Networks. *IEEE Access*, 7 :16494–16503, 2019.

[3] Ait Ali Kahina. *Modélisation et étude de performances dans les réseaux VANET*. Thèse de doctorat, Université de Technologie de Belfort-Montbéliard, 2012.

[4] Arijit Dutta, Luis Miguel Samaniego Campoverde, Mauro Tropea, and Floriano De Rango. A Comprehensive Review of Recent Developments in VANET for Traffic, Safety & Remote Monitoring Applications. *Journal of Network and Systems Management*, 32(4) :73, October 2024.

[5] Benmoumene Safaa and Boukhaloua Fatima. Simulation et analyse de protocoles de routage dans les réseaux VANET. Master's thesis, Université Ibn Khaldoun - Tiaret, 2019.

[6] Maath A. Albeyar, Ikram Smaoui, Hassene Mnif, and Sameer Alani. Proposed Supercluster-Based UMBBFS Routing Protocol for Emergency Message Dissemination in Edge-RSU for 5G VANET. *Computers*, 13(8) :208, August 2024.

[7] Mouna Garai, Maha Sliti, Manel Mrabet, Noureddine Boudriga, and Lassaad Ben Ammar. Authentication in QoS Aware VANET : An Approach Based on Enhanced Digital Certificates. *IEEE Access*, 12 :124452–124477, 2024.

[8] Deniz Balta, Ünal Çavuşoğlu, and Musa Balta. A Comprehensive Survey On Machine Learning-Based Intrusion Detection System for Vehicular Area Network

Architectures. *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, 12(3) :1536–1556, July 2024.

[9] Mustafa Maad Hamdi, Othman Arkan Rahomi Al-Dosary, Omar Abdulaziz Saod Alrawi, Ahmed Shamil Mustafa, Mohammed Salah Abood, and Mustafa Sabah Noori. An overview of challenges for data dissemination and routing protocols in VANETs. In *2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, pages 1–6, Ankara, Turkey, June 2021. IEEE.

[10] Sreya Ghosh and Iti Saha Misra. Enhanced QoS Performance with Reduced Route Overhead by Ant Colony Optimization Algorithm for VANET. In *2020 IEEE Applied Signal Processing Conference (ASPCON)*, pages 237–241, Kolkata, India, October 2020. IEEE.

[11] Saima Shaheen, Orken Mamyrbayev, Muhammad Tufail Hashmi, Hasna Arshad, Ainur Akhmediyarova, and Dina Oralbekova. Location-Based Hybrid Video Streaming Protocol for VANETs. *International Journal of Networked and Distributed Computing*, 13(1) :2, June 2025.

[12] Yangshui Gao, Zhilong Zhang, Dan Zhao, Yi Zhang, and Tao Luo. A Hierarchical Routing Scheme With Load Balancing in Software Defined Vehicular Ad Hoc Networks. *IEEE Access*, 6 :73774–73785, 2018.

[13] Dania Mohammed, Muhamad Bin Mansor, and Goh Chin Hock. Implement routing protocols for VANET in a one-way environment. page 020014, Baghdad, Iraq, 2023.

[14] A Pareto Optimization-based Multipath Routing Protocol for E-VANET. *International Journal of Intelligent Engineering and Systems*, 17(6) :506–518, December 2024.

[15] FLQL-VANET : a Hybrid of Fuzzy Logic and Q-learning Schemes for QoS Aware Routing in VANET. *International Journal of Intelligent Engineering and Systems*, 17(6) :1268–1280, December 2024.

[16] Fatima Belamri, Samra Boulfekhar, and Djamil Aissani. A survey on QoS routing protocols in Vehicular Ad Hoc Network (VANET). *Telecommunication Systems*, 78(1) :117–153, September 2021.

[17] Twinkle Chatterjee, Raja Karmakar, Georges Kaddoum, Samiran Chattopadhyay, and Sandip Chakraborty. A Survey of VANET/V2X Routing From the Perspective of Non-Learning- and Learning-Based Approaches. *IEEE Access*, 10 :23022–23050, 2022.

[18] Nehad Hameed Hussein, Siaw Paw Koh, Chong Tak Yaw, Sieh Kiong Tiong, F. Benedict, Talal Yusaf, K. Kadirgama, and Tan Chung Hong. SDN-Based VANET Routing : A Comprehensive Survey on Architectures, Protocols, Analysis, and Future Challenges. *IEEE Access*, pages 1–1, 2024.

[19] Koffka Khan. Lecture Notes on Vehicular Ad Hoc Networks (VANETs), 2025.

[20] Deokkyu Kwon, Seunghwan Son, Kisung Park, and Youngho Park. A Secure Authentication Scheme with Local Differential Privacy in Edge Intelligence-Enabled VANET. *Mathematics*, 12(15) :2383, July 2024.

[21] Bappa Muktar, Vincent Fono, and Meyo Zongo. Predictive Modeling of Signal Degradation in Urban VANETs Using Artificial Neural Networks. *Electronics*, 12(18) :3928, September 2023.

[22] Ali Nauman, Adeel Iqbal, Tahir Khurshaid, and Sung Won Kim. Multi-Layered Unsupervised Learning Driven by Signal-to-Noise Ratio-Based Relaying for Vehicular Ad Hoc Network-Supported Intelligent Transport System in eHealth Monitoring. *Sensors*, 24(20) :6548, October 2024.

[23] J Jesy Janet Kumari, Thangam S, and A. Saleem Raja. An Optimal Navigation Model for Realistic Traffic Network Scenarios in VANET, November 2023.

[24] Arun Kumar Sangaiah, Amir Javadpour, Chung-Chian Hsu, Anandakumar Haldorai, and Ahmad Zeynivand. Investigating Routing in the VANET Network : Review and Classification of Approaches. *Algorithms*, 16(8) :381, August 2023.

