



N° d'ordre :
Série :



الجمهورية الجزائرية الديمقراطية الشعبية

Republique Algerienne Democratique Et Populaire

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة محمد خيضر بسكرة

Université Mohamed Kheider –Biskra

كلية العلوم الدقيقة وعلوم الطبيعة والحياة

Faculté Des Sciences Exactes Et Des Sciences De La Nature Et De La Vie

قسم الإعلام الآلي

Département D'informatique

Thèse

Présentée pour l'obtention du grade de docteur en sciences en informatique

Spécialité : Informatique

Par : Zeroual Djazia

Sujet :

Une approche basée réseaux de capteurs pour
l'internet des objets au smart city

Soutenue publiquement le : 10 /06/2025 - devant le jury composé de :

Président : Labib Sadek Terrissa

Prof. Université de Biskra

Rapporteur : Okba Kazar

Prof. Ass Université de Sharjah

Membres:

Saber Benharzallah

Prof. Université de Batna

Rachid Seghir

Prof. Université de Batna

Soheyb Ayad

M.C.A Université de Biskra

Samir Bourekkache

M.C.A Université de Biskra

Année universitaire : 2024 – 2025



الجمهورية الجزائرية الديمقراطية الشعبية

People's Democratic Republic of Algeria

وزارة التعليم العالي والبحث العلمي

Ministry of Higher Education and Scientific Research

جامعة محمد خيضر بسكرة

Mohamed Khider University - Biskra

كلية العلوم الدقيقة وعلوم الطبيعة والحياة

Faculty of Exact Sciences and Sciences of Nature and Life

قسم الإعلام الآلي

Computer Science Department

THESIS

In candidacy for the degree of Doctor Science

Presented by: Zeroual Djazia

TITLE

**A sensor network-based approach for the
Internet of Things in the smartcity**

Defended on: 10 / 06 /2025 - In front of the jury composed of:

President: Labib Sadek Terrissa

Prof. University of Biskra

Director of thesis: Okba Kazar

Prof. Ass University of Sharjah

Members:

Saber Benharzallah

Prof. University of Batna

Rachid Seghir

Prof. University of Batna

Soheyb Ayad

M.C.A University of Biskra

Samir Bourekkache

M.C.A University of Biskra

Academic year: 2024 – 2025

Abstract

The emerging technologies like Information and communications technology (ICT), Artificial Intelligence (AI) and Internet of Things (IOT) have a huge influence on the development of smart city, which improves the daily life of residents. The intelligent transportation system (ITS) is one of the main requirements of a smart city. The application of machine-learning (ML) technology in the development of driver assistance systems, has improved the safety and the comfort of the experience of traveling by road. In this work, we propose the development of an intelligent driving system for road accident risks prediction that can extract maximum required information to alert the driver in order to avoid risky situations that may cause traffic accidents. The current acceptable Internet-of-vehicle (IOV) solutions rely heavily on the cloud, as it has virtually unlimited storage and processing power. However, the Internet disconnection problem and response time are constraining its use. In this case, the concept of vehicular edge computing (V.Edge.C) can overcome these limitations by leveraging the processing and storage capabilities of simple resources located closer to the end user, such as vehicles or roadside infrastructure. In this thesis, we propose an Intelligent and Collaborative Cloud-V.Edge Driver Assistance System (ICEDAS) framework based on machine learning to predict the risks of traffic accidents. The proposed framework consists of two models, CLOUD_DRL and V.Edge_DL, Each one complements the other, together, these models work to enhance the effectiveness and accuracy of crash prediction and prevention. The obtained results show that our system efficient and it can help to reduce road accidents and save thousands of citizens' lives.

Keywords: *IOV, Deep Learning, Deep Reinforcement Learning, Cloud Computing, V.Edge Computing, Cloud-V.Edge Collaboration.*

Résumé

Les technologies émergentes telles que les technologies de l'information et de la communication (TIC), l'intelligence artificielle (IA) et l'Internet des objets (IoT) ont une influence considérable sur le développement des villes intelligentes, ce qui améliore la vie quotidienne des résidents. Le système de transport intelligent (ITS) est l'une des principales exigences d'une ville intelligente. L'application de la technologie d'apprentissage automatique (ML) dans le développement de système d'assistance à la conduite a amélioré la sécurité et le confort de l'expérience du voyage sur la route. Dans ce travail, nous proposons le développement d'un système de conduite intelligent pour la prédiction des risques d'accident routier qui peut extraire les informations nécessaires au maximum pour alerter le conducteur afin d'éviter les situations risquées pouvant causer des accidents de la route. Les solutions actuellement acceptables d'Internet des véhicules (IOV) reposent fortement sur le cloud, car il dispose d'un stockage et d'une puissance de traitement pratiquement illimitée. Cependant, le problème de déconnexion d'Internet et le temps de réponse limitent son utilisation. Dans ce cas, le concept de calcul en périphérie des véhicules (V.Edge.C) peut surmonter ces limitations en exploitant les capacités de traitement et de stockage de ressources simples situées plus près de l'utilisateur final, telles que les véhicules ou les infrastructures routières. Dans cette thèse, nous proposons un système d'assistance à la conduite intelligent et collaboratif (ICEDAS) basé sur l'apprentissage automatique pour prédire les risques d'accident de la circulation. Le système proposé se compose de deux modèles, Cloud_DRL et V.Edge_DL, chacun complétant l'autre. Ensemble, ces modèles travaillent à améliorer l'efficacité et la précision de la prédiction et de la prévention des accidents. Les résultats obtenus montrent que notre système est efficace et peut aider à réduire les accidents de la route et à sauver des milliers de vies de citoyens.

Mots-clés : *IOV, Apprentissage profond, Apprentissage par renforcement profond, Informatique en nuage, Informatique en Périphérie, Collaboration nuage - Périphérie.*

الملخص

إن التقنيات الناشئة مثل تكنولوجيا المعلومات والاتصالات، والذكاء الاصطناعي وإنترنت الأشياء لها تأثير كبير على تطوير المدينة الذكية، مما يحسن الحياة اليومية للسكان. يعد نظام النقل الذكي أحد المتطلبات الرئيسية للمدينة الذكية. أدى تطبيق تقنية التعلم الآلي في تطوير أنظمة مساعدة السائق إلى تحسين السلامة والراحة في تجربة السفر على الطرق. في هذا العمل نقترح تطوير نظام قيادة ذكي للتنبؤ بمخاطر حوادث الطرق يمكنه استخلاص الحد الأقصى من المعلومات المطلوبة لتنبيه السائق لتجنب المواقف الخطرة التي قد تسبب حوادث مرورية. تعتمد حلول إنترنت المركبات المقبولة حاليًا بشكل كبير على السحابة، حيث تتمتع بقدرة تخزين ومعالجة غير محدودة تقريبًا. ومع ذلك، فإن مشكلة انقطاع الإنترنت ووقت الاستجابة تقيد استخدامها. في هذه الحالة، يمكن لمفهوم الحوسبة الطرفية للمركبات التغلب على هذه القيود من خلال الاستفادة من إمكانيات المعالجة والتخزين للموارد البسيطة الموجودة بالقرب من المستخدم النهائي، مثل المركبات أو البنية التحتية على جانب الطريق. في هذه الأطروحة، نقترح نظام ذكي و تعاوني بين الحوسبة السحابية و الطرفية للسيارات لمساعدة السائق و الذي يعتمد على التعلم الآلي للتنبؤ بمخاطر الحوادث المرورية. ويتكون النظام المقترح من نموذجين، التعلم العميق و التعلم المعزز العميق ، كل منهما يكمل الآخر، وتعمل هذه النماذج معًا على تعزيز فعالية ودقة التنبؤ بالحوادث والوقاية منها. وتظهر النتائج التي تم الحصول عليها أن نظامنا فعال ويمكن أن يساعد في تقليل حوادث الطرق وإنقاذ حياة الآلاف من المواطنين.

الكلمات المفتاحية: إنترنت المركبات ، التعلم العميق، التعلم المعزز العميق، الحوسبة السحابية، الحوسبة الطرفية ،

التعاون بين الحوسبة السحابية و الطرفية .

Acknowledgement

First and foremost, all praise and thanks to **Allah**.

I would like to express my deep and sincere gratitude to my research supervisor, **Prof. Kazar Okba**, for giving me the opportunity to conduct this research and for providing invaluable guidance throughout. His dynamism, vision, sincerity, and motivation have deeply inspired me.

I am also extremely grateful to **Prof Benharzallah Saber** for his attentive listening and relevant advice.

Special thanks to all the jury members who have accepted to preside over and review this work: **Prof. Labib Sadek Terrissa** from Biskra University as the jury president, **Prof. Rachid Seghir** from Batna 2 University, **Dr. Soheyb Ayad** from Biskra University, and **Dr. Samir Bourekkache** from Biskra University.

Additionally, I extend my sincere appreciation to **Dr. Bensari Mochira** for her guidance and insightful criticisms. I am incredibly grateful for her support and consider it an honor to work with her.

Dedication

*To my dear parents, who have been my unwavering support and
inspiration from the very beginning.*

*To my beloved children, the joys of my life, whose love and understanding
have fueled my determination.*

*To my supportive husband, whose encouragement and belief in me have
been instrumental throughout this journey.*

*To my dear brothers and sisters, whose companionship and
encouragement have been invaluable.*

*To all my extended family and friends, for their boundless love, patience,
and unwavering support.*

Thank you all for being there every step of the way.

Zeroual Djazia

Contents

<i>Abstract</i>	<i>i</i>
<i>Résumé</i>	<i>ii</i>
<i>List of Figures</i>	<i>x</i>
<i>List of Tables</i>	<i>viii</i>
<i>List of abbreviations</i>	<i>ix</i>

General introduction

1. <i>Context</i>	<i>1</i>
2. <i>Problem statements</i>	<i>2</i>
3. <i>Contributions</i>	<i>3</i>
4. <i>Thesis Structure</i>	<i>4</i>

Chapter (1): Preliminaries and Basic Concepts

1.1. <i>Introduction</i>	<i>6</i>
1.2. <i>Machine Learning</i>	<i>6</i>
1.2.1. <i>Supervised Learning</i>	<i>7</i>
1.2.2. <i>Unsupervised Learning</i>	<i>12</i>
1.2.3. <i>Reinforcement learning</i>	<i>13</i>
1.3. <i>Deep Learning</i>	<i>14</i>
1.3.1. <i>Convolutional Neural Networks (CNN)</i>	<i>16</i>
1.3.2. <i>Recurrent Neural Networks (RNN)</i>	<i>18</i>
1.3.3. <i>Long Short-Term Memory</i>	<i>18</i>
1.3.4. <i>Auto-Encoder</i>	<i>19</i>
1.4. <i>Deep Reinforcement Learning</i>	<i>20</i>
1.4.1. <i>What is reinforcement learning?</i>	<i>21</i>
1.4.2. <i>Markov Decision Process (MDP)</i>	<i>22</i>
1.4.3. <i>Q-Learning</i>	<i>24</i>
1.4.4. <i>Deep Q-Network (DQN)</i>	<i>25</i>
1.5. <i>Wireless Sensor Networks</i>	<i>28</i>
1.5.1. <i>What is a WSN?</i>	<i>28</i>
1.5.2. <i>WSN applications</i>	<i>29</i>
1.5.3. <i>WSNs for Road Safety applications</i>	<i>30</i>
1.5.4. <i>Challenges in Wireless Sensor Networks</i>	<i>31</i>

1.5.5	<i>WSNs and IOT in a smart city</i>	31
1.5.6	<i>WSNs and ML</i>	32
1.6	<i>Internet Of Vehicles (IOV)</i>	33
1.6.1	<i>IOV's Definition</i>	34
1.6.2	<i>Communication architectures of IOV</i>	36
1.6.3	<i>IOV-layered architecture</i>	36
1.6.4	<i>Challenges in IoV</i>	38
1.6.5	<i>IOV Applications</i>	38
1.7	<i>Cloud -Fog And Edge Computing</i>	39
1.7.1	<i>Cloud computing:</i>	40
1.7.2	<i>Fog computing</i>	43
1.7.3	<i>Edge computing</i>	44
1.7.4	<i>From Cloud Computing To Fog And Edge Computing</i>	47
1.7.5	<i>Cloud-edge computing for the Internet of Vehicles (IoV)</i>	47
1.8	<i>Conclusion</i>	48
<i>Chapter (2): AI in accident risk prediction- State of the art</i>		49
2.1	<i>Introduction</i>	49
2.2	<i>Traffic Accident risk Prediction Using Classical Techniques</i>	49
2.3	<i>Deep Learning For Traffic Accident Prediction</i>	55
2.4	<i>Deep Reinforcement Learning for Traffic Safety</i>	62
2.5	<i>Conclusion</i>	65
 <i>Chapter (3): Collaborative Cloud.DRL - V. Edge.DL approach for Predicting Traffic Accident Risk</i>		
3.1	<i>Introduction</i>	67
3.2	<i>Proposed System Architecture</i>	67
3.2.1	<i>Cloud Layer</i>	67
3.2.2	<i>V.Edge Layer</i>	68
3.3	<i>Cloud_DRL For Accident Risk Prediction</i>	69
3.3.1	<i>Scenarios</i>	69
3.3.2	<i>Key Elements of Cloud _DRL</i>	70
3.3.3	<i>Deep Q-Network (DQN)</i>	72
3.4	<i>V.Edge_Deep Learning For Accident Risk Prediction</i>	75
3.4.1	<i>Data Collection at the Edge</i>	76

3.4.2	<i>Cloud Processing and Training</i>	76
3.4.3	<i>Model Deployment to the Edge (Vehicle)</i>	77
3.5	<i>Collaboration V.Edge_DL / Cloud _DRL</i>	77
3.5.1	<i>V2C (V.Edge-Cloud)</i>	78
3.5.2	<i>V2V (V.Edge-V.Edge)</i>	79
3.6	<i>Conclusion</i>	80

Chapter (4): Experiments and results

4.1	<i>Introduction</i>	81
4.2	<i>Data Selection</i>	82
4.3	<i>Programming Language, Libraries And Used Environment</i>	82
4.4	<i>Data Pre-Processing</i>	84
4.4.1	<i>Data Loading</i>	84
4.4.2	<i>Data Cleaning</i>	84
4.4.3	<i>Data Scaling</i>	84
4.4.4	<i>Handling Class Imbalance</i>	85
4.5	<i>Exploratory Data Analysis (EDA)</i>	85
4.5.1	<i>Accident Distribution Across human factors</i>	87
4.5.2	<i>Accident Distribution Across Environmental Conditions</i>	88
4.5.3	<i>Accident Distribution Across Time</i>	89
4.6	<i>Feature Importance</i>	91
4.7	<i>Evaluation Metrics</i>	94
4.8	<i>Results And Discussion</i>	95
4.8.1	<i>Data Splitting</i>	95
4.8.2	<i>Cloud _DRL Vs V.Edge_DL learning curves</i>	96
4.8.3	<i>Performance comparison</i>	99
4.8.4	<i>Cloud-Edge Collaboration: A Flask-Based implementation</i>	102
4.8.5	<i>Cloud _DRL – V.Edge_DL collaboration</i>	104
4.8.6	<i>Discussion</i>	104
4.9	<i>Conclusion</i>	106