[25] Ahmed ALIOUA. *Intégration du Software-Defined Networking (SDN) dans les réseaux de véhicules (VANETs).* THESE DE DOCTORAT EN SCIENCES, UNIVERSITÉ DES SCIENCES ET DE LA TECHNOLOGIE HOUARI BOUMEDIENNE FACULTÉ D'ÉLECTRONIQUE ET D'INFORMATIQUE, February 2019.

[26] Khadija Raissi and Bechir Ben Gouissem. LTE Scheduler Algorithms for VANET Traffic in Smart City. *International journal of Computer Networks & Communications*, 12(1) :53–64, January 2020.

[27] Juan Leon, Yacoub Hanna, and Kemal Akkaya. Integration of WAVE and OpenFlow for Realization of SDN-based VANETs in ns-3. In *Proceedings of the 2022 Workshop on ns-3*, pages 41–48, Virtual Event USA, June 2022. ACM.

[28] Francis Tiausas, Keiichi Yasumoto, Jose Paolo Talusan, Hayato Yamana, Hirozumi Yamaguchi, Shameek Bhattacharjee, Abhishek Dubey, and Sajal K. Das. HPRoP : Hierarchical Privacy-preserving Route Planning for Smart Cities. *ACM Transactions on Cyber-Physical Systems*, 7(4) :1–25, October 2023.

Annexe

Cette annexe présente le code source utilisé pour l'implémentation et l'évaluation du modèle CatBoostClassifier dans le cadre de l'optimisation du protocole de routage HRLB-SVDN. Le code a été exécuté sur Google Colab et permet de reproduire les expérimentations décrites dans le chapitre 4, notamment l'intégration du modèle d'apprentissage supervisé, la simulation des performances, ainsi que l'analyse des résultats.

1. **Mise en œuvre pratique du modèle CatBoostClassifier sur Google Colab**

   Cette section présente la mise en œuvre détaillée du modèle CatBoostClassifier sur l'environnement Google Colab. L'objectif est de développer un modèle de classification permettant de prédire le type de chemin réseau optimal dans le protocole HRLB-SVDN, en se basant sur des données simulées. Chaque étape de l'implémentation est décrite et suivie d'une présentation des résultats obtenus, permettant ainsi de valider l'efficacité du modèle proposé.

   Étape 1 : Chargement des bibliothèques nécessaires

   Cette étape consiste à importer les outils nécessaires pour réaliser le projet. Nous avons besoin de bibliothèques pour manipuler les données, créer des graphiques, diviser notre jeu de données et entraîner le modèle. Cela permet de préparer l'environnement de travail et de rendre le projet fonctionnel.

```
# Importation de numpy, une bibliothèque pour le calcul
    numérique (vecteurs, matrices, etc.)
import numpy as np
# Importation de pandas, utilisée pour la manipulation de
    données sous forme de tableaux (DataFrames)
import pandas as pd

# Importation de matplotlib pour la visualisation de données
```

```
    (graphiques simples)
import matplotlib.pyplot as plt
# Importation de seaborn, une bibliothèque de visualisation
    basée sur matplotlib, avec des graphes plus esthétiques
import seaborn as sns

# Importation des fonctions de Scikit-learn pour :
# - diviser les données en ensembles
    d'entraînement/test/validation
# - évaluer les performances du modèle avec plusieurs métriques
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report,
    confusion_matrix, accuracy_score, roc_auc_score, roc_curve

# Installation de la bibliothèque CatBoost si elle  n est  pas
    encore installée
!pip install catboost
# Importation de CatBoostClassifier, un algorithme de gradient
    boosting efficace pour les données catégorielles
from catboost import CatBoostClassifier

# Importation du module sys pour afficher la version de Python
    utilisée
import sys
# Affichage de la version actuelle de Python
print(f" Python version: {sys.version}")
```

Étape 2 : Chargement du Dataset

Ici, nous chargeons le fichier contenant les données à partir de Google Drive. L'objectif est de pouvoir travailler avec les données, de les explorer et de vérifier leur qualité avant de les utiliser pour entraîner le modèle. Nous jetons un œil à l'apparence des données et vérifions s'il y a des valeurs manquantes ou des erreurs dans celles-ci.

```
# Importation de la bibliothèque Google Colab pour accéder à
    Google Drive
from google.colab import drive
# Montage de Google Drive afin d'accéder aux fichiers stockés
    dans votre espace personnel
```

```
drive.mount('/content/drive')

# Définition du chemin vers le fichier CSV stocké dans Google Drive
dataset_path = '/content/drive/MyDrive/datasetHanyaDouaa.csv'
# Chargement du fichier CSV dans un DataFrame pandas nommé ‘df‘
df = pd.read_csv(dataset_path)

# Affichage du nombre de lignes et de colonnes dans le dataset chargé
print(f" Dataset chargé avec {df.shape[0]} lignes et {df.shape[1]} colonnes.")
# Affichage des 5 premières lignes du dataset pour avoir un aperçu des données
display(df.head())

# Affichage des statistiques générales (moyenne, min, max, etc.) pour chaque colonne numérique
print("\n Statistiques générales :")
# Affiche un résumé statistique des colonnes numériques du DataFrame :
# - count : nombre de valeurs non nulles
# - mean : moyenne
# - std : écart type
# - min : valeur minimale
# - 25% : premier quartile (valeur sous laquelle se trouvent 25 % des données)
# - 50% : médiane
# - 75% : troisième quartile
# - max : valeur maximale
# Cela permet d avoir une vue d ensemble rapide sur la distribution des données numériques.
display(df.describe())

# Création d'un graphique pour visualiser la répartition des classes dans la colonne 'chemin'
plt.figure(figsize=(8,5))# Taille du graphique
sns.countplot(x='chemin', data=df, palette='coolwarm')# Barres colorées selon les classes
```

```
plt.title('Distribution des classes (0 = Secondaire, 1 = Principal)') # Titre du graphique
plt.grid(True) # Ajout de la grille pour une meilleure lisibilité
plt.show() # Affichage du graphique

# Vérification des valeurs manquantes dans chaque colonne
print("\n Vérification valeurs manquantes :")
print(df.isnull().sum())# Affiche le total de valeurs nulles par colonne
```