General Conclusion

1.	<i>Summary</i>	107
2.	<i>Perspective</i>	108

List of Figures

Chapter (1): Preliminaries and Basic Concepts

<i>Figure 1. 1: machine learning algorithms</i>	9
<i>Figure 1. 2: Support Vector Machine</i>	11
<i>Figure 1. 3: Artificial Neural Network</i>	12
<i>Figure 1.4: K-Means Clustering</i>	13
<i>Figure 1.5:Single-neuron perceptron model</i> .	14
<i>Figure 1.6:Deep Neural Network</i>	15
<i>Figure 1.7:Convolutional Neural Networks</i>	17
<i>Figure 1.8: Recurrent Neural Networks</i>	18
<i>Figure 1.9: Long Short-Term Memory</i>	19
<i>Figure 1.10: Auto Encoder</i>	20
<i>Figure 1.11:Deep Reinforcement Learning</i>	20
<i>Figure 1.12: Reinforcement Learning</i>	22
<i>Figure 1.13: Markov Decision Process</i>	22
<i>Figure 1.14: Deep Q-Network</i>	26
<i>Figure 1.15: The structure of DQN with replay buffer</i>	27
<i>Figure 1.16: Typical WSN architecture</i>	29
<i>Figure 1.17: WSNs Applications</i>	29
<i>Figure 1.18: The Internet of Vehicles scenario</i>	34
<i>Figure 1.19: Five types of vehicular communications of IOV</i>	36
<i>Figure 1.20: The five-layers architecture of IoV</i>	37
<i>Figure 1.21: Cloud, fog and edge architecture</i>	40
<i>Figure 1.22: Architecture of vehicular edge computing</i>	45

Chapter (3): Collaborative Cloud.DRL - V. Edge.DL approach for Predicting Traffic Accident Risk

<i>Figure 3.1:Cloud_DRL and V.Edge_DL system architecture</i>	68
<i>Figure 3. 2: Traffic environment-state description by the discrete-state Markov process</i>	69
<i>Figure 3.3: Proposed Cloud DRL based accident risk</i>	70
<i>Figure 3.4: Cloud_DQL accident risk prediction</i>	72
<i>Figure 3. 5: deep reinforcement learning process</i>	74
<i>Figure 3. 6: Regional V.Edge_DL traffic accident risk prediction</i>	76
<i>Figure 3. 7: Collaboration / communication (V. Edge -cloud)</i>	80

Chapter (4): Experiments and results

<i>Figure 4.1: Flow-Based Methodology for Data-Driven Prediction</i>	81
<i>Figure 4.2: Accident Distribution</i>	86
<i>Figure 4.3: Accident Distribution Across human factors</i>	87
<i>Figure 4. 4:Accident Distribution Across Environmental Conditions</i>	89
<i>Figure 4.5: Accident Distribution Across Time</i>	90
<i>Figure 4. 6: Attribute importance scores</i>	91
<i>Figure 4. 7:Confusion matrix</i>	94

<i>Figure 4.8: Experiment procedure.</i>	95
<i>Figure 4. 9: Learning curves forDRL Model</i>	98
<i>Figure 4.10: Learning curves for DL Model</i>	98
<i>Figure 4.11: Visual comparison with baselines.</i>	100
<i>Figure 4.12: Confusion Matrix Results of : DRL(a) , DL(b) , LR (c), SVM(d), DT (e), RF(f), XGboost(g).</i>	101
<i>Figure 4. 13: : Cloud-Edge Communication Architecture for Accident Risk Prediction – Flask Implementation</i>	1042
<i>Figure 4. 14: Cloud _DRL - V.Edge_DL Collaboration</i>	104

List of Tables

Chapter (1): Preliminaries and Basic Concepts

<i>Table1. 1 : Q-table</i>	<i>24</i>
----------------------------------	-----------

Chapter (2): AI in accident risk prediction: State of the art

<i>Table2. 1: Related work of application of Classical algorithms for accident risk prediction</i>	<i>53</i>
--	-----------

<i>Table2. 2: Related work of application of DL for accident risk prediction.....</i>	<i>59</i>
---	-----------

<i>Table2. 3: Related work of application of DRL for accident risk prediction</i>	<i>64</i>
---	-----------

Chapter (4): Experiments and results

<i>Table4. 1: Python Libraries Utilized.</i>	<i>83</i>
---	-----------

<i>Table4. 2: Input factors.....</i>	<i>92</i>
--------------------------------------	-----------

<i>Table4. 3: Main metrics for classification</i>	<i>95</i>
---	-----------

<i>Table4. 4: DRL hyperparameters</i>	<i>96</i>
---	-----------

<i>Table4. 5: DL hyperparameters.....</i>	<i>97</i>
---	-----------

<i>Table4. 6: Comparison of (Cloud_DRL, V.Edge_DL) with baseline.</i>	<i>99</i>
--	-----------

List of abbreviations

<i>ICT</i>	Information and Communications Technology
<i>ITS</i>	Intelligent Transportation System
<i>AI</i>	Artificial Intelligence
<i>ML</i>	Machine Learning
<i>ICEDAS</i>	Intelligent and Collaborative Cloud-V.Edge Driver Assistance System
<i>LR</i>	Linear Regression
<i>DT</i>	Decision Tree
<i>RF</i>	Random Forest
<i>NB</i>	Naive Bayes
<i>SVM</i>	Support Vector Machines
<i>ANN</i>	Artificial Neural Network
<i>BP</i>	Back propagation
<i>DNN</i>	Deep Neural network
<i>CNNs</i>	Convolutional Neural Networks
<i>RNNs</i>	Recurrent Neural Networks
<i>FC</i>	Fully Connected
<i>LSTM</i>	Long Short-Term Memory
<i>GRU</i>	Gated Recurrent Unit
<i>DL</i>	Deep Learning
<i>RL</i>	Reinforcement Learning
<i>DRL</i>	Deep reinforcement learning
<i>MDP</i>	Markov Decision Process
<i>DQN</i>	Deep Q-Network
<i>DDQN</i>	Double Deep Q-Network
<i>VANETs</i>	Vehicular Ad-hoc Networks
<i>EU</i>	End User
<i>IOT</i>	Internet of Things
<i>IIOT</i>	Industrial Internet of Things

<i>IOV</i>	Internet of-Vehicle
<i>C.C</i>	Cloud Computing
<i>VCC</i>	Vehicular Cloud Computing
<i>VEC</i>	vehicular edge computing
<i>RSUs</i>	Roadside Units
<i>V2V</i>	Vehicle-to-Vehicle
<i>V2I</i>	Vehicle-to-Infrastructure
<i>V2C</i>	Vehicle-to-Cloud
<i>V2P</i>	Vehicle-to-Pedestrian
<i>V2S</i>	Vehicle-to-Sensor
<i>WAVE</i>	Wireless Access in Vehicular Environments
<i>WiFi</i>	Wireless Fidelity
<i>4G</i>	Fourth Generation
<i>LTE</i>	Long Term Evolution
<i>TCP/IP</i>	Transmission Control Protocol/Internet Protocol
<i>OL</i>	Objects layer
<i>CL</i>	Connection layer
<i>AIL</i>	Artificial Intelligence layer
<i>AL</i>	Application layer
<i>BL</i>	Business layer
<i>BDA</i>	Big Data Analysis
<i>SPI</i>	Service-Platform-Infrastructure
<i>CRM</i>	Customer Relationship Management
<i>SaaS</i>	Software as a Service
<i>IaaS</i>	Infrastructure as a Service
<i>PaaS</i>	Platform as a Service
<i>AWS</i>	Amazon Web Services
<i>MDCs</i>	Micro Data Centers
<i>MEC</i>	Mobile Edge Computing
<i>GPS</i>	Global Positioning System
<i>UK</i>	United Kingdom.
<i>US</i>	United States
<i>MLP</i>	Multilayer Perceptron

<i>TP</i>	True Positive
<i>TN</i>	True Negative
<i>FP</i>	False Positive
<i>FN</i>	False Negative
<i>CM</i>	Confusion Matrix
<i>GPU</i>	Graphics Processing Unit
<i>RAM</i>	Random Access Memory
<i>GA</i>	Genetic Algorithm
<i>GWO</i>	Grey Wolf Optimizer
<i>PSO</i>	Particle Swarm Optimization
<i>SSA</i>	Social Spider Algorithm
<i>MSE</i>	Mean Squared Error
<i>RMSE</i>	Root Mean Squared Error
<i>SLP</i>	Single layer perceptron
<i>MLP</i>	Multilayer perceptron
<i>RBF</i>	Radial basis function
<i>TASP</i>	Traffic Accident Severity Prediction
<i>FM2GI</i>	Feature Matrix to Gray Image
<i>SDAE</i>	Stack Denoising Auto-Encoder
<i>DSTGCN</i>	Deep Spatio-Temporal Graph Convolutional Network
<i>OSM</i>	OpenStreetMap
<i>SGMN</i>	Spatial Gated Memory Network
<i>ATSC</i>	Adaptive Traffic Signal Control
<i>DARC</i>	Double actors and regularized critics
<i>DRIVE</i>	Deep ReInforced accident anticipation with Visual Explanation
<i>SUMO</i>	Simulation of Urban Mobility
<i>VSL</i>	Variable speed limit

General introduction

1. Context

Wireless Sensor Networks (WSNs) are essential for smart city transportation systems, particularly in addressing critical challenges like traffic safety. These networks play a key role in monitoring traffic flow, detecting accidents, and optimizing routes to improve road user safety. Every day, thousands of lives are lost or altered forever due to road accidents. Regardless of age, gender, or stage in life, people face the risk of road accidents while walking, driving, commuting, or undertaking long journeys. Sadly, these accidents can devastate families and communities, leaving a profound and lasting impact on those affected. According to the World Health Organization (WHO), approximately 1.35 million road accidents occur each year, causing serious injuries to an estimated 20 to 50 million individuals worldwide. Currently, road accidents rank as the eighth leading cause of global death and are projected to become the seventh by 2030 if current trends persist [1]. Additionally, road accidents result in the depletion of public resources and impose significant economic burdens in real-world contexts.

To enhance traffic safety, many studies have been conducted to help the development of Active Traffic Management Systems. The main areas of interest covered by these studies are: i) *black-spot detection where road traffic accidents have been concentrated*, these spots are recognized as locations where the frequency and features of accidents stand out, or where there are potential safety hazards compared to other typical areas, typically observed over an extended period (usually 1–3 years). Such distinctions are influenced by various factors including road conditions, traffic flow, climate, and environmental factors [2][3][4] [5].

ii) *Detection of traffic incidents in real time*, which involves the utilization of sensors, cameras, and other data sources to promptly identify and report traffic incidents as they happen and alert people to reduce their effects [6]. iii) *Road accidents prediction, where the prime goal of this research is to predict the road accidents before they occur* [7][8][9]. Predicting the likelihood of traffic accidents is crucial for preventing their occurrence and minimizing the resulting damages proactively. However, accurately predicting traffic accident risk at a detailed spatiotemporal level poses challenges, primarily due to the intricacies of the traffic environment, human behavior, and limited availability of real-time traffic data [10]. In this sense, machine learning (ML) has emerged as a promising tool for forecasting traffic accidents

risk. This capability can provide drivers with early warnings and valuable information to help them avoid potential hazards on the road [11]. With the recent development of Internet-of-vehicle (IOV) technology and the advancement in wireless communications, and computational systems, new opportunities have opened up for intelligent traffic safety, comfort, and efficient solutions.

2. Problem statements

The advent of machine learning and communication technologies has led to the development of intelligent transportation systems (ITS) aimed at enhancing traffic efficiency and addressing public concerns regarding road safety. However, implementing these ITS applications, which rely heavily on machine learning, demands substantial resources and must meet stringent quality of service (QoS) standards, particularly in terms of processing speed and accuracy. Furthermore, due to the limited computing power and storage resources of vehicles, meeting these QoS requirements solely with the vehicles' resources poses a considerable challenge [12]. On the other hand, deep reinforcement learning is considered a superior machine learning approach due to its ability to adapt to complex and dynamic transportation systems. This algorithm relies on continuous interaction with the environment to perceive each new state of the system and respond accordingly. Such interaction necessitates processing massive amounts of data, which requires high-performance computing and storage resources. Unfortunately, these resources are not readily available within vehicles. This highlights the critical need for innovative solutions to overcome resource limitations and enable the effective deployment of machine learning techniques in intelligent transportation systems.

Due to the robust resources supported by cloud computing, it is indeed feasible to consider it as a viable solution to the resource constraints posed by vehicles and to aggregate all the data on road accidents sent by vehicles. However, it is essential to acknowledge the distance between the cloud and the driver, which can introduce latency issues. In certain cases, such as emergencies, this latency can be critical, affecting the response time of the system to the driver. Disruptions or disconnections can exacerbate latency issues and hinder real-time communication between vehicles and the cloud. This presents a potential obstacle in delivering timely responses to drivers, especially in urgent situations where immediate action is required.

Therefore, while cloud computing provides abundant resources to surpass the limitations of onboard vehicle systems, addressing latency and connectivity challenges is crucial for ensuring prompt and effective communication between vehicles and the cloud. Innovative solutions, such as edge computing and caching mechanisms, may be necessary to mitigate these issues and enhance the overall performance and reliability of intelligent transportation systems.

3. Contributions

The Internet-Of-Vehicle (IOV) solutions to traffic safety problems rely heavily on the cloud, as it has virtually unlimited storage and processing power; where data must be moved from the data source location (IOV sensors) to a centralized location in the cloud. However, in addition to the Internet disconnection problem, the cloud might be far from the location of sensors and devices generating these data, which will cause the response time to be slow. Therefore, this might restrict the use of a solution that is based on the cloud, for sudden car accidents prediction.

The concept of V.Edge Computing is an efficient alternative to overcome the limitations of using machine-learning models in the cloud platform. Many emergency predictions take place close to the end user; therefore, they can be process at the edge nodes. This reduces the impact of communication delay and internet disconnection. In this context, we propose an Intelligent Collaborative Cloud-V.Edge Driver Assistance Systems (ICEDAS) framework based on machine learning, which predicts the risks of traffic accidents.

This framework takes advantages of the strengths of the two platforms, where a Deep Q-Learning Network (DQN) algorithm is adopted in the cloud in order to train intelligent agent to warn the driver of any foreseeable risk of traffic accident based on the huge historical data available on the cloud. On the other hand, a deep learning algorithm can be deployed on the V.Edge platform for inference, covering potential response absences by the cloud in predicting sudden traffic risk due to the platform's proximity to the end user. The DL algorithm is trained in the cloud, taking advantage of its scalability and high-end computing resources for model training. The proposed (ICEDAS) aims to achieve the following main objectives:

- 1) *The system must be able to react in a timely manner to warn the driver before entering a critical state:* This involves not only detecting potential hazards but also assessing the severity of the situation and delivering warnings promptly enough for the driver to take preventive

action. Real-time data processing and analysis are crucial to ensure that warnings are issued with minimal delay, allowing drivers to respond effectively and avoid accidents.

2) *The system must deliver adaptive messages to each driver who is at risk of a traffic accident based on their personal conditions:* This entails personalized risk assessment, taking into account individual driving behaviors, preferences, and environmental factors. By analyzing historical data and real-time information, the system can tailor warnings and recommendations to each driver's specific situation, increasing the likelihood of effective intervention and accident prevention.

3) *The system must have the ability to use the cloud and V.Edge to predict the accident risk in an efficient manner:* Leveraging cloud computing and edge computing technologies allows for scalable and distributed processing of vast amounts of data required for accurate accident risk prediction. By harnessing the computational power of the cloud and the proximity of edge devices to vehicles, the system can perform complex analyses and predictive modeling efficiently, enabling timely identification of potential risks and proactive measures to mitigate them. This hybrid approach optimizes resource utilization and enhances the system's overall effectiveness in ensuring road safety.

4. Thesis Structure

The remainder of this thesis is organized as follows:

Chapter (1) introduces the fundamental concepts relevant to this thesis, including classical machine learning algorithms, deep learning, reinforcement learning, and the integration of deep learning with reinforcement learning. It also covers key technologies such as Wireless Sensor Networks (WSNs), the Internet of Vehicles (IoV), cloud computing, edge computing, and their integration.

Chapter (2) presents the state of the art in the application of various machine learning algorithms in the field of road safety, with a particular focus on accident risk prediction.

Chapter (3) provides a detailed theoretical presentation of our proposed approach, which is divided into three main parts. First, we introduce the core algorithm—Deep Reinforcement Learning—and describe the process of developing our machine learning models. Next, we explore the deployment of deep learning algorithms on the edge platform, explaining the

rationale behind this decision. Finally, we examine the cooperation between the cloud and edge platforms, demonstrating how their integration contributes to an effective solution for predicting road accident risks.