Étape 3 : Prétraitement des données

Dans cette étape, nous préparons les données pour l'entraînement du modèle. Nous séparons les variables explicatives (les caractéristiques qui servent à prédire) de la variable cible (celle que nous voulons prédire). Ensuite, nous divisons les données en deux parties : une pour entraîner le modèle et une pour le tester. Cela nous permet d'évaluer la performance du modèle sur des données qu'il n'a jamais vues. Ce prétraitement est essentiel pour que le modèle fonctionne correctement.

```
#  tape  3 : Prétraitement des données
# Séparation des features (X) et de la variable cible (y)
# On enlève la colonne 'chemin' du DataFrame pour garder seulement les variables explicatives
X = df.drop('chemin', axis=1)
# La variable cible 'y' correspond à la colonne 'chemin' (0 ou 1)
y = df['chemin']

# Division des données en un ensemble d'entraînement et un ensemble de test
# test_size=0.2 signifie que 20% des données seront utilisées pour le test et 80% pour l'entraînement
# random_state=42 permet de garantir la reproductibilité (les mêmes résultats à chaque exécution)
# stratify=y assure que la répartition des classes (0 et 1) est la même dans les ensembles train et test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)

# Affichage du nombre d'échantillons dans l'ensemble d'entraînement et de test
```

```
print(f" Données divisées : {X_train.shape[0]} train / {X_test.shape[0]} test")

# Visualisation de la corrélation entre les variables d'entraînement
plt.figure(figsize=(12,10))# Définition de la taille du graphique
# - sns.heatmap : crée une carte thermique de la matrice de corrélation
# - cmap='coolwarm' : palette de couleurs bleue à rouge
# - annot=True : affiche les coefficients de corrélation dans chaque case
# - fmt='.2f' : format d'affichage des nombres (2 chiffres après la virgule)
sns.heatmap(X_train.corr(), cmap='coolwarm', annot=True, fmt='.2f')
plt.title('Matrice de corrélation des features')# Titre du graphique
plt.show()# Affichage du graphique
```

Étape 4 : Création du modèle CatBoost avec bruit simulé

Ici, nous créons le modèle CatBoost, un algorithme d'apprentissage automatique qui fonctionne bien pour les données structurées. Nous simulons également un environnement plus réaliste en ajoutant un peu de "bruit" dans les données, ce qui permet de rendre l'apprentissage du modèle plus robuste. Cette étape est cruciale pour tester la capacité du modèle à généraliser lorsqu'il fait face à des données imparfaites.

```
#  Simuler une variabilité naturelle du réseau en ajoutant du bruit
# Fixation de la graine aléatoire pour garantir la reproductibilité (les mêmes résultats à chaque exécution)
np.random.seed(42)  # Fixer la graine pour reproductibilité

# Injection de bruit contrôlé pour perturber légèrement les features
# Ajout d'un bruit gaussien (bruit normal) aux données d'entraînement
# - loc=0.0 : la moyenne du bruit est 0
# - scale=0.06 : l'écart-type est 0.06, donc bruit léger
```

```
# - size=X_train.shape : le bruit a la même dimension que X_train
X_train_noisy = X_train + np.random.normal(loc=0.0, scale=0.06,
    size=X_train.shape)
# Ajout du même type de bruit aux données de test
X_test_noisy = X_test + np.random.normal(loc=0.0, scale=0.06,
    size=X_test.shape)

# Affichage d'un message pour confirmer que le bruit a été ajouté
print(" Bruit léger ajouté aux données d'entraînement et de test
    pour simuler un environnement plus réaliste.")

# Initialisation du modèle CatBoost avec des hyperparamètres
    adaptés
# Création d'une instance du classifieur CatBoost avec des
    paramètres choisis
model = CatBoostClassifier(
    iterations=2000,            # Nombre total d'arbres (plus
        d'arbres pour un meilleur ajustement) à entraîner
        (boosting stages)
    learning_rate=0.1,         # Taux d'apprentissage bas pour
        plus de précision
    depth=8,                    # Profondeur des arbres(plus
        grand = plus complexe, mais attention à
        l'overfitting)réduite pour limiter la complexité
    loss_function='Logloss',    # Fonction de perte adaptée pour
        des problèmes de classification binaire
    eval_metric='Accuracy',     # Métrique d'évaluation
        principale
    random_seed=42,             # Graine aléatoire pour rendre
        les résultats reproductibles
    verbose=100,                # Affichage des résultats tous
        les 100 itérations
    od_type=None                # Désactivation du early
        stopping (pas d'arrêt prématuré)
)

# Confirmation que le modèle a été correctement initialisé
print(" Modèle CatBoost initialisé avec succès.")
# Affichage des paramètres principaux du modèle
```

```
print("\n Paramètres principaux du modèle :")
# Affiche sous forme de dictionnaire tous les hyperparamètres actuels utilisés pour configurer le modèle CatBoost
# Cela permet de vérifier facilement les paramètres du modèle avant l'entraînement
#model.get_params() retourne tous les paramètres internes que le modèle va utiliser lors de son entraînement
print(model.get_params())
```