Chapter (4) focuses on the implementation of the proposed system, beginning with the selection of datasets essential for experimentation. The datasets were then meticulously preprocessed to ensure optimal quality for analysis, including steps such as data cleaning and normalization. Next, we implemented the three core contributions of our framework, each designed to address specific challenges in traffic accident risk prediction using advanced machine learning techniques. Throughout the implementation process, we continuously monitored and evaluated the performance of our algorithms, aiming to achieve high prediction accuracy and outperform traditional methods, as demonstrated by the evaluation metrics computed for each algorithm.

In the **conclusion**, we present a comprehensive summary of the contributions made in this thesis and emphasize the promising results achieved through the application of machine learning, particularly within the collaborative framework designed to reduce accident risk in smart cities. Furthermore, we outline potential directions for future research, including the integration of computer vision techniques to further enhance road safety, as well as the extension of the proposed framework to other critical application domains beyond traffic management.

Chapter (1): Preliminaries and Basic Concepts

1.1. Introduction

This chapter presents the main concepts related to our thesis and gives an overview of the domains we will use. Section 2 presents Machine Learning and its different categories. Section 3 introduces Deep Learning and its various models. Deep Reinforcement Learning, as a specialized area, is covered in Section 4. Wireless Sensor Networks (WSNs) are discussed in Section 5. Furthermore, the integration of Internet of Vehicles is examined in Section 6, followed by an exploration of the intersection between Cloud Computing and Internet of Vehicles in Section 7.

1.2. Machine Learning

Throughout history, humans have relied on an array of tools to streamline tasks, showcasing the ingenuity of the human mind in inventing various machines. These innovations have significantly simplified life by facilitating travel, industrial processes, computing, and more. Among these advancements, machine learning stands out as a notable contribution [13]. Machine learning (ML) is a subfield of artificial intelligence (AI), empowering computers to "self-learn" learn from provided data and enhance their performance over time. By analyzing large amounts of data and "learning" from its past mistakes, ML algorithms can eventually come up with accurate predictions on their own. Essentially, machine learning leverages accumulated experience to refine its algorithms and models [14]. It finds applications in diverse fields such as recommender systems, image identification, and email filtering and speech recognition. Unlike traditional approaches, modern machine learning benefits from new computing technologies, allowing computers to learn without explicit programming. It hinges on iterative processes where models adjust with fresh data, ensuring reproducible results. Stemming from artificial intelligence research, machine learning focuses on algorithms enabling computers to learn independently. Coined by Arthur Samuel in 1959, it involves creating prediction models from historical training data, incorporating computer science and

statistics. The topic of Machine Learning within Artificial Intelligence (AI) is rapidly growing in interest and prominence. marketing, finance, Healthcare, infrastructure, autonomous vehicles, recommendation systems, gaming, social media, chatbots, cyber security and many more all make use of machine learning techniques [14]. By leveraging vast processing power, machine learning automates tasks, demonstrating unmatched speed and scalability, revolutionizing various domains. However, computers have the capability to learn from their own interactions and tackle problems with minimal human intervention via a process called machine learning [14]. There are various methods through which machines acquire knowledge. In some cases, machines are trained and, in some cases, machines learn on their own. These methods can be broadly categorized into three main types of machine learning:

- Supervised Learning,
- Unsupervised Learning,
- Reinforcement Learning,

In this part, we will to discuss each type of this methods:

1.2.1. Supervised Learning

Supervised learning algorithms are designed to learn from examples. The term "supervised" comes from the idea that the learning process is guided, much like a teacher overseeing a student's progress. In supervised learning, the data consists of input variables and corresponding output variables, meaning the dataset is labeled. Labeled data means that each input comes with an associated output, providing the algorithm with the correct answers during training. During training, the algorithm identifies patterns in the input data and associates them with the correct outputs. After training, a supervised learning algorithm can take new, unseen inputs and predict their labels based on what it has learned. The primary goal of supervised learning is to accurately predict the correct label for new input data [15]. Common examples include spam filtering, illustrating how supervised learning operates under supervision. Additionally, supervised learning algorithms can be further categorized into two types.

a) Classification

Classification is the process of categorizing output data into distinct classes based on one or more input variables. It is commonly utilized when the output variable is discrete or categorical, such as determining if an email is "spam" or "not spam," identifying the presence or absence of a "disease," predicting whether it will "rain" or "not rain," or making binary decisions like "Yes" or "No" and 0 or 1. When the algorithm aims to sort input variables into two distinct classes, it is known as binary classification, as seen in the case of email categorization as "spam" or "not spam." On the other hand, when the algorithm seeks to classify input variables into more than two classes, it is termed as multiclass classification, such as in handwritten character recognition where the classes range from 0 to 9 [15]. Some examples of Classification tasks are Classifying the credit card transactions as legitimate or fraudulent, classifying secondary structures of protein as alpha-helix, beta-sheet or random coil and categorize the news stories as finance, weather, entertainment and sports. [16].

b) Regression

Regression procedure is applied if there is a correlation between the input and output variables. It is used when the value of the output variable is continuous or real, such as house price, weather forecasting, stock price prediction, and so on. Several well-liked regression algorithms that fall under supervised learning are listed below *Figure 1.1*. In this section, we carry out a study of the most used supervised machine learning algorithms for classification and regression techniques:

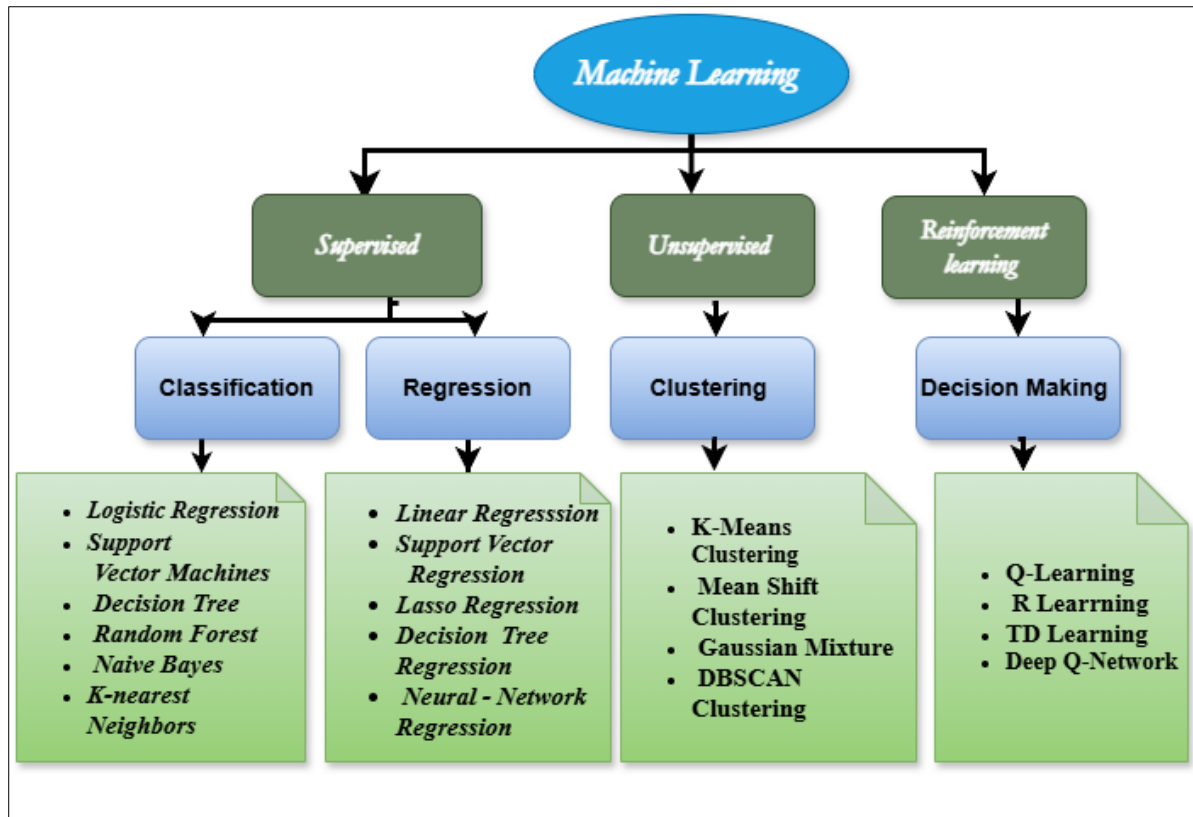


Figure 1. 1: machine learning algorithms

A. Linear Regression & Multiple linear Regression

Linear Regression (LR) stands out as one of the most renowned algorithms in machine learning. Its primary function revolves around forecasting continuous values, such as salaries, ages, or product prices. LR establishes the connection between dependent and independent variables. When employing Simple Linear Regression, a solitary independent variable is utilized to anticipate the value of a quantitative dependent variable. In simpler mathematical terms, the equation for simple linear regression can be broken down as follows:

$y = ax + b$. (x) represents an independent variable, also known as the predictor variable, (y) denotes the dependent variable, often referred to as the target variable. This methodology is deployed on sample data to discern a straightforward line within the linear regression model [17].

B. Decision tree

Decision Tree (DT) serves as a classification technique that executes classification via a learning tree structure. Within this tree, each node signifies a feature or attribute of the data, while the branches symbolize conjunctions of features that guide classifications. Additionally, every leaf node corresponds to a specific class label. Unlabeled samples can undergo classification by comparing their feature values with the nodes present in the decision tree. DT offers numerous benefits, including intuitive expression of knowledge, straightforward implementation, and notable classification accuracy [18].

C. Random Forest (RF)

Random Forest (RF) is a versatile method widely utilized for various classification tasks. This machine learning approach combines data aggregation, bagging, and Decision Tree (DT) models using subsets. It effectively selects feature subsets from each node of the tree while avoiding correlations within the bootstrapped set. When tasked with classifying companies and ratios, RF functions as a forest comprising k trees, thereby providing robust computational capabilities [19].

D. Naive Bayes

Naive Bayes (NB) is a classification technique rooted in Bayes Theorem, making the assumption of independence among predictors. Essentially, it posits that the presence of a particular feature in a class is unrelated to the presence of any other feature. Typically applied in text classification, Naive Bayes serves clustering and classification purposes based on the conditional probability of occurrence [20].

E. Support Vector Machine

Another most widely used state-of-the-art machine learning technique is Support Vector Machine (SVM), SVM defined regression as well as classification problems. The primary objective of employing Support Vector Machines (SVM) is to establish the optimal decision boundary, effectively segregating the n -dimensional space into distinct classes. This ensures accurate placement of new data points. The hyperplane, termed as the best decision boundary, serves this purpose.

- *Support Vector*: These are the nearest positive and negative points, pivotal in defining the decision boundary.
- *Hyperplane*: Positioned at the center, this line serves as the dividing boundary between classes.
- *Margin*: This refers to the gap between two parallel lines, signifying the width of the separation between classes [17].

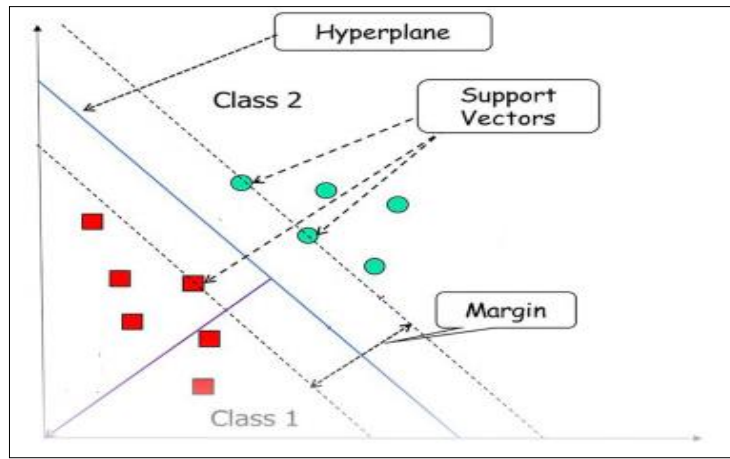


Figure 1.2: Support Vector Machine.

F. Artificial Neural Network

Artificial Neural Network (ANN) utilizes Back-propagation (BP) as a supervised learning strategy to train deep neural networks. Through BP, the network adjusts neuron weights based on calculated errors, facilitating learning from the training process. ANN represents a blend of supervised, unsupervised, and reinforcement learning techniques. *Figure 1.3* illustrates an ANN with multiple layers. The input layer, denoted by i variables such as input1, input2, to input n , receives inputs. Hidden layers, represented by h variables $h1$, $h2$, to h_n , process these inputs. Finally, the output layer comprises o variables, labeled as output1 to output n [17].

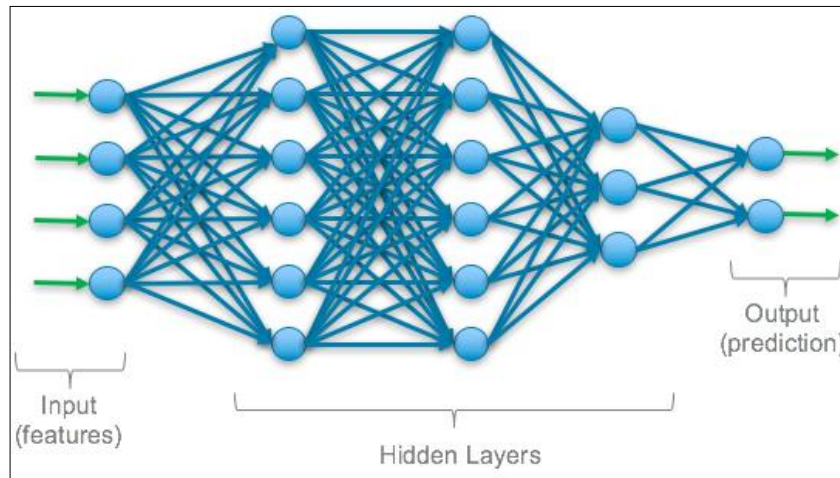


Figure 1. 3: Artificial Neural Network

1.2.2. Unsupervised Learning

Unlike supervised learning, unsupervised learning algorithms operate without labeled inputs, meaning there are no corresponding outputs provided. Essentially, the goal of an unsupervised learning algorithm is to uncover patterns, structures, or knowledge within unlabeled data by clustering sample data into distinct groups based on their similarities. These unsupervised learning techniques are extensively utilized in clustering and data aggregation tasks. In the subsequent sections, we will present comprehensive descriptions of commonly employed unsupervised learning algorithms, including k-means and self-organizing map [18].

a) K-Means Clustering

K-means stands out as a straightforward unsupervised learning algorithm adept at addressing clustering problems. The method offers a simple approach to categorize a dataset into a specified number of clusters. The core concept revolves around defining k centers, each representing a cluster. Strategic placement of these centers is crucial as their positioning significantly impacts the clustering outcome. Hence, the optimal strategy involves positioning the centers as far apart from each other as possible [20].

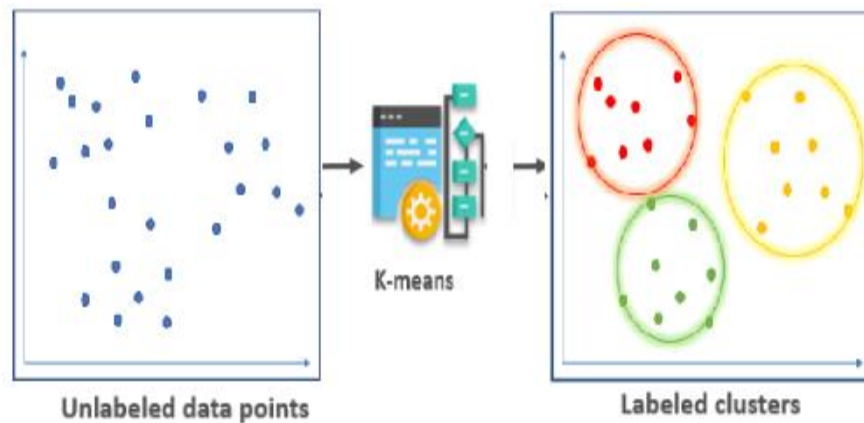


Figure 1.4: K-Means Clustering

b) Self-Organizing Map (SOM)

SOM, also known as Self-Organizing Feature Map, is a widely recognized unsupervised neural network model frequently employed for tasks like dimensionality reduction and data clustering. Typically, SOM comprises two layers: an input layer and a map layer. When utilized for data clustering, the number of neurons within the map layer corresponds to the intended number of clusters. Each neuron within the map layer possesses a weight vector [18].