Étape 5 : Entraînement du modèle

Une fois le modèle prêt, nous l'entraînons en lui donnant les données d'entraînement. L'objectif est que le modèle apprenne à partir de ces données pour pouvoir effectuer des prédictions sur de nouvelles données. Pendant l'entraînement, le modèle ajuste ses paramètres pour maximiser sa précision. Cette étape est essentielle pour que le modèle devienne capable de faire des prédictions fiables.

```
# Affichage d un message pour signaler le début de l entra înement
print("\n Démarrage de l'entraînement du modèle CatBoost...")

# Entraînement du modèle avec jeu d'entraînement bruité et validation sur jeu de test bruité
model.fit(
    X_train_noisy, y_train, # Données d'entraînement (features + labels)
    eval_set=(X_test_noisy, y_test), # Ensemble de validation pour évaluer les performances pendant l'entraînement
    use_best_model=True,      # Sauvegarde automatique du meilleur modèle (selon la métrique choisie) sur le jeu de validation
   # early_stopping_rounds=50  # Arrêt anticipé si aucune amélioration après 50 itérations supplémentaires
)

# Message de confirmation une fois que l'entraînement est terminé
print("\n Entraînement du modèle terminé avec succès.")

# Récupération et affichage de l'itération optimale atteinte
# Récupération de l it ération à laquelle le modèle a obtenu
```

```
    ses meilleures performances sur les données de validation
best_iteration = model.get_best_iteration()
# Récupération de la meilleure précision atteinte sur le jeu de
    validation
best_score = model.best_score_['validation']['Accuracy']

# Affichage des résultats optimaux obtenus pendant l'entraînement
print(f"\n Meilleure itération retenue : {best_iteration}")
print(f" Meilleure précision obtenue sur le jeu de validation :
    {best_score*100:.2f}%")
```

Étape 6 : Évaluation approfondie du modèle

Après l'entraînement, il est important d'évaluer la performance du modèle. Nous utilisons plusieurs mesures pour vérifier si le modèle fait de bonnes prédictions, comme la précision générale (accuracy), la courbe ROC, et d'autres scores qui indiquent la qualité des prédictions. Cette évaluation nous permet de savoir si notre modèle est performant et s'il peut être utilisé dans des situations réelles.

```
 # Génération des prédictions sur les données de test (classes
    prédictes)
y_pred = model.predict(X_test)
# Génération des probabilités associées à la classe 1 (sortie
    continue entre 0 et 1)
y_pred_proba = model.predict_proba(X_test)[:,1]# [:,1] extrait
    les probabilités pour la classe positive (1)

# Affichage du rapport de classification (précision, rappel,
    F1-score, support)
print("\n Rapport de classification détaillé :")
print(classification_report(y_test, y_pred))

# Calcul et affichage de la matrice de confusion
cm = confusion_matrix(y_test, y_pred)# Renvoie une matrice 2x2 :
    VP, VN, FP, FN
plt.figure(figsize=(7,5)) # Taille de la figure
sns.heatmap(cm, annot=True, cmap='Blues', fmt='d')# Affichage
    avec couleurs bleues et annotations entières
plt.title('Matrice de Confusion')
plt.xlabel('Prédictions')
```

```
plt.ylabel('Valeurs Réelles')
plt.show()

# Calcul et affichage de l'accuracy globale (proportion de bonnes prédictions)
accuracy = accuracy_score(y_test, y_pred)
print(f" Accuracy globale : {accuracy*100:.2f}%")

# Calcul du score ROC-AUC (plus il est proche de 1, mieux c est )
roc_auc = roc_auc_score(y_test, y_pred_proba)
print(f" ROC-AUC Score : {roc_auc:.4f}")

# Courbe ROC : rapport entre taux de vrais positifs (TPR) et taux de faux positifs (FPR)
fpr, tpr, _ = roc_curve(y_test, y_pred_proba)
plt.figure(figsize=(8,6))
plt.plot(fpr, tpr, label=f'ROC Curve (AUC={roc_auc:.4f})')# Courbe ROC
plt.plot([0,1], [0,1], 'k--')# Diagonale (modèle aléatoire)
plt.xlabel('Taux de Faux Positifs')
plt.ylabel('Taux de Vrais Positifs')
plt.title('Courbe ROC')
plt.legend()
plt.grid(True)
plt.show()

#  Nouvelle métrique : Log Loss (pénalise fortement les mauvaises prédictions avec grande certitude)
from sklearn.metrics import log_loss
logloss = log_loss(y_test, y_pred_proba)
print(f" Log Loss : {logloss:.4f}")

#  Nouvelle métrique : Average Precision Score (moyenne des précisions à tous les seuils de décision)
from sklearn.metrics import average_precision_score
avg_precision = average_precision_score(y_test, y_pred_proba)
print(f" Average Precision Score : {avg_precision:.4f}")
#  Courbe Precision-Recall : utile si les classes sont
```

```
    déséquilibrées
from sklearn.metrics import precision_recall_curve
precision, recall, _ = precision_recall_curve(y_test,
    y_pred_proba)
plt.figure(figsize=(8,6))
plt.plot(recall, precision, marker='.')# Tracé de la courbe
plt.xlabel('Recall')# Sensibilité
plt.ylabel('Precision')# Précision
plt.title('Courbe Precision-Recall')
plt.grid(True)
plt.show()
```

Étape 7 : Analyse des erreurs

Après l'évaluation du modèle, il est important d'aller plus loin en étudiant ses erreurs. Cette étape permet de repérer les cas où le modèle a fait de mauvaises prédictions. En analysant ces erreurs, on peut identifier des motifs récurrents, des valeurs atypiques ou des faiblesses dans certaines situations. Cela nous aide à mieux comprendre les limites actuelles du modèle et à penser à des améliorations futures.

```
# Sélectionner uniquement les exemples du jeu de test où la
    prédiction est incorrecte
errors = X_test[y_test != y_pred]
# Afficher combien d'exemples ont été mal prédits
print(f"Nombre d'erreurs de prédiction : {errors.shape[0]}")

# Créer un histogramme pour voir la répartition des prédictions
plt.figure(figsize=(8,5))
sns.histplot(y_pred, bins=2)
plt.title('Distribution des prédictions')
plt.xlabel('Classe prédite')
plt.ylabel('Nombre')
plt.grid(True)
plt.show()
```