1.2.3. Reinforcement learning

Reinforcement learning is a machine learning technique that involves training an agent to make decisions based on feedback from the environment. It is a way to solve optimal control problems without having a model of the environment. The agent interacts with the environment, detects its state, and receives reward signals, as it gets closer to its goal. The agent uses these rewards to determine successful actions and learns to choose the right action given any state of the environment. Reinforcement learning has been successfully applied to tasks like game playing and robot control. It serves as a common language for engineers, biologists, and cognitive scientists to exchange their findings in goal-directed behaviors. Deep reinforcement learning techniques have also been developed, which use deep learning algorithms to solve complex uncertain tasks with continuous action and state spaces. The decision to use single-agent or multi-agent reinforcement learning depends on the problem and environment complexity.

1.3. Deep Learning

Deep learning, a subset of machine learning, utilizes artificial neural networks (ANNs) to analyze complex patterns and adapt to dynamic environments. Unlike traditional ML models, deep learning can process unstructured data such as text, images, and audio, making it a powerful tool for automation [21].

It has gained significant traction in areas such as computer vision, natural language processing, speech recognition, and robotics. Neural networks, which are loosely inspired by the structure and function of the human brain, form the core of deep learning systems. Deep neural networks—those with multiple hidden layers—are particularly powerful, as their depth allows them to capture intricate relationships between inputs and outputs. This capability has led to breakthroughs across many industries, including autonomous driving and healthcare. However, deep learning comes with challenges, such as the need for large datasets, substantial computational power, and limited interpretability of models. Despite these hurdles, deep learning continues to evolve and holds tremendous promise for the future of artificial intelligence and technology. According to [22], the general architecture of a deep neural network presented as follow:

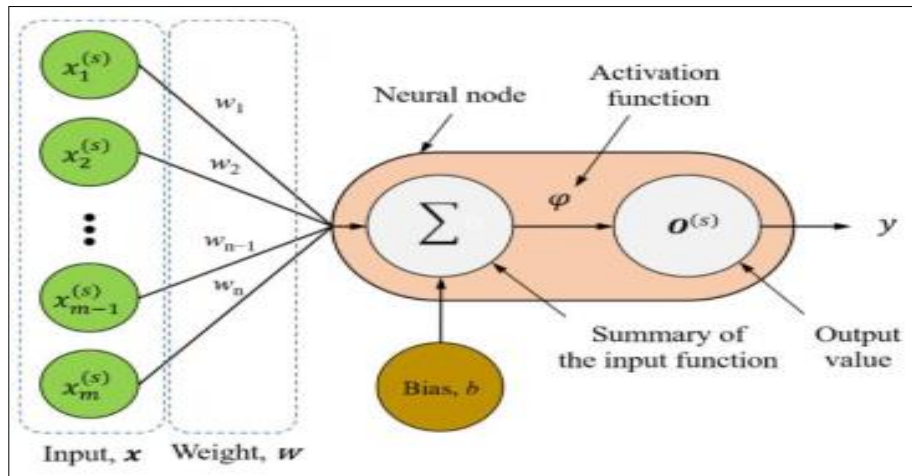
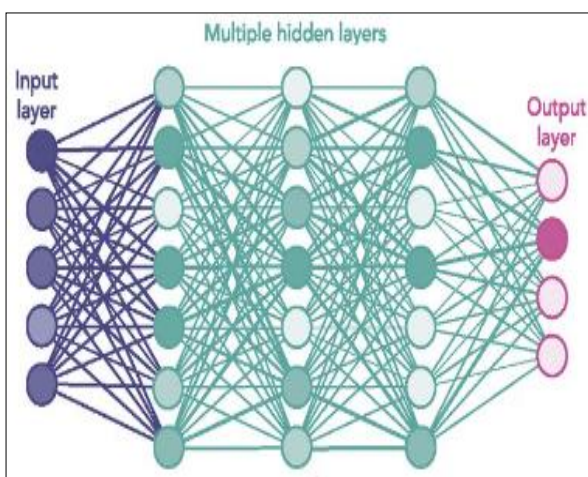


Figure 1.5: Single-neuron perceptron model [22].

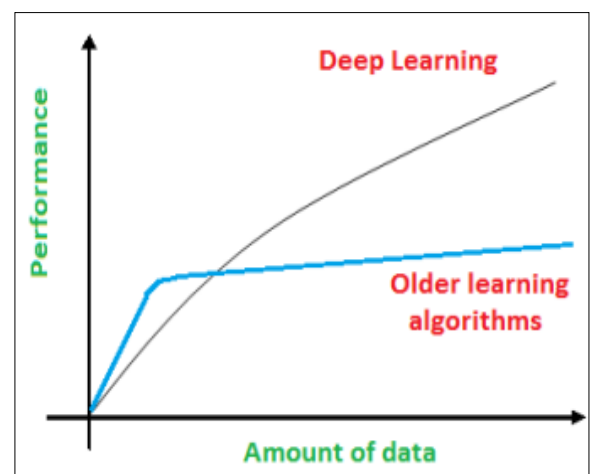
- *Neurons*: Neurons serve as the fundamental components of neural networks. An artificial neuron, also referred to as a perceptron, operates as a mathematical function. It takes one or

more inputs, multiplies them by corresponding weights, and then aggregates these values. The resultant sum is then passed through an activation function, yielding the output of the neuron.

- **Weights:** Weights are pivotal in determining the significance of features in predicting the target outcome. Each artificial neuron utilizes a set of weighted inputs to generate relevant outputs. A positive weight associated with a feature indicates a direct relationship between that feature and the target outcome. Conversely, a negative weight suggests an inverse relationship between the feature and the target outcome.
- **Activation Function:** Activation functions are crucial for the computational efficiency of a training model and facilitate the network in learning intricate patterns within data. The sum of weighted inputs is directed to the activation function, which establishes the threshold requirement for the summed input. This process aids in determining the activation state of the neuron.
- **Neuron Network:** Deep learning involves the utilization of extensive and complex neural networks. These networks consist of numerous layers of neurons organized in a structured fashion, akin to the organization of cells in the human brain. The learning process within neurons occurs across these layers, where weighted inputs undergo multiple nonlinear transformations before producing an output. The schematic view of Deep Neural network (DNN) is as shown in Figure 1.6. In deep learning, the input layer receives initial data, while hidden layers process information between the input and output layers, ultimately producing the result.



(a) DL Structure



(b) DL Performance

Figure 1.6: Deep Neural Network.

Recently, two primary factors have contributed to the newfound utility of deep learning:

- 1) *Deep learning necessitates extensive sets of labeled data.* For instance, the advancement of driverless cars demands vast collections of images and extensive hours of video footage.
- 2) *Deep learning requires significant computational resources.* Advanced GPUs offer parallel processing capabilities that are particularly efficient for deep learning tasks. When coupled with clusters or cloud computing, this enables development teams to significantly reduce the time required for training deep learning networks [23].

Deep learning encompasses discriminative (supervised) and generative (unsupervised) approaches, with prominent models including CNNs and RNNs for discrimination and GANs and AEs for generation. The following section presents into various types of deep learning models.

1.3.1. Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNNs) represent a robust category of deep learning models extensively utilized across diverse applications such as object detection, speech recognition, computer vision, image classification, and bioinformatics. Furthermore, they have exhibited notable proficiency in tasks involving time series prediction. CNNs are structured as feedforward neural networks that exploit convolutional operations to extract salient features from data. Unlike traditional methodologies, CNNs autonomously discern and understand features from the input data, obviating the need for manual feature extraction . Inspired by visual perception, CNNs are designed to emulate the intricate processes of human visual cortex. Key elements of CNNs encompass the convolutional layer, pooling layer, and fully connected layer. Illustrated in Figure1.7 is a representative CNN architecture tailored for image classification endeavors [24].

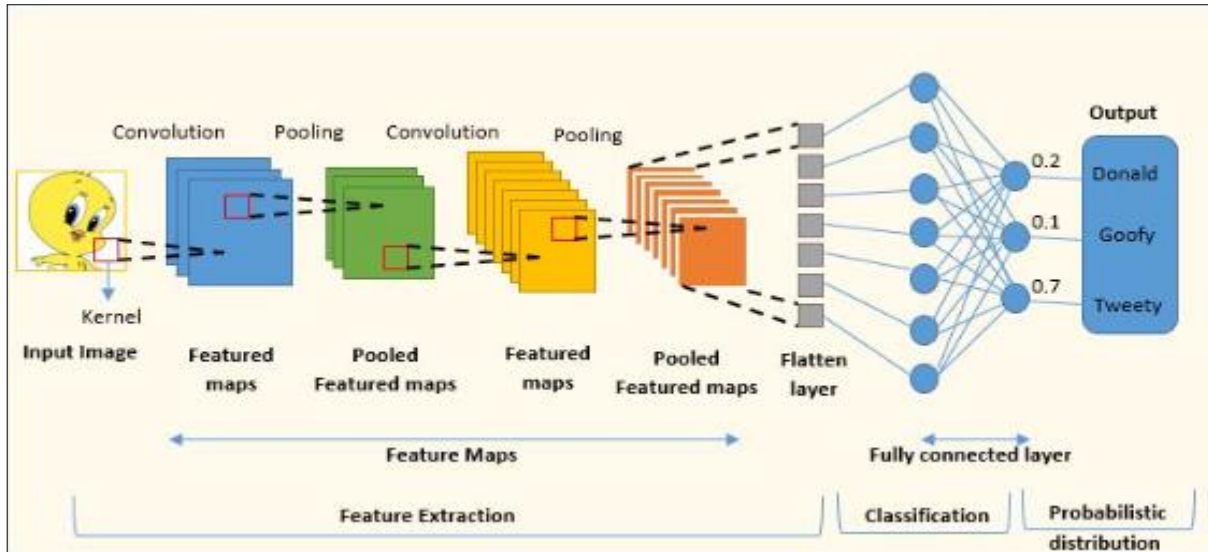


Figure 1.7: Convolutional Neural Networks.

- **Convolutional layer:** The convolutional layer in CNNs extracts features from input data through convolution operations across multiple layers. Lower layers capture basic features like texture and edges, while higher layers identify more abstract features. Learnable convolution kernels, typically of equal length and width, slide over input feature maps to execute convolution operations, as illustrated in Figure 1.7 [24].
- **Pooling layer:** The pooling layer, typically positioned after the convolutional layer, reduces network connections via down-sampling and dimensionality reduction, addressing computational load and overfitting. It enhances CNNs' ability to recognize objects amidst distortion by pooling various image dimensions, yielding more robust output feature maps. Various pooling methods like Max Pooling and Average Pooling contribute to this process [24].
- **Fully Connected (FC) Layer:** The Fully Connected (FC) layer is typically situated at the conclusion of CNN architecture. In this layer, each neuron establishes connections with all neurons in the preceding layer, akin to a traditional multi-layer perceptron neural network. Input for the FC layer is derived from the last pooling or convolutional layer, represented as a vector obtained by flattening the feature maps. Serving as the classifier in the CNN, the FC layer facilitates the network in making predictions [24].

1.3.2. Recurrent Neural Networks (RNN)

In the field of machine learning, Recurrent Neural Networks (RNNs) are notable for their capacity to handle sequential data. Unlike conventional neural networks, RNNs possess a memory element enabling them to store information from prior inputs. This characteristic renders them well-suited for tasks involving temporal sequences, including but not limited to predicting stock prices, language modeling, and speech recognition. Recurrent Neural Networks are valuable tools for grasping the sequential arrangement of text, wherein the significance of each word hinges on preceding ones or sentences. This makes them apt for tasks such as language translation, sentiment analysis, and generating text.

One drawback of basic RNNs is their limited short-term memory, hindering their effectiveness with long sequences. To address this issue, more sophisticated RNN variations have emerged, such as Long Short-Term Memory (LSTM), bidirectional LSTM, and Gated Recurrent Unit (GRU).

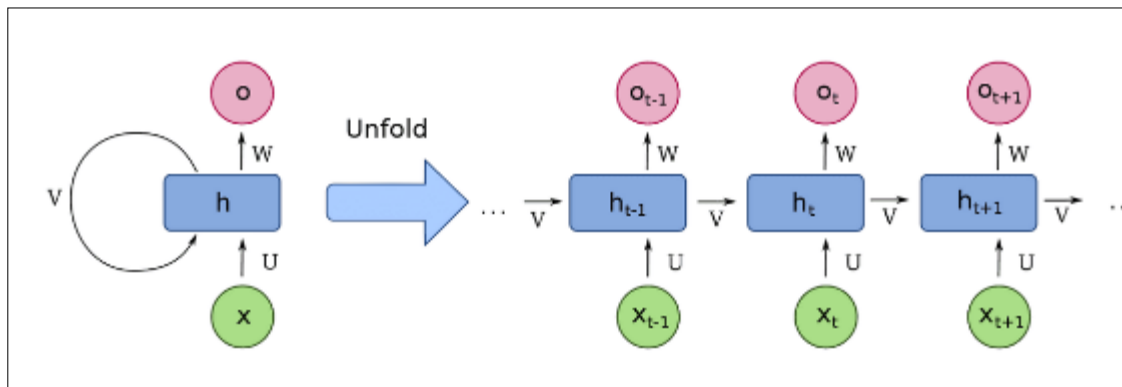


Figure 1. 8: Recurrent Neural Networks.

1.3.3. Long Short-Term Memory

Long Short-Term Memory (LSTM) is a widely utilized type of RNN architecture incorporating specialized units aimed at addressing the vanishing gradient problem. Within an LSTM unit, a memory cell possesses the capability to retain data over extended durations, with the regulation of information flow into and out of the cell orchestrated by three distinct gates. Specifically, the 'Forget Gate' discerns the retention of pertinent information from the preceding state cell,

discarding obsolete data, while the 'Input Gate' governs the influx of new information into the cell state. Simultaneously, the 'Output Gate' dictates and manages the generation of outputs. Given its adeptness in resolving recurrent network training challenges, the LSTM network stands out as one of the most triumphant RNN implementations. The architecture of LSTM model is reflected in Figure 1.9.

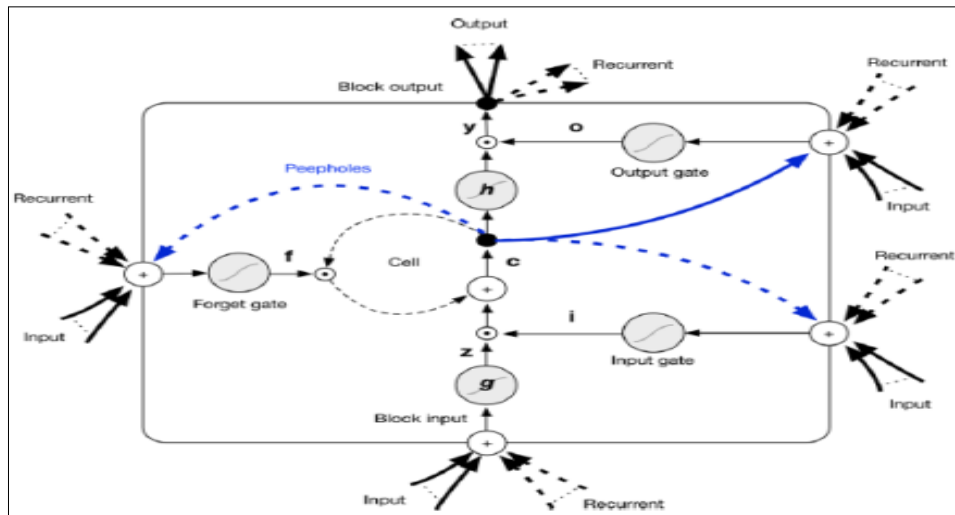


Figure 1. 9: Long Short-Term Memory.

1.3.4. Auto-Encoder

Deep autoencoders are unsupervised neural networks used to transform input vectors into corresponding outputs, enabling dimensional manipulation of data representation. They exhibit adaptability by learning compressed data encoding unsupervisedly and can be trained incrementally, layer by layer, reducing computational resources. These networks extract essential features from input data, with hidden layers having lower dimensionality for feature compression. As shown in Figure 1.10, the standard architecture includes an input layer for initial data, hidden layers for encoding with fewer neurons, and an output layer for reconstructing the input faithfully.

Their utility extends across diverse domains, providing benefits in tasks such as data compression, anomaly detection, and feature learning. Operating without the need for labeled data during training, they function unsupervised, rendering them useful in situations where acquiring labeled data proves challenging.

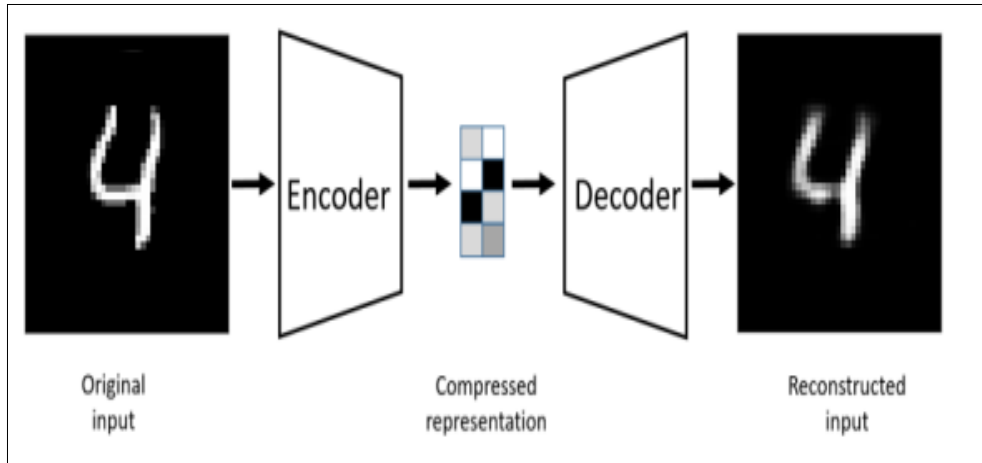


Figure 1.10: Auto Encoder

1.4. Deep Reinforcement Learning

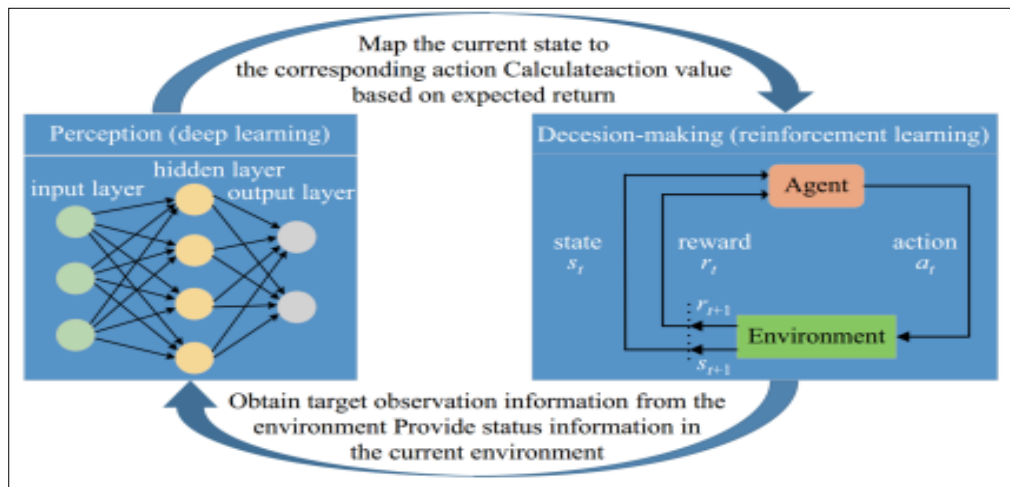


Figure 1.11: Deep Reinforcement Learning [28]

Reinforcement Learning (RL) stands out among Machine Learning (ML) disciplines for its unique trial-and-error approach, where algorithms learn by interacting with their environment. Considered a vibrant field within artificial intelligence, RL is seen as a pivotal step towards achieving artificial general intelligence. Recent years have witnessed significant advancements in RL, particularly with the fusion of deep learning techniques, known as deep reinforcement learning.

Deep reinforcement learning (DRL) is a cutting-edge approach that combines reinforcement learning with deep learning techniques, enabling machines to learn optimal behaviors through interaction with their environment. This fusion empowers DRL systems to navigate complex decision-making tasks across diverse domains such as healthcare, robotics, smart grids, finance, and more. In healthcare, DRL assists in disease diagnosis, treatment optimization, and personalized healthcare interventions. In robotics, it facilitates autonomous navigation, manipulation, and task execution. Smart grids benefit from DRL's optimization of energy production, distribution, and consumption. In finance, DRL revolutionizes algorithmic trading, risk management, and portfolio optimization. Despite challenges, DRL holds promise for transformative impacts on various industries by unlocking unprecedented capabilities in artificial intelligence.

1.4.1. What is reinforcement learning?

Reinforcement learning (RL) offers a broad strategy for addressing reward-driven challenges. RL seeks to emulate the manner in which humans acquire new knowledge, primarily through interaction with the environment rather than direct instruction from a teacher. For example, when a baby learns to wave hands, cry and laugh, it learns from the feedback from parents. When we drive a vehicle, we learn to turn left and right to avoid the crash of the vehicle on the road. RL serves as the mechanism through which machines acquire the capability to attain objectives through interactions with their environment. From a mathematical standpoint, RL is also framed as a problem of sequential decision-making and control. For instance, when driving a vehicle, we need to choose turning right or left every time after we make the previous decision. Reinforcement learning is one type of machine learning. In supervised learning, algorithms are developed to make outputs mimic the labels given in the training set. In contrast to supervised learning, offering supervision in RL poses challenges due to the lack of knowledge about the correct decisions. For instance, when driving a vehicle, it's impractical to assign labels to every image captured by the camera [25].

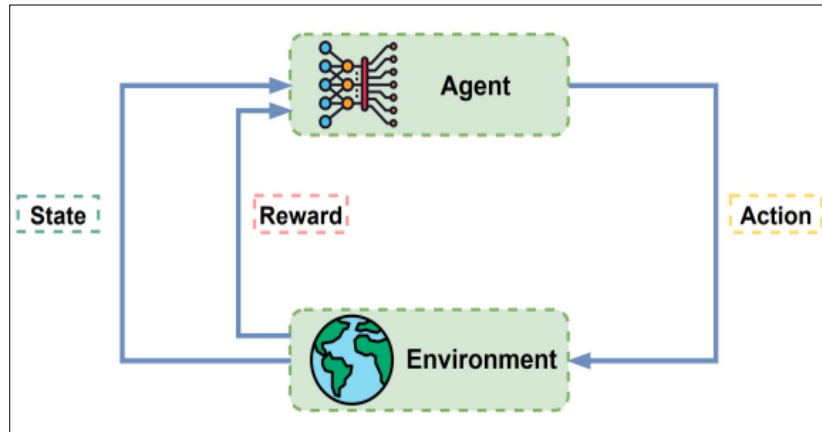


Figure 1.12: Reinforcement Learning

1.4.2. Markov Decision Process (MDP)

An MDP serves as a mathematical tool in (deep) reinforcement learning, designed for tackling sequential decision-making challenges amidst uncertainty. It aids agents in devising optimal strategies within given environments to attain desired states efficiently. By employing MDP policies, agents adapt their actions based on environmental dynamics, aiming to enhance their performance. This optimization process revolves around a reward mechanism, wherein actions are prioritized according to their anticipated rewards.

An MDP is characterized by a finite set of states, denoted as S , along with an action set A , a reward function $R_a(s, s')$, and a transition probability function $P(s, s')$. In real-world scenarios, the exact transition probability distributions are often unknown. Instead, simulators can be

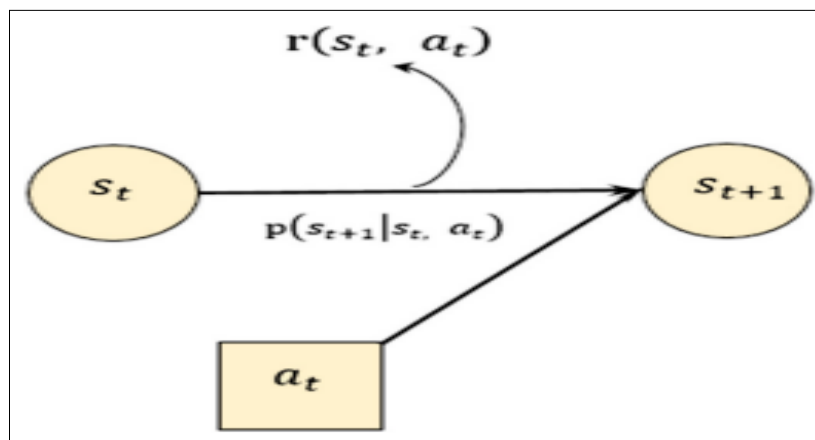


Figure 1.13: Markov Decision Process

utilized to indirectly model the MDP by generating samples from these distributions, enabling the estimation of the transition probability function. Additionally, the model adheres to the Markov property, indicating that the next state solely depends on the current state and action. This property offers the advantage of eliminating the need to consider all states in the trajectory. It's crucial to appropriately translate the problem into an MDP formulation to facilitate analysis using Reinforcement Learning (RL). Key terms in defining the MDP include [26]:

a) *Environment:* The environment represents the domain in which an agent or software algorithm operates and interacts. It receives inputs such as the actions taken by the agent and its current state, while producing outputs consisting of the next state and any associated rewards. This environment encompasses any entity that processes and influences an agent's actions and their resulting outcomes, including scenarios like games, healthcare systems, or the living environment of an agent.

b) *States:* States refer to the collection of environmental states, denoted as S , which consists of a finite set $\{s_1, \dots, s_N\}$, where N represents the size of the state space, denoted as $|S| = N$. A state serves as a distinct representation encapsulating all essential aspects within the problem being modeled [27].

c) *Actions:* Actions, denoted as A , constitute a finite set $\{a_1, \dots, a_K\}$, with K representing the size of the action space, denoted as $|A| = K$. They are instrumental in influencing the system's state. The set of actions that can be executed within a specific state $s \in S$ is indicated by $A(s)$, where $A(s) \subseteq A$. In some systems, not all actions can be applied in every state, but in general we will assume that $A(s) = A$ for all $s \in S$. In more structured representations, the fact that some actions are not applicable in some states, is modeled by a precondition function pre (precondition function): $S \times A \rightarrow \{\text{true}, \text{false}\}$, stating whether action $a \in A$ is applicable in state $s \in S$ [27].

d) *Transition Function:* The Transition Function determines the system's movement from one state to another upon applying action $a \in A$ in state $s \in S$. This transition is governed by a probability distribution over potential outcomes. Represented as $T: S \times A \times S \rightarrow [0, 1]$, denoted by $T(s, a, s')$, it signifies the likelihood of transitioning from state s to state s' after action a is taken. It's essential that $T(s, a, s')$ is non-negative and doesn't exceed 1 for all states s , actions a , and possible next states s' . Additionally, for each state s and action a , the sum of probabilities of all possible next states equals 1, ensuring a proper probability distribution. Alternatively, if

action a is not feasible in state s , $T(s, a, s')$ can be set to 0 for all $s' \in S$. To establish a chronological sequence of actions, a discrete global clock is introduced, denoted as $t = 1, 2, \dots$. This allows for the distinction between different states (and actions) occurring sequentially during interactions. The system is deemed Markovian if the outcome of an action solely hinges on the current state, disregarding prior actions and visited states (history). Mathematically, this translates to $P(s_{t+1} | s_t, a_t, s_{t-1}, a_{t-1}, \dots) = P(s_{t+1} | s_t, a_t) = T(s_t, a_t, s_{t+1})$ [27].

e) Reward: The Reward function $R_a(s, s')$ provides a numerical value indicating the benefit an agent receives for being in a particular state after executing an action $a \in A$ in state $s \in S$ leading to new state $s' \in S$. Rewards serve as indicators of the state's utility, offering higher rewards for favorable states and lower rewards for unfavorable ones. They serve as feedback for the agent, conveying positive or negative outcomes based on its actions.

Figure 1.14 denotes an MDP process, in each time step t , an action a_t is performed on a process in the present state s_t , and there is a transition to the next state s_{t+1} . A Reward r_t is acquired in this transition.

1.4.3. Q-Learning

Q-learning is a well-established off-policy reinforcement learning (RL) algorithm that employs the Temporal Difference method. Off-policy learning implies that the agent learns from data generated by different policies or from past experiences, achieving high learning efficiency and sample utilization. In the Q-learning algorithm, for each state, denoted as ' s ', the algorithm records the Q-value for all feasible actions within that state in a data structure like Table 1.1[28].

Table 1. 1 . Q-table

Q-table	a_1	a_2		a_n
S_1	$Q(S_1, a_1)$	$Q(S_1, a_2)$		$Q(S_1, a_n)$
S_2	$Q(S_2, a_1)$	$Q(S_2, a_2)$		$Q(S_2, a_n)$
.....				
S_n	$Q(S_n, a_1)$	$Q(S_n, a_2)$		$Q(S_n, a_n)$

The learning process updates the Q-table iteratively. The Q value update step is given below:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha (r_{t+1} + \gamma \max_{a \in A} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t))$$

where $\gamma \in [0, 1]$ accounts for the discount factor and can only take a value of 1 in very rare MDPs condition. α is learning rate, $\max_{a \in A} Q(s_{t+1}, a_{t+1})$ is the maximum future reward for all actions in state s_t , $(r_{t+1} + \gamma \max_{a \in A} Q(s_{t+1}, a_{t+1}))$ denotes true Q value and $Q(s_t, a_t)$ is estimate Q value. Use this equation to iterate the Q values in the table until they converge. Then for each state, the action is chosen by referring to the maximum Q value in Q-table to obtain the optimal policy. The algorithm is described below:

Algorithm Q-Learning

- 1: Initialize $Q(s, a)$ randomly
- 2: **Repeat**
- 3: Select action a_t based on an exploration policy $\pi(s, a)$
- 4: Perform action a_t , observe reward r_{t+1} and new state s_{t+1}
- 5: $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha (r_{t+1} + \gamma \max_{a \in A} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t))$
- 6: $s_t \leftarrow s_{t+1}$
- 7: **Until Termination**

1.4.4. Deep Q-Network (DQN)

In this section, we'll delve into one of the most widely recognized Deep Reinforcement Learning (DRL) algorithms known as Deep Q Network (DQN). Familiarizing oneself with DQN is crucial as numerous cutting-edge DRL techniques draw inspiration from it. Initially introduced by researchers at Google's DeepMind in 2013 within the paper titled "Playing Atari with Deep Reinforcement Learning", the DQN algorithm was outlined along with its architectural intricacies. The authors elucidated its remarkable efficacy in achieving human-level performance when playing Atari games [29].

The aim of reinforcement learning is to identify the optimal policy, which is the strategy yielding the highest cumulative reward (the total sum of rewards obtained in an episode). To

derive this policy, we begin by computing the Q-function. Subsequently, utilizing the Q-function, we determine the policy by selecting the action in each state that maximizes the Q-value [29]. In the case where the environment consists of a large number of states and actions, it will be very expensive to compute the Q values of all possible state-action pairs in an exhaustive fashion. The optimal approach involves employing a neural network to approximate the Q-values, with this neural network referred to as the Q-network. When utilizing a deep neural network for this purpose, it is termed as a Deep Q Network (DQN). The Q function is denoted by $Q_{\theta}(s, a)$ where θ is the parameter of neural network. As Figure 1.14 shows, we just feed state D as an input to the network and it returns the Q value of all actions in state D, which are *up*, *down*, *left*, and *right*, as output. Then, we select the action that has the maximum Q value. Since action *right* has a maximum Q value, we select action *right* in the state D.