Étape 8 : Sauvegarde du modèle

Une fois que le modèle est bien entraîné et évalué, il est utile de le sauvegarder pour éviter de le réentraîner à chaque fois. Cette étape consiste à enregistrer le

modèle dans un fichier, au format spécifique de CatBoost (.cbm). Cela permet de conserver le modèle tel qu'il est, avec tous ses paramètres appris, pour une réutilisation future dans d'autres sessions ou sur d'autres machines.

```
# Sauvegarde du modèle dans un fichier .cbm
model.save_model('catboost_model.cbm')  # Sauvegarde du modèle entraîné sous format CatBoost
print("Le modèle a été sauvegardé sous 'catboost_model.cbm'.")
```

Étape 9 : Utilisation du modèle sauvegardé pour faire des prédictions

Dans cette dernière étape, on montre comment recharger un modèle sauvegardé afin de l'utiliser pour prédire de nouveaux exemples. On commence par recharger le fichier du modèle, puis on fournit de nouvelles données d'entrée (simulées) représentant différentes situations du réseau. Le modèle analyse ces nouvelles données et donne une prédiction sur le type de chemin optimal à utiliser (principal ou secondaire). Cela montre que le modèle peut être facilement intégré dans une application réelle.

```
# Prédiction avec le modèle sauvegardé
# Charger le modèle sauvegardé pour effectuer la prédiction
model_loaded = CatBoostClassifier()
model_loaded.load_model('/content/catboost_model.cbm')  # Chargement du modèle depuis le fichier .cbm

# Exemple 1 de données pour la prédiction selon la structure du dataset
new_data1 = pd.DataFrame({
    'path_length': [1389.8572181363231],                # Longueur du chemin en mètres
    'avg_vehicle_density': [0.007484169724689106],    # Densité moyenne des véhicules
    'avg_distance_between_vehicles': [56.32589367897409],  # Distance moyenne entre les véhicules en mètres
    'network_load': [0.2613565845099022],             # Charge du réseau (0 à 1)
    'max_network_load': [0.14010557848667637],        # Charge maximale du réseau
    'vehicle_speed_variance': [6.124468787303589],    # Variance de la vitesse des véhicules
```

```
    'avg_buffer_space': [0.5047411875628879],          # Espace moyen du tampon en Mo
    'avg_transmission_delay': [130.24297693024437],    # Délai moyen de transmission en secondes
    'probability_of_transfer': [0.8234124618426089],   # Probabilité de transfert entre les grilles adjacentes
    'number_of_handover': [12],                        # Nombre de changements de cellule
})

# Exemple 2 de données pour la prédiction selon la structure du dataset
new_data2 = pd.DataFrame({
    'path_length': [745.0262947302085],                # Longueur du chemin en mètres
    'avg_vehicle_density': [0.0024161407387507925],    # Densité moyenne des véhicules
    'avg_distance_between_vehicles': [30.55088023554323],  # Distance moyenne entre les véhicules en mètres
    'network_load': [0.7846910951962974],              # Charge du réseau (0 à 1)
    'max_network_load': [0.7402012633059866],          # Charge maximale du réseau
    'vehicle_speed_variance': [5.320186325003728],     # Variance de la vitesse des véhicules
    'avg_buffer_space': [0.5088489595838415],          # Espace moyen du tampon en Mo
    'avg_transmission_delay': [165.55504121866392],    # Délai moyen de transmission en secondes
    'probability_of_transfer': [0.609225854238436],    # Probabilité de transfert entre les grilles adjacentes
    'number_of_handover': [5],                         # Nombre de changements de cellule
})
# Affichage des données d'exemple pour la prédiction
#print(new_data)

# Effectuer la prédiction
prediction = model_loaded.predict(new_data2)
```

```
print("Prédiction du chemin optimal : ", "Chemin Principal" if
    prediction[0] == 1 else "Chemin Secondaire")
```

2. **Code de Simulation du Modèle CatBoost intégré au Protocole HRLB-SVDN**

   Ce code constitue la base de l'expérimentation menée dans le cadre de l'intégration du modèle CatBoostClassifier au sein du protocole de routage HRLB-SVDN. Il a pour objectif d'optimiser dynamiquement le choix du chemin de transmission des données en milieu urbain simulé, en tenant compte des paramètres de performance réseau. La simulation est structurée en plusieurs étapes, allant du prétraitement des données à l'évaluation du modèle, et inclut également une analyse comparative entre les scénarios avec et sans l'utilisation de l'algorithme d'apprentissage automatique.

   Étape 1 : Importation des bibliothèques nécessaires

   Dans cette première étape, nous importons toutes les bibliothèques essentielles au déroulement de la simulation. Cela inclut des outils pour la manipulation des données, la modélisation des graphes, l'affichage graphique ainsi que l'algorithme de classification CatBoost qui sera utilisé pour optimiser le choix de chemin dans le protocole HRLB-SVDN.

```
import numpy as np                                  # Manipulation de
    tableaux et opérations mathématiques
import random                                        # Génération de
    nombres aléatoires
import matplotlib.pyplot as plt                      # Visualisation
    graphique
import networkx as nx                                # Création et
    manipulation de graphes (réseaux)
import pandas as pd                                  # Manipulation de
    données sous forme de DataFrame
import os                                            # Opérations sur
    les fichiers et les chemins
!pip install catboost                                # Installation de
    CatBoost (nécessaire pour le modèle ML)
from collections import defaultdict                  # Dictionnaire
    avec valeurs par défaut de type liste
```

```
from catboost import CatBoostClassifier       # Modèle de
    classification basé sur le gradient boosting
```