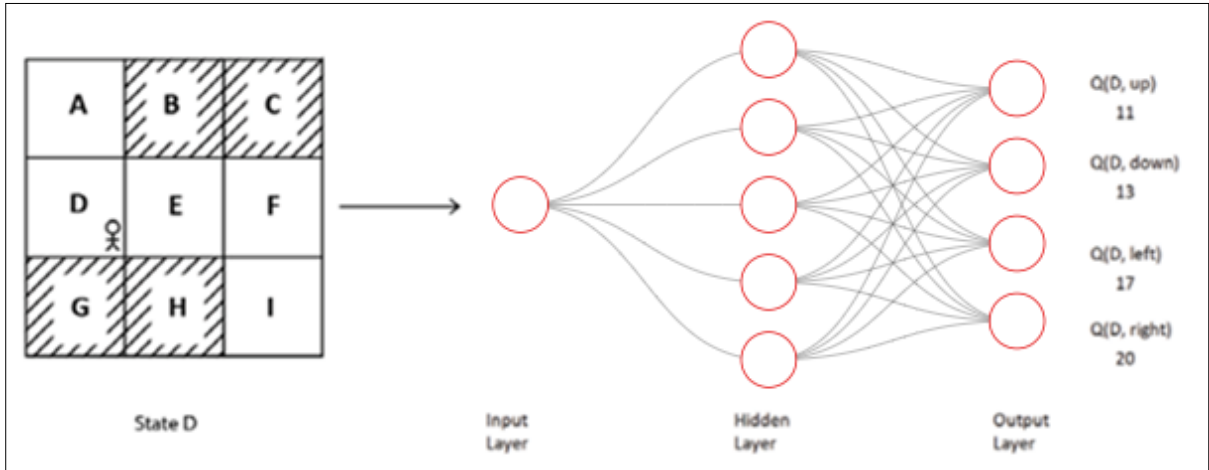


Figure 1. 14: Deep Q-Network

Deep Q-Network (DQN) is an improved model-free Q-Learning approach for discrete actions that approximates the Q-function as $Q(st, at; \theta)$ through deep neural networks. The objective function is represented by the loss function that is shown in the following equation [28]:

$$J(\theta) = E(s_t, a_t, r_{t+1}, s_{t+1}) [(y_t - Q(s_t, a_t, \theta))^2]$$

Where y_t is the learning target:

$$y_t = r_{t+1} + \gamma \max_{a' \in A} Q(s_{t+1}, a'; \theta')$$

A buffer known as a replay buffer is utilized to collect the agent's experience, and based on this experience, the network is trained.

- **Replay buffer**

The transition from a state s to the subsequent state s' , initiated by the agent's action a and followed by the receipt of reward r , is recorded in a buffer known as a replay buffer or experience replay, commonly denoted as (s, a, r, s') . This transition data constitutes the agent's accumulated experience, which is stored within the replay buffer across numerous episodes. The underlying principle of utilizing the replay buffer to retain the agent's experience is to facilitate the training of our DQN by sampling experiences (transitions) from this buffer.

The agent's transition information is collected over many episodes and saved in the replay buffer. A random minibatch of transitions is sampled from the replay buffer and used to train the network. Below is an illustration of a replay buffer [29]. Figure 1.15 shows the training flowchart of DQN. DQN maintains two networks : target network and evaluate network.

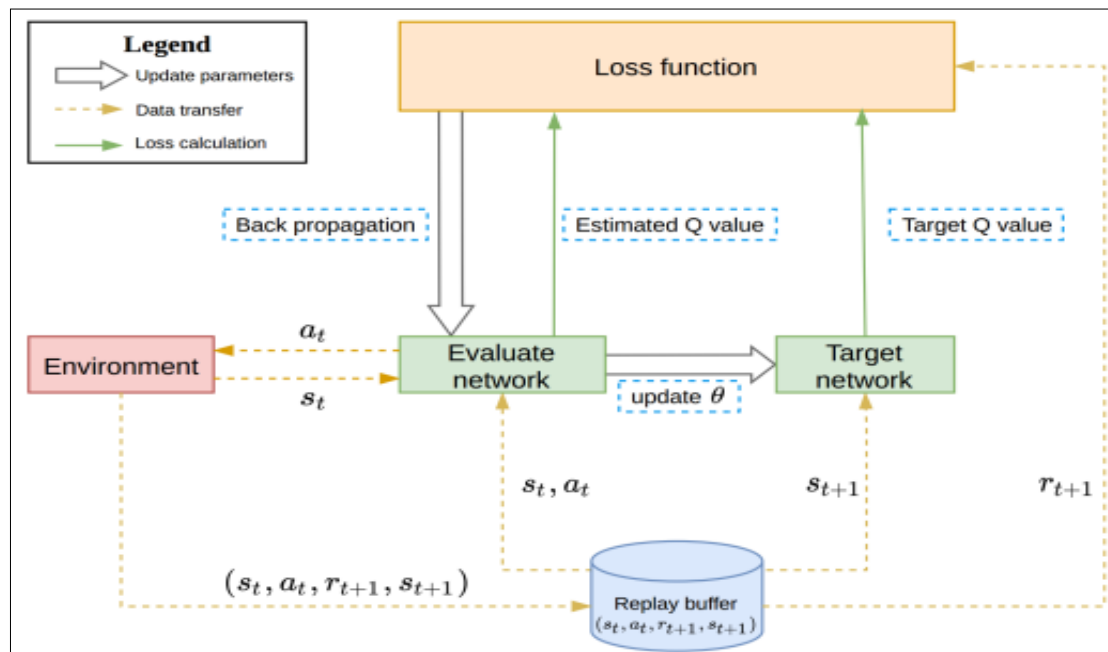


Figure 1. 15: The structure of DQN with replay buffer

1.5. Wireless Sensor Networks

Wireless Sensor Networks (WSNs) are sophisticated systems comprising sensors, computing components, and communication devices designed to enable real-time monitoring, data collection, and response to events within a specific environment. Essentially, they serve as bridges between the digital and physical realms, facilitating seamless interaction and data exchange. WSNs have gained substantial momentum in recent years due to their diverse and expanding applications across various sectors [30]. They offer unparalleled flexibility and scalability, making them integral to modern information technologies. By leveraging advancements in sensor technology and wireless communication, WSNs empower users to gather crucial insights and make informed decisions based on real-time data. This transformative technology continues to evolve rapidly, driving innovation and reshaping how we interface with and understand our surroundings.

1.5.1. What is a WSN?

A Wireless Sensor Network (WSN) comprises distributed sensors and one or more sink nodes, also known as base stations. These sensors continuously monitor physical conditions like temperature, vibration, or motion in real-time, generating sensory data. Each sensor node can both originate and route data. In applications such as event monitoring, sensors transmit data to sinks when detecting specific events. Sinks then collect this data, which can be relayed to end-users through various means like direct connections, the Internet, satellite, or wireless links. Figure 1.16 illustrates a standard WSN architecture [31].

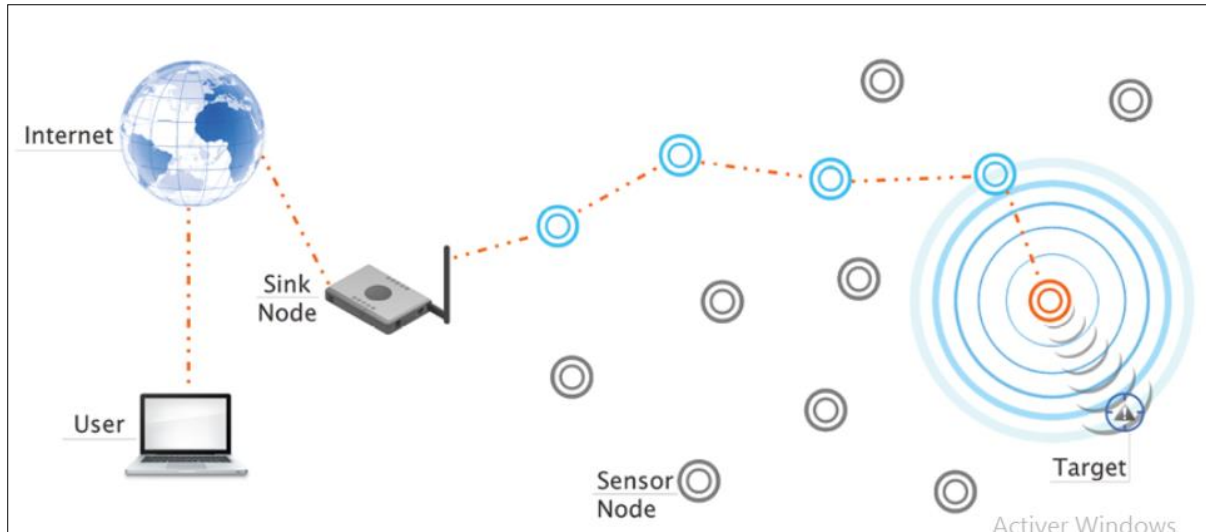


Figure 1.16: Typical WSN architecture [31].

1.5.2 WSN applications

In recent years, sensors have become smaller, more affordable, and smarter, enabling wireless integration within networks for efficient communication. The design and infrastructure of wireless sensor networks (WSNs) are shaped by specific application needs, objectives, costs, hardware capabilities, and maintenance requirements. WSNs consist of distributed autonomous devices equipped with sensors to detect various parameters like temperature and sound across different applications [33].



Figure 1.17: WSNs Applications [32].

Initially developed for military use, WSNs are now predominantly applied in civilian sectors such as weather monitoring, healthcare, and traffic management. environmental monitoring, home automation, chemical and biological assault detection, smart grid deployment, surveillance, and many more. Wireless Sensor Networks (WSNs) are crucial in aquaculture and the oil industry, serving purposes such as data collection, offshore exploration, disaster prevention, tactical surveillance, and pollution monitoring. These networks play important roles in these sectors by facilitating diverse functions like gathering data, conducting offshore surveys, preventing disasters, conducting surveillance operations, and monitoring pollution levels [34].

1.5.3 WSNs for Road Safety applications

In road traffic safety applications, wireless sensor nodes play a critical role in accident detection and response. These nodes are equipped with advanced sensors like magnetometers and micro-radars, enabling them to continuously monitor vehicle behavior and traffic conditions. By analyzing data in real-time, these sensors can detect sudden changes such as abrupt deceleration, rapid acceleration, or unexpected maneuvers that may indicate a potential accident.

The data collected by these sensors is processed locally or transmitted to a central monitoring system. In the event of a detected anomaly, such as a collision or erratic driving behavior, alerts can be immediately triggered to notify authorities and emergency responders. This rapid response capability is crucial for reducing emergency response times and improving outcomes in accident scenarios.

Moreover, wireless sensor networks contribute to overall traffic management by providing valuable insights into traffic flow, congestion patterns, and road conditions. By integrating these sensors into road infrastructure, such as traffic lights and signage, traffic control systems can be optimized to enhance safety and efficiency.

The widespread adoption of wireless sensor technology in road traffic safety reflects its versatility and effectiveness in improving transportation systems. By leveraging these advanced sensor networks, cities and transportation agencies can implement proactive measures to mitigate accidents, enhance emergency response capabilities, and ultimately create safer road environments for drivers and pedestrians alike.

1.5.4 Challenges in Wireless Sensor Networks

Wireless Sensor Networks (WSNs) are revolutionary systems that enable real-time data collection and monitoring in diverse environments. However, these networks encounter various challenges that impact their performance, reliability, and scalability. Understanding these challenges is essential for developing effective solutions to optimize WSN deployments and maximize their utility across different applications. Challenges in WSNs include [30]:

1. **Reliability:** WSNs, being wireless networks, are susceptible to issues like packet loss, which can be particularly critical in applications such as chemical attack detection where data integrity is crucial.
2. **Power Consumption:** Sensor nodes in WSNs are typically battery-powered, leading to limited node lifespan and emphasizing the need for energy-efficient designs across all aspects of network operation.
3. **Node Size:** The push for miniaturization in WSNs presents ongoing challenges in developing smaller nodes that maintain or exceed the efficiency of larger counterparts, despite current sensor nodes already being as small as a coin.
4. **Mobility:** Applications like vehicle tracking demand WSNs capable of dynamically changing routing paths and infrastructure to accommodate mobile nodes, posing significant challenges for network management and stability.
5. **Privacy and Security:** Unlike wired channels, wireless communication in WSNs is vulnerable to unauthorized access and data interception, necessitating robust privacy and security measures to safeguard sensitive information and network integrity.

1.5.5 WSNs and IOT in a smart city

Wireless Sensor Networks (WSNs) are integral to the Internet of Things (IoT), providing connectivity, security, control, and awareness across diverse applications. These networks employ low-power, compact sensors to efficiently monitor and gather data from various environments. WSNs consist of numerous sensor nodes equipped with processing, detection, communication, and power components, collaborating to collect and transmit data. Despite their utility, WSNs face constraints like energy efficiency, processing power, memory, topology management, mobility, and lifespan limitations. Sensor nodes operate over short wireless ranges and autonomously organize into networks after deployment, often without prior

planning. Each node contains sensor modules for measuring physical parameters and essential components like battery, memory, processing, and communication modules. Energy conservation is critical due to limited battery life, with nodes employing various communication modes to optimize power usage. WSNs herald a transformative shift towards interconnected smart spaces, where miniature devices seamlessly integrate wireless technology into everyday life [35]. WSNs can be classified into different categories based on various criteria [30], as follows:

- Deterministic and non-deterministic networks.
- Static and mobile networks.
- Single-sink and multi-sink networks.
- Static-sink and mobile-sink networks.
- Self-reconfigurable and non-self-configurable networks.
- Single-hop and multi-hop networks.
- Homogeneous and heterogeneous networks.

Cutting-edge Artificial Intelligence (AI) and Machine Learning (ML) techniques are poised to revolutionize fully automated IoT applications. In smart cities, low-data-rate Wireless Sensor Networks (WSNs) play a crucial role in monitoring and managing diverse applications. Sensor nodes are the cornerstone of IoT technology infrastructure. Nonetheless, researchers face challenges in areas like power management, security, and data handling, prompting the development of innovative techniques and technologies to tackle these issues [36] [37].

1.5.6 WSNs and ML

Wireless sensor networks (WSNs) operate in dynamic environments characterized by rapid and unpredictable changes, which can be influenced by external factors or intentional interventions by system designers. In response to these challenges, WSNs leverage machine learning techniques to adapt without the need for frequent redesigns. Machine learning enables WSNs to learn from data patterns and environmental cues, optimizing resource allocation and extending network lifespan [38]. By applying machine learning algorithms, WSNs can autonomously adjust sensor configurations, routing protocols, and data processing strategies to efficiently cope with dynamic environmental conditions. This integration of machine learning with WSNs enhances their adaptability and performance in real-world scenarios, ensuring

reliable operation and responsiveness in dynamic environments. Using machine learning (ML) techniques in wireless sensor networks (WSNs) offers several benefits:

1. **Adaptability:** ML allows WSNs to adapt to changing environmental conditions and varying network dynamics without manual intervention, improving overall efficiency and performance.
2. **Enhanced Resource Management:** ML algorithms optimize resource allocation within WSNs, leading to better utilization of energy, memory, and bandwidth resources, thereby extending the network's operational lifespan.
3. **Fault Detection and Diagnosis:** ML models can detect anomalies and faults in sensor data, enabling proactive maintenance and troubleshooting to prevent network disruptions.
4. **Predictive Analytics:** ML enables predictive modeling based on historical sensor data, facilitating proactive decision-making and timely responses to emerging trends or events.
5. **Optimized Routing and Data Processing:** ML algorithms can optimize data routing and processing within WSNs, minimizing latency and reducing energy consumption.
6. **Security and Anomaly Detection:** ML techniques enhance network security by identifying and mitigating security threats, such as intrusion detection and prevention.
7. **Scalability:** ML-driven automation and optimization enable WSNs to scale efficiently to accommodate growing data volumes and network complexity.

Overall, integrating machine learning into wireless sensor networks enhances their intelligence, adaptability, and performance, making them more effective in various applications across dynamic environments.

1.6 Internet Of Vehicles (IOV)

Recently, there was much interest in Technology which has emerged greatly to the development of smart city. The Internet of Things (IoT) encompasses a global network linking countless small objects, enabling seamless communication between them. Within this vast network, all interconnected devices have the capability to interact with one another. However, when we focus specifically on connecting these small objects, particularly vehicles, over the internet, we designate it as the Internet of Vehicles (IoV). With the rapid growth of urban populations comes a corresponding surge in vehicle ownership, presenting significant challenges. Consequently, there arises a pressing demand for innovative traffic management

solutions based on IoV technology to address the complexities of modern transportation systems [39].

1.6.1 IOV's Definition

The Internet of Vehicles (IoV) represents a dynamic convergence of technologies, including Vehicular Ad-hoc Networks (VANETs), Internet of Things (IoT), and mobile computing. This emerging field envisions vehicles as intelligent nodes interconnected through the internet, facilitating seamless communication and data exchange. In this context, vehicles interact not only with each other but also with pedestrians and roadside units, creating a comprehensive network for information sharing and collaboration [39]. Moreover, the IoV has captured significant attention in the realm of intelligent transportation systems, promising to revolutionize the way vehicles connect with each other and with infrastructure. It aims to establish robust network connections, integrating various communication technologies to enhance the standard architecture of vehicle communication [40]. This transformative potential of IoV has spurred rapid market growth, with major automotive manufacturers such as BMW and Tesla already incorporating self-driving features into their vehicles. Furthermore, companies like Uber and Google are actively exploring self-driving mechanisms, indicating the widespread adoption and investment in IoV technologies [39]. As the automotive industry continues to evolve, it consistently introduces new features aimed at enhancing both comfort and safety on the roads. The Internet of Vehicles comprises the following elements: Vehicles,

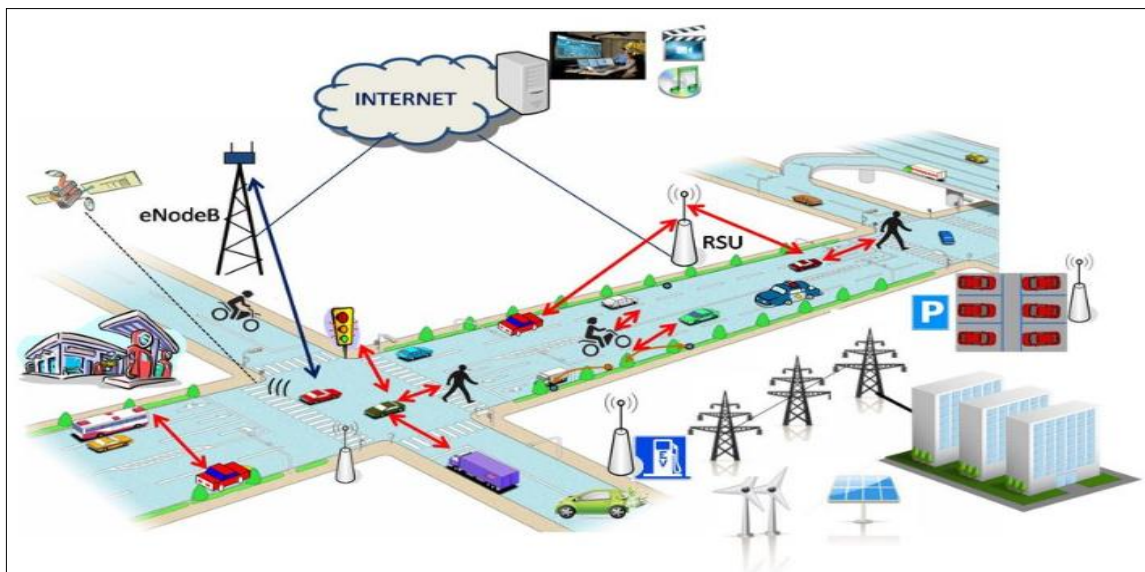


Figure 1.18: The Internet of Vehicles scenario [41].

Sensors, Roadside Units (RSUs), Infrastructure, Personal devices, and Human involvement.

1.6.2 Communication architectures of IOV

In IoV, the devices are vehicles, sensors, personal devices, cloud servers and infrastructure devices such as RSU and many more. In the literature, several different types of communications are discussed for IoV. The following illustrates the five of them (*Figure 1.19*) [42]:

1. *Vehicle-to-Vehicle (V2V) Communication*: This involves vehicles exchanging data concerning speed, direction, location, braking, and stability to prevent accidents, alleviate congestion, enhance fuel efficiency, and optimize routes.
2. *Vehicle-to-Infrastructure (V2I) Communication*: This entails data exchange between vehicles and roadside infrastructure like lane markings, road signs, traffic lights, and RSUs. V2I facilitates the development of applications aimed at bolstering safety, mobility, and environmental benefits.
3. *Vehicle-to-Cloud (V2C) Server Communication*: This encompasses the exchange of information between vehicles and cloud servers, supporting applications such as security, firmware updates, and entertainment.
4. *Vehicle-to-Pedestrian (V2P) Communication*: This includes communication between vehicles and pedestrians to avert potential accidents. V2P enables interaction between vehicles and pedestrians through applications on personal devices like smartphones and wearables.
5. *Vehicle-to-Sensor (V2S) Communication*: This facilitates communication between sensors and vehicles, crucial for functions like collision protection by detecting obstacles and individuals in a vehicle's path, thus enhancing safety against accidents.

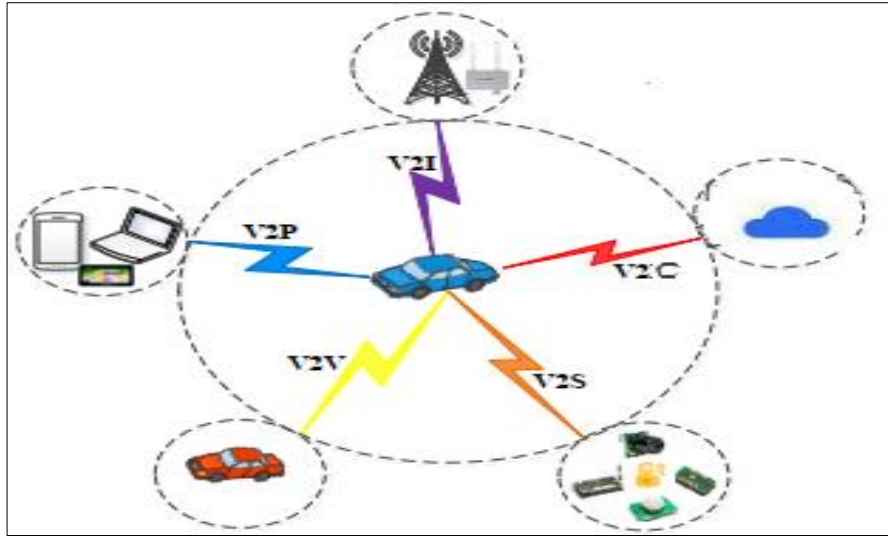


Figure 1. 19: Five types of vehicular communications of IOV.

1.6.3 IOV-layered architecture

Several variants of IoV layered architectures are presented, where the most comprehensive has 5-layers: *Objects (perception) (OL)*, *Connection (CL)*, *Artificial Intelligence (AIL)*, *Application (AL)* and *Business (BL)* layer. These layers are different from those of the classical TCP/IP architectures, but the layering principles are still preserved.

- a) Perception layer:** The first layer of the architecture is represented by the different types of sensors and actuators attached to vehicles, RSUs, smartphones and other personal that collect information from the various elements (speed, position, tire pressure, oil pressure, direction, pollution levels, collision detection, forward obstacle, side obstacle, , temperature, etc). The primary responsibility of the layer is to gather information regarding vehicle, traffic environment and devices.
- b) Connection layer:** The second layer of the architecture is represented by a virtual universal network coordination module for heterogeneous networks involving WAVE, Wi-Fi, 4G/LTE and satellite networks, through which the perceived information from the lower layer is securely transferred to the artificial intelligence layer for processing. The connection layer ensures *interoperability* with all the available networks to support all the communication models (V2V, V&R, V&P, V&I).

- c) **Artificial Intelligence Layer:** The *artificial intelligence* layer is represented by a generic virtual cloud infrastructure, working as an information management centre. It stores, processes and analyzes the information received from the lower layer and then takes decisions. Its major components are: *Vehicular Cloud Computing (VCC)*, *Big Data Analysis (BDA)* and *Expert System*.
- d) **Application Layer:** The *application* layer (AL) contains smart applications (e.g., for traffic safety and efficiency, multimedia-based infotainment and web based utility). The AL include safety and efficiency applications (VANET legacy) and provides smart services to End Users (EU) based on intelligent analysis.
- e) **Business Layer:** The fifth layer of the architecture is represented by the operational management module of IoV. The major responsibility of the layer is to foresight strategies for the development of business models based on the application usage data and statistical analysis of the data. Different types of analysis tools including graphs, flowchart, comparison tables, use case diagram, etc., are the major part of the layer.

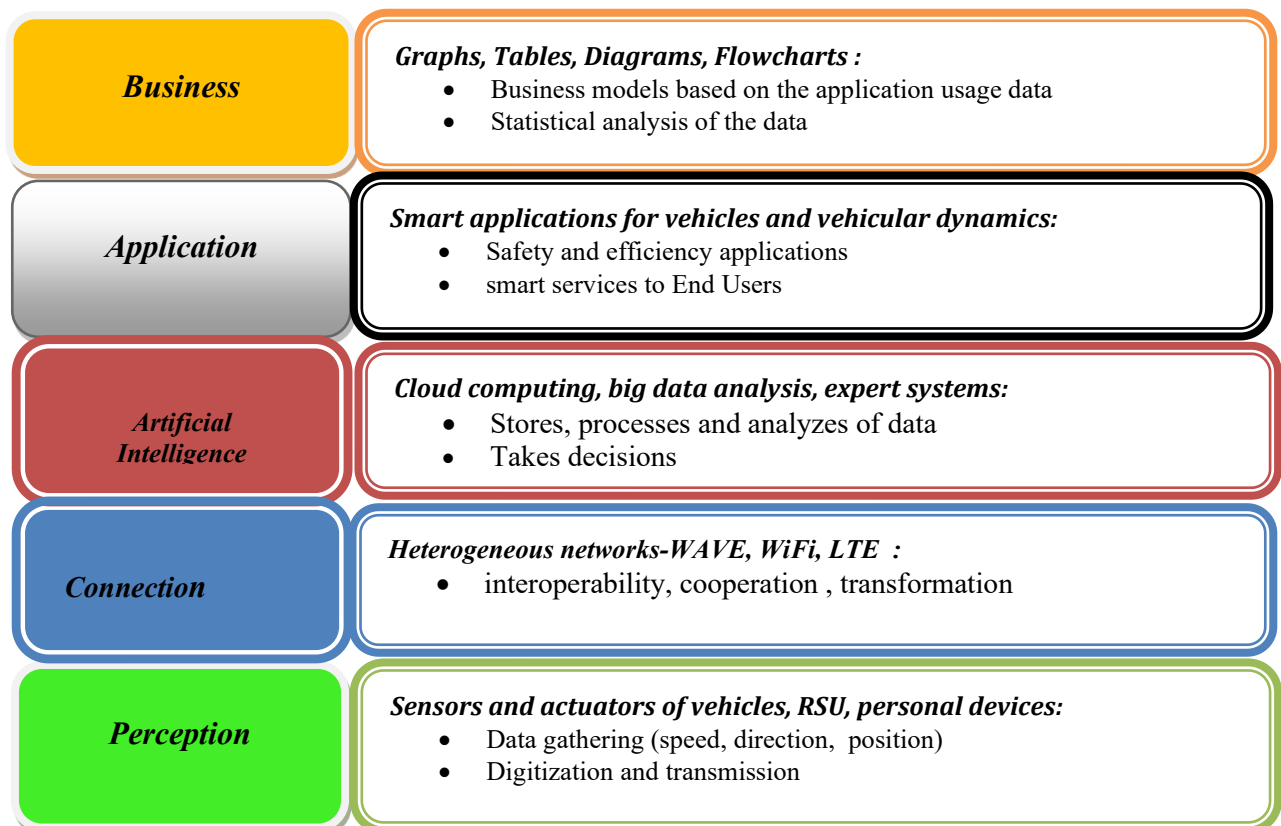


Figure 1 .20: The five-layers architecture of IoV.

1.6.4 Challenges in IoV

The IoV system has to face different types of hurdles before it is adopted. All the critical issues need to be resolved before it is successfully adopted in the autonomous vehicle market. Some of the selected critical challenges in IoV are listed below [43].

- a) *Delay Constraints*: Timely delivery of safety-related messages is crucial in IoV applications, particularly during emergencies. Strict delay constraints are necessary, requiring minimal delay even at the cost of other factors. However, achieving such efficiency proves challenging with the current communication infrastructure.
- b) *Lack of Standards*: The absence of proper communication standards hinders the seamless exchange of information in IoV systems. Developing comprehensive standards is essential for transparent integration with existing protocols and further progress in system development. Integration of different communication systems is key to overcoming this obstacle.
- c) *Network Connectivity*: Poor and unstable network connectivity in remote areas poses a significant challenge to IoV systems. Reliable connectivity forms the backbone of such systems, necessitating intelligent and sustainable solutions to improve connectivity, particularly in rural regions.
- d) *Fault Tolerance*: IoV systems must exhibit fault tolerance, ensuring uninterrupted communication even in the presence of malicious vehicles. Establishing highly reliable communication networks is crucial for maintaining system integrity.
- e) *Interoperability*: Achieving interoperability among vehicles is a pivotal challenge in IoV systems. Addressing various interoperability issues, including handoff timing and optimal wireless network technology selection, is essential. Well-organized and scalable management and communication among vehicular nodes are necessary for effective interoperability.
- f) *Security and Privacy*: The integration of diverse technologies, services, and standards in IoV underscores the critical importance of data security and privacy protection. As an open and public network, IoV is susceptible to intrusions and cyber-attacks, posing risks of physical damage and privacy breaches.

1.6.5 IOV Applications

The Internet of Vehicles (IoV) encompasses a wide array of applications, including [44]:

- *Enhanced Safety*: This involves collaborative collision avoidance systems that utilize sensors to detect potential collisions and issue warnings to drivers. It includes the transmission of periodic status updates and emergency messages triggered by critical events like accidents, traffic congestion, or adverse road conditions.
- *Traffic Management*: IoV promises to revolutionize urban congestion management, transportation logistics, and urban traffic patterns, thereby influencing our collective lifestyle and commuting experiences.
- *Rapid Crash Response*: Connected vehicles can automatically transmit real-time crash data, including the vehicle's precise location, to emergency responders. This capability accelerates emergency response times, potentially saving lives in critical situations.
- *Convenient Services*: Remote access to vehicles enables a range of services such as remote door unlocking and stolen vehicle recovery. Moreover, IoV facilitates transportation agencies' access to up-to-date traffic, transit, and parking data, streamlining the management of transportation systems and reducing congestion.
- *Infotainment*: Connected vehicles offer a plethora of in-vehicle entertainment options, delivering streaming music and information directly through the dashboard, enhancing the driving experience with online connectivity and multimedia content.

Additional applications encompass electronic toll collection, traffic guidance systems for safe navigation, autonomous vehicles, intelligent vehicle control mechanisms, preemptive crash avoidance systems, real-time traffic flow monitoring, and advancements in vehicle autonomy technologies.