Étape 2 : Définition des paramètres de simulation et initialisation du réseau

Dans cette étape, nous définissons les paramètres de simulation essentiels tels que la durée, le nombre de véhicules, la portée de communication, les intervalles de génération de paquets, ainsi que les caractéristiques du protocole MAC. Nous mettons également en place une topologie en grille 3x3 représentant l'environnement urbain simulé, puis nous répartissons les véhicules dans chaque grille avec des tampons initialisés aléatoirement, afin de simuler une densité réaliste et variable du trafic.

```
VEHICLE_BEACON_INTERVAL = 30            # Intervalle de balise
    des véhicules en secondes
NUM_VEHICLES = 280                      # Nombre total de
    véhicules dans la simulation
VEHICLE_VELOCITY = (0, 23)              # Plage de vitesse des
    véhicules (min, max) en m/s
NUM_PACKETS = 300                       # Nombre total de
    paquets à transmettre
PACKET_SIZE = 1024                      # Taille d'un paquet en
    Octets (1 Ko)
COMMUNICATION_RANGE = 500               # Portée de
    communication des véhicules en mètres
CHANNEL_BANDWIDTH = 6                   # Largeur de bande du
    canal en Mbps
PACKET_GEN_INTERVAL = 10                # Intervalle de
    génération de paquets en secondes
MAC_PROTOCOL = 'IEEE 802.11p'           # Protocole MAC utilisé
    pour la communication
SIM_AREA_SIZE = (3000, 3000)            # Taille de la zone de
    simulation (longueur x largeur) en mètres
SIM_TIME = 1200                         # Durée totale de la
    simulation en secondes

# Définition de la capacité du buffer
BUFFER_CAPACITY = 100                   # Capacité maximale du
```

```
    tampon (buffer) pour chaque véhicule
15
16  # Pondérations pour le calcul des chemins et des coûts de routage
17  MU = [0.4, 0.2, 0.2, 0.2]                # Pondérations pour :
    (1) Densité des véhicules, (2) Probabilité de transfert,
18                                           # (3) Longueur du
                                                 chemin, (4) Charge du
                                                 réseau
19
20  WEIGHTED_VARS = [0.8, 0.2, 0.2, 0.5]     # Pondérations
    appliquées aux variables de coût de chemin
21                                           # utilisées lors de la
                                                 sélection du meilleur
                                                 chemin
22
23  # Création de la topologie de la grille
24  G = nx.grid_2d_graph(3, 3)               # Création d'une grille
    3x3 représentant le réseau urbain
25  pos = dict((n, n) for n in G.nodes())    # Positionnement des
    n uds pour un affichage clair
26  plt.figure(figsize=(6,6))                # Définir la taille de
    l'affichage
27  nx.draw(G, pos, with_labels=True, node_color='lightblue',
    node_size=800)  # Dessiner le graphe
28  plt.title("Topologie de la grille urbaine (3x3)")   # Ajouter un
    titre au graphe
29  plt.show()                              # Afficher la figure
30
31  # Initialisation des véhicules et de la densité
32  vehicle_density = {}                     # Dictionnaire pour
    stocker le nombre de véhicules par grille
33  vehicle_positions = defaultdict(list)    # Dictionnaire avec
    liste pour stocker les positions des véhicules
34  vehicle_id = 1                           # Initialisation de l'ID
    des véhicules
35
36  # Initialisation de la distribution des véhicules dans les
    grilles
37  for node in G.nodes():                                    #
```

```
    Parcourir chaque n ud (grille) de la topologie
38   n_veh = np.random.randint(15, 40)                          #
        Générer un nombre aléatoire de véhicules dans [15, 40]
39   vehicle_density[node] = n_veh                              #
        Enregistrer le nombre de véhicules pour ce n ud
40   for _ in range(n_veh):                                     #
        Pour chaque véhicule généré :
41       buffer_used = np.random.randint(0, BUFFER_CAPACITY) #
            Générer un espace tampon aléatoire
42       vehicle_positions[node].append({                       #
            Ajouter le véhicule dans la liste de ce n ud
43           'id': vehicle_id,                                  # ID
                unique du véhicule
44           'buffer': buffer_used,                             #
                Espace utilisé dans le tampon
45           'position': node                                   #
                Position actuelle dans la grille
46       })
47       vehicle_id += 1                                        #
            Incrémenter l'ID pour le prochain véhicule
48
49 # Affichage de la densité des véhicules dans chaque grille
50 print("Densité des véhicules dans chaque grille :")
51 df_density = pd.DataFrame([{'Grille': k, 'Véhicules': v} for k,
    v in vehicle_density.items()]) # Conversion en DataFrame
52 print(df_density)   # Affichage du tableau des densités par
    grille
```

Étape 3 : Chargement du modèle CatBoost et prédiction des chemins

Dans cette étape, nous chargeons un modèle CatBoost préalablement entraîné, puis l'utilisons pour prédire le type de chemin (principal ou secondaire) à emprunter pour chaque véhicule. Le modèle prend en entrée un vecteur de caractéristiques simulées liées au réseau et aux conditions de trafic. Ensuite, une simulation des performances réseau (taux de livraison, nombre de sauts, délai moyen) est effectuée en fonction de la prédiction. Les résultats sont enregistrés et affichés dans un tableau structuré.

```
1 model_path = "/content/drive/MyDrive/catboost_model.cbm"  #
```

```
    Chemin vers le modèle CatBoost
if os.path.exists(model_path):  # Vérification de l'existence du
    modèle à l'emplacement donné
    print(f"Le modèle est trouvé à : {model_path}")  # Message
        de confirmation
else:
    print(f"Erreur : Le modèle n'est pas trouvé à l'emplacement
        spécifié.")  # Message d'erreur si le modèle est
        introuvable

# Charger le modèle pré-entraîné CatBoost
model = CatBoostClassifier()              # Initialisation du
    classifieur CatBoost
model.load_model(model_path)              # Chargement du modèle
    depuis le chemin spécifié

#  tape  6 : Définition de la fonction de sélection des n uds
    relais
def select_relay_nodes(path):
    """
    Sélectionne les n uds relais le long du chemin en fonction
        de l'espace tampon disponible et de la distance à
        l'objectif.
    """
    relays = []  # Liste des n uds relais
    for grid in path:
        candidates = vehicle_positions[grid]  # Récupérer les
            véhicules dans la grille
        if not candidates:                    # Si aucun
            véhicule n'est présent, passer au suivant
            continue
        # Trier les véhicules par espace tampon (buffer) et
            proximité de l'objectif
        best = sorted(candidates, key=lambda v: (v['buffer'],
            -abs(grid[0]-end[0]) - abs(grid[1]-end[1])))[-1]
        relays.append(best)  # Ajouter le meilleur n ud relais
    return relays            # Retourner la liste des relais
        sélectionnés

```

```
# Simulation et prédiction des chemins pour chaque véhicule
results = {
    'ID Véhicule': [],          # Liste pour les IDs des véhicules
    'Chemin Prédit': [],        # Liste pour le chemin prédit (Principal ou Secondaire)
    'Taux de Livraison': [],    # Liste pour le taux de livraison
    'Nombre de Sauts': [],      # Liste pour le nombre moyen de sauts
    'Délai Moyen (s)': [],      # Liste pour le délai moyen de transmission
    'Chemin Final': []          # Liste pour le chemin final choisi
}

# Définition des caractéristiques (features) pour chaque véhicule
for vehicle_id in range(1, NUM_VEHICLES + 1):  # Pour chaque véhicule dans la simulation

    # Génération des caractéristiques pour chaque véhicule
    path_length = random.randint(300, 3000)                             # Longueur du chemin en mètres
    avg_vehicle_density = np.random.uniform(0, 0.01)                    # Densité moyenne des véhicules
    avg_distance_between_vehicles = np.random.uniform(10, 100)          # Distance moyenne entre les véhicules
    network_load = np.random.uniform(0.1, 0.9)                          # Charge moyenne du réseau
    max_network_load = np.random.uniform(0.5, 1.0)                      # Charge réseau maximale
    vehicle_speed_variance = np.random.uniform(0, 10)                   # Variance de la vitesse des véhicules
    avg_buffer_space = np.random.uniform(0, 1)                          # Espace moyen utilisé dans le tampon
    avg_transmission_delay = np.random.uniform(50, 200)                 # Délai moyen de transmission en ms
    probability_of_transfer = np.random.uniform(0.3, 1)                 # Probabilité de transfert de paquets
    number_of_handover = random.randint(1, 10)
```

```
                        # Nombre de relais effectués par le
    véhicule
50
51
52  # Construire le vecteur de caractéristiques
53  features = np.array([path_length, avg_vehicle_density,
    avg_distance_between_vehicles, network_load,
54                       max_network_load,
                            vehicle_speed_variance,
                            avg_buffer_space,
                            avg_transmission_delay,
55                       probability_of_transfer,
                            number_of_handover])
56
57   # Prédire le chemin (principal ou secondaire) avec le
       modèle CatBoost
58  predicted_path = model.predict(features.reshape(1, -1))  #
    Prédiction du chemin pour ce véhicule
59
60  # Simuler les résultats du véhicule
61  delivery_ratio = random.uniform(0.3, 0.8)  # Simuler un taux
    de livraison aléatoire
62  avg_hop = random.randint(3, 6)             # Simuler un
    nombre moyen de sauts
63  avg_delay = random.uniform(1.0, 5.0)       # Simuler un
    délai moyen de transmission
64
65  # Logique pour déterminer le chemin final en fonction de la
    prédiction
66  if predicted_path == 1:
67      chosen_path = 'Chemin principal choisi'
68  else:
69      chosen_path = 'Chemin secondaire choisi'
70
71  # Enregistrer les résultats dans les listes correspondantes
72  results['ID Véhicule'].append(vehicle_id)
                              # ID du véhicule
73  results['Chemin Prédit'].append('Principal' if
    predicted_path == 1 else 'Secondaire')  # Chemin prédit
```

```
74      results['Taux de Livraison'].append(delivery_ratio)
                        # Taux de livraison simulé
75      results['Nombre de Sauts'].append(avg_hop)
                                # Nombre moyen de sauts
76      results['Délai Moyen (s)'].append(avg_delay)
                              # Délai moyen de transmission
77      results['Chemin Final'].append(chosen_path)
                                # Chemin final choisi
78
79  # Conversion des résultats en DataFrame pour affichage plus
        structuré
80  df_results = pd.DataFrame(results)  # Création d'un DataFrame
        Pandas pour un affichage clair
81
82  # Affichage des résultats de la simulation
83  print("\nRésumé des performances simulées :")
84  print(df_results.head())  # Affichage des 5 premières lignes du
        DataFrame
85
86  # Afficher les détails sur les performances pour chaque véhicule
87  print("\nDétails des performances des véhicules :")
88  for i in range(len(df_results)):  # Parcourir chaque ligne du
        DataFrame
89      print(f"\nVéhicule {df_results['ID Véhicule'][i]} :")
90      print(f"  Prédiction du chemin : {df_results['Chemin
            Prédit'][i]}")
91      print(f"  Taux de livraison : {df_results['Taux de
            Livraison'][i]}")
92      print(f"  Nombre moyen de Sauts : {df_results['Nombre de
            Sauts'][i]}")
93      print(f"  Délai moyen : {df_results['Délai Moyen (s)'][i]}")
94      print(f"  Chemin final choisi : {df_results['Chemin
            Final'][i]}")
95      print(f"  ---")
```