1.7 Cloud -Fog And Edge Computing

Cloud computing stands as the predominant method for managing IoT data, yet alongside it, fog and edge computing have gained substantial traction. These alternative approaches aim to enhance data processing speed and efficiency, while also bringing intelligence closer to the IoT devices responsible for generating and acting upon data, such as sensors and actuators. In this discussion, we will conduct a side-by-side comparison of these three data technologies, delving into their distinctions, characteristics, and applications within the realm of Internet of Vehicles (IoV).

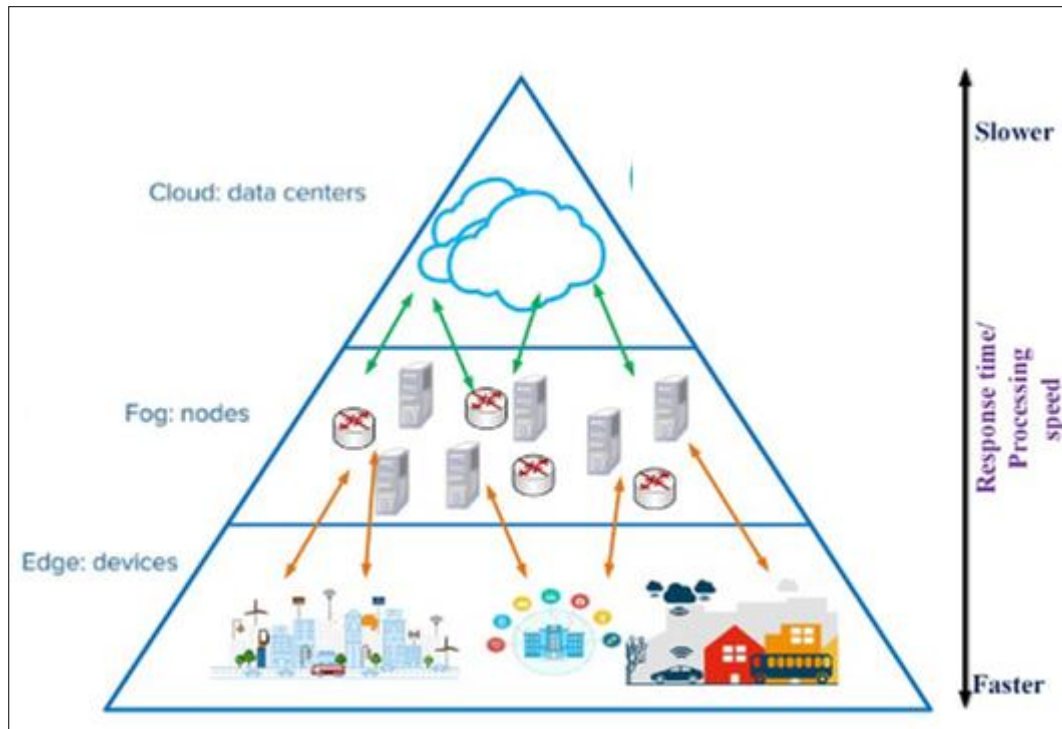


Figure 1 .21: Cloud, fog and edge architecture.

1.7.1 Cloud computing:

Cloud Computing, an emerging technology in various fields, operates through internet-based computing. It facilitates hosting and delivering diverse software and services over the internet, tailoring computational resources to user demands and requirements. These resources encompass extensive storage, high-performance servers, diverse operating systems, and network capabilities [45]. At its core, cloud computing involves storing and accessing data and programs via the internet from remote locations, diverging from local storage and computing. This remote environment offers properties like scalability and elasticity, making it distinct from a mere remote machine. The metaphorical cloud represents the internet, enabling data and program access from anywhere, anytime, and via any device [46].

a) Cloud Essential Characteristics

Cloud computing possesses five essential characteristics [46], presented below:

1. *On-demand self-service*: Users can autonomously procure computing resources, like server time and network storage, without direct interaction with service providers.
2. *Broad network access*: Services are accessible over the network via standard mechanisms, accommodating various client platforms, from mobile phones to laptops.
3. *Elastic resource pooling*: Providers aggregate resources to serve multiple users through a multitenant model, dynamically allocating physical and virtual resources based on demand. Users generally lack control over the exact resource location but may specify it at a higher level of abstraction.
4. *Rapid elasticity*: Resources can be swiftly provisioned and released, scaling out or in as needed. Users perceive these capabilities as unlimited and purchasable in any quantity at any time.
5. *Measured service*: Cloud systems employ metering capabilities to optimize resource utilization. Usage is monitored, controlled, and reported, ensuring transparency for both providers and consumers across various services, such as storage, processing, and bandwidth.

b) Cloud Service Offering Models

The cloud offers three main service models: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS), collectively known as the service-platform-infrastructure (SPI) model of the cloud.

1. *SaaS*: Users access applications hosted by providers over the internet, with the provider managing the underlying infrastructure. Examples include CRM and business analytics software.
2. *PaaS*: Users deploy their applications on the cloud infrastructure using supported programming languages and tools. The provider manages the infrastructure, offering scalability and maintenance. Examples include Google App Engine and Microsoft Azure Services.
3. *IaaS*: Users provision fundamental computing resources on a pay-per-use basis, including processing, storage, and networks. Users have control over the operating systems and applications, while the provider manages the underlying infrastructure. Amazon Web Services (AWS) is a notable IaaS provider.

c) Challenges & Issues

Cloud computing, despite offering numerous advantages, also faces various challenges and issues. Here are some of the main ones [47]:

1. *Security*: Data security is one of the primary concerns in cloud computing. Companies are often reluctant to move their sensitive data to cloud environments due to concerns related to privacy, regulatory compliance, and the risk of cyberattacks.
2. *Compliance*: Companies must adhere to industry-specific regulations, which can complicate the migration to the cloud. Cloud service providers must also ensure they comply with compliance standards for data stored in their data centers.
3. *Integration*: Integrating existing systems with cloud services can be a challenge. Companies may encounter difficulties connecting legacy applications to cloud services, leading to compatibility issues.
4. *Performance*: Although cloud service providers have significantly improved their performance, some users may still experience delays due to network latency or other performance-related issues.
5. *Availability*: Dependency on cloud service providers means that companies are vulnerable to service outages or unavailability. This highlights the importance of having business continuity and disaster recovery plans in place.
6. *Unexpected Costs*: While cloud computing can offer cost savings; it can also result in unexpected costs. Companies need to be aware of fees related to bandwidth, additional storage, and other complementary services.
7. *Resource Management*: Managing resources in the cloud can be complex, especially in multi-cloud or hybrid environments. Optimizing resource utilization while avoiding waste is a constant challenge.
8. *Data Sovereignty*: Some countries have strict laws governing data storage and processing, which can pose data sovereignty issues for companies using cloud services located in other jurisdictions.
9. *Scalability*: While the cloud offers great scalability, some applications may encounter scaling issues, especially if they were not designed to operate in distributed cloud environments.

10. *Governance*: Establishing effective governance policies for data management, compliance, and security can be complex, especially in cloud environments where resources are shared.

It is important to note that these challenges vary depending on the specific nature of the business, its needs, and its cloud adoption strategy. Organizations must develop well-thought-out strategies to mitigate these challenges and maximize the benefits of cloud computing.

1.7.2 Fog computing

Fog computing is a model that extends cloud computing to the edge of a network, allowing for compute, storage, and networking services to be provided between end devices and cloud data centers. It aims to address the latency and bandwidth issues of current cloud systems, as well as the scalability problems caused by the growth of smart devices. Fog computing adds another layer between edge devices and the cloud core, providing low latency and real-time interactive applications. It distributes computing, storage, control, and networking functions closer to users along a cloud-to-thing continuum. Fog devices, positioned between the cloud and smart devices, enable real-time applications, location-based services, and mobility support. Cisco has promoted the concept of fog computing in various areas, such as smart grid, connected vehicles, and wireless sensor networks.

a) Application Areas of Fog Computing

Fog computing, as highlighted by Cisco, is poised to significantly impact several critical domains. These include:

1. *Smart Cities*: Fog computing enables real-time processing of data from various sensors and devices in urban environments, facilitating efficient management of resources and services.
2. *Healthcare*: Fog computing enhances healthcare systems by enabling remote patient monitoring, facilitating timely medical interventions, and improving overall patient care delivery.
3. *Industrial Internet of Things (IIoT)*: Fog computing supports the IIoT by providing localized processing capabilities for data generated by industrial sensors and devices, thereby improving operational efficiency and enabling predictive maintenance.

4. *Transportation*: In the transportation sector, fog computing enables intelligent traffic management, enhances vehicle safety through real-time data analysis, and supports the development of autonomous vehicles.
5. *Retail*: Fog computing facilitates personalized customer experiences in retail by analyzing customer data in real-time, optimizing inventory management, and enabling targeted marketing strategies.

1.7.3 Edge computing

a) Definition

Edge computing, involves performing computation at the edge of the network, involving downstream data for cloud services and upstream data for IoT services. This "Edge" encompasses any computing and network resources between data sources and cloud data centers. Examples include smartphones, gateways in smart homes, Micro Data Centers (MDCs), and Cloudlets. The core concept is to conduct computing close to data sources for efficiency. While Edge computing and Fog computing are often used interchangeably, Edge computing emphasizes the Things side, while Fog computing focuses more on infrastructure. Edge computing is expected to have a significant societal impact similar to Cloud computing. Things can serve as both data consumers and producers. At the edge, they can request services and content from the cloud and perform computing tasks locally. Edge computing entails tasks like computing offloading, data storage, caching, processing, and distributing services from cloud to users. The design of the edge must meet efficiency requirements in areas such as reliability, security, and privacy protection [48].

b) Vehicular edge computing

VEC, or Vehicular Edge Computing, combines modern Mobile Edge Computing (MEC) with conventional vehicular networks. VEC's primary goal is to relocate communication, computing, and caching resources closer to vehicular users. This approach offers the potential to address the growing demands of edge devices for low latency and high bandwidth by bringing resources nearer to where they're needed. Unlike traditional MEC setups, VEC faces unique challenges due to the fast mobility of vehicles, resulting in frequent and dynamic changes in network topology. Additionally, it must contend with complex communication

characteristics arising from the rapidly changing channel environment over time. In VEC, vehicles possess specific communication, computation, and storage resources. Roadside Units (RSUs), acting as edge servers, are strategically positioned near vehicles to collect, process, and promptly store data. Given the limitations in capacity, vehicles have the option to transfer their computation-heavy and time-sensitive tasks to these edge servers. This offloading process significantly reduces response times and effectively eases the strain on backhaul networks [49].

c) Vehicular edge computing architecture

Typically, this architecture is composed of three layers, i.e., vehicular terminals as the user layer, RSUs as the MEC layer and cloud servers as the cloud layer.

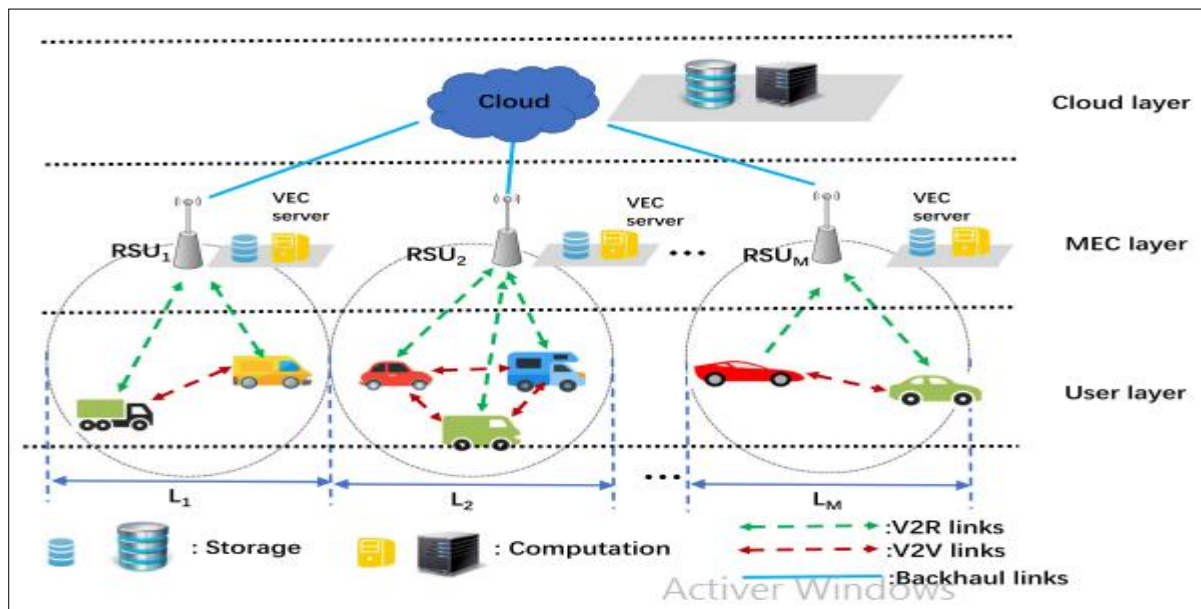


Figure 1. 22: Architecture of vehicular edge computing.

- **Vehicular terminals**

In VEC, vehicular terminals, primarily vehicles, exhibit distinct characteristics compared to regular mobile nodes. These features include:

Sensing: Vehicles are equipped with various sensors such as cameras, radars, and GPS, enabling them to gather information from both inside and outside their environment.

Communication: Vehicles can exchange data with other vehicles or Roadside Units (RSUs) through V2V (vehicle-to-vehicle) and V2R (vehicle-to-RSU) communication methods.

Computing: Apart from offloading computation tasks to edge servers or the cloud, vehicles can also handle some tasks locally.

Storage: Vehicles have idle storage space that can be utilized to cache popular content, facilitating data sharing.

- **Edge servers**

Typically represented by RSUs in VEC, are strategically positioned along city roads. These servers boast abundant communication, computation, and storage capabilities compared to vehicles. Their main functions include receiving, processing, and potentially uploading data sent by vehicles. Utilizing computation offloading and caching techniques, RSUs effectively manage stringent performance demands. Additionally, they offer a range of services to vehicles, including video streaming, traffic management, and navigation assistance.

- **Cloud servers**

Cloud servers are situated in a distant cloud environment and are capable of receiving data from edge servers. In contrast to edge servers, cloud services possess extensive computation and storage capabilities and cover a wider geographic area. By gathering data from mobile nodes and edge servers, cloud servers gain a comprehensive overview of the covered region. This cloud-based approach enables global-level management and centralized control, facilitating optimal decision-making processes.

d) Vehicular edge computing advantages

The key advantages of VEC are as follows.

1. **Response time:** in VEC, comprising offloading, processing, and return time, is significantly reduced due to the proximity of edge servers to vehicular users, particularly advantageous for safety-critical applications.
2. **Energy efficiency:** Smart vehicle proliferation leads to a surge in diverse vehicular applications, causing significant energy consumption, yet electric vehicles with VEC assistance can effectively support them despite limited energy capacity.
3. **Bandwidth:** Smart vehicle data explosion necessitates diverse content requests, challenging cloud computing's bandwidth; VEC relocates resources to the network edge, relieving back-haul networks' strain efficiently.

4. **Storage:** In VEC, edge servers near vehicular users store data, facilitated by caching technology, alleviating storage pressure on distant clouds.
5. **Proximity services:** VEC facilitates proximity services for vehicular users by leveraging edge servers, ensuring enhanced user experience while efficiently managing traffic.

1.7.4 From Cloud Computing To Fog And Edge Computing

Cloud computing enables convenient access to shared computing resources over the network, but faces challenges like bandwidth and latency issues due to increasing access devices. Fog computing extends cloud services to edge networks, providing closer proximity to user devices and emphasizing local data storage and processing. Similarly, edge computing allows computation at the network's edge, closer to data sources. While fog computing relies on interconnection among nodes, edge computing operates in isolated edge nodes, offering services near data sources to meet critical requirements like real-time optimization and security. Both fog and edge computing supplement traditional cloud computing by providing compute, storage, and networking services between end devices and the cloud [50].

1.7.5 Cloud-edge computing