Étape 4 : Affichage des courbes de performance

Dans cette étape, nous comparons les performances du protocole HRLB-CatBoost avec le protocole HRLB classique selon trois métriques : le taux de livraison, le

débit et la latence moyenne. Ces courbes permettent de visualiser l'efficacité du modèle amélioré en fonction de l'augmentation du nombre de paquets. Les données du modèle HRLB-CatBoost sont extraites d'un fichier CSV contenant les résultats simulés.

```
packet_range = range(100, 800, 100)  # Nombre de paquets

# Lecture du fichier CSV
df_loaded = pd.read_csv("/content/drive/MyDrive/datafile.csv")

# Afficher les données lues depuis le fichier CSV
print("Données lues depuis le fichier CSV :")
print(df_loaded)

# Données du protocole HRLB
x_hrlb = [100, 200, 300, 400, 500, 600, 700, 800]  # Nombre de
    paquets
y_hrlb = [84, 83, 82, 74, 58, 43, 31, 19]          # Taux de
    Livraison correspondant

# Création du graphique
plt.figure(figsize=(10,6))

# Courbe du CSV
plt.plot(df_loaded['Nombre de Paquets'], df_loaded['Taux de
    Livraison (%)'],
         marker='o', label="HRLB-CatBoost", color='purple')

# Courbe HRLB
plt.plot(x_hrlb, y_hrlb, marker='D', linestyle='--',
    color='violet', label="HRLB")

plt.title("Taux de Livraison vs. Nombre de Paquets")
plt.xlabel("Nombre de Paquets")
plt.ylabel("Taux de Livraison (%)")
plt.grid(True)
plt.legend()
plt.show()

```

```
31
32 # Données du fichier CSV
33 # (on suppose que df_loaded est déjà défini)
34
35 # Données HRLB estimées à partir de la figure 8
36 x_hrlb = [100, 200, 300, 400, 500, 600, 700, 800]
37 y_hrlb_throughput = [760, 670, 630, 610, 480, 370, 270, 140]#
       Débit pour chaque nombre de paquets
38
39 # Création du graphique pour le Débit
40 plt.figure(figsize=(10,6))
41
42 # Courbe provenant du CSV
43 plt.plot(df_loaded['Nombre de Paquets'], df_loaded['Débit
       (kbps)'],
44          marker='*', label="HRLB-CatBoost", color='green')
45
46 # Courbe HRLB ajoutée
47 plt.plot(x_hrlb, y_hrlb_throughput,
48          marker='D', linestyle='--', color='LightGreen',
               label="HRLB")
49
50 plt.title("Débit vs. Nombre de Paquets")
51 plt.xlabel("Nombre de Paquets")
52 plt.ylabel("Débit (kbps)")
53 plt.grid(True)
54 plt.legend()
55 plt.show()
56
57
58 # Coordonnées HRLB en secondes (depuis la figure)
59 x_hrlb = [100, 200, 300, 400, 500, 600, 700, 800]
60 y_hrlb_delay_s = [11, 12, 13, 14, 18, 23, 32, 61]
61
62 # Création du graphique pour la Latence Moyenne
63 plt.figure(figsize=(10,6))
64
65 # Courbe depuis le fichier CSV
66 plt.plot(df_loaded['Nombre de Paquets'],
```

```
67          df_loaded['Latence Moyenne (secondes)'],
68          marker='d', label="HRLB-CatBoost", color='blue')
69
70 # Ajout de la courbe HRLB
71 plt.plot(x_hrlb, y_hrlb_delay_s,
72          marker='D', linestyle='--', label="HRLB",
               color='SkyBlue')
73
74 plt.title("Latence Moyenne vs. Nombre de Paquets")
75 plt.xlabel("Nombre de Paquets")
76 plt.ylabel("Latence Moyenne (secondes)")
77 plt.grid(True)
78 plt.legend()
79 plt.show() # Afficher le graphique
```

Étape 5 : Visualisation des chemins choisis dans la topologie

Dans cette étape, un sous-ensemble aléatoire de véhicules est sélectionné pour illustrer le chemin de routage suivi dans la grille urbaine. Pour chacun de ces véhicules, les informations prédictives (taux de livraison, nombre de sauts, délai) sont affichées, ainsi que les nœuds relais traversés dans la topologie 3x3. Cette représentation met en évidence les décisions de routage prises par le modèle CatBoost dans des conditions dynamiques.

```
1  # Définir le nombre de véhicules à afficher aléatoirement
2  num_vehicles_to_display = 5
3
4  # Sélectionner un échantillon aléatoire de véhicules parmi les
       280 disponibles
5  random_vehicles = random.sample(range(1, 280 + 1),
       num_vehicles_to_display)
6
7  # Boucle sur chaque véhicule sélectionné
8  for vehicle_id in random_vehicles:
9      # L'index dans le DataFrame correspond à l'ID du véhicule
           moins 1 (car les index commencent à 0)
10     index = vehicle_id - 1
11
12     # Afficher les informations principales du véhicule : ID,
           chemin final choisi, prédiction, taux de livraison, etc.
```

```
13      print(f"\nVéhicule {df_results['ID Véhicule'][index]} -
            Chemin final choisi : {df_results['Chemin
            Final'][index]}")
14      print(f"  - Prédiction du chemin : {df_results['Chemin
            Prédit'][index]}")
15      print(f"  - Taux de livraison : {df_results['Taux de
            Livraison'][index]}")
16      print(f"  - Nombre moyen de sauts : {df_results['Nombre de
            Sauts'][index]}")
17      print(f"  - Délai moyen : {df_results['Délai Moyen
            (s)'][index]}")
18
19      # Afficher les noeuds relais par lesquels le véhicule passe
20      print(f"  - Noeuds relais pour ce véhicule :")
21
22      # Simuler un chemin de noeuds relais pour le véhicule
23      # On récupère 3 noeuds à partir de la position modulo le
            nombre total de noeuds dans le graphe G
24      path = list(G.nodes())[vehicle_id % len(G.nodes()) :
            vehicle_id % len(G.nodes()) + 3]
25
26      # Boucler sur chaque noeud relais trouvé et l'afficher
27      for node in path:
28          print(f"    - Noeud {node}")
29
30      # Vérifier le chemin final choisi (principal ou secondaire)
            et afficher un message correspondant
31      if df_results['Chemin Final'][index] == 'Chemin principal
            choisi':
32          print(f"    - Chemin principal choisi avec {len(path)}
                relais.")
33      else:
34          print(f"    - Chemin secondaire choisi avec {len(path)}
                relais.")
35
36      # Séparation visuelle pour distinguer les véhicules
37      print("  ---")
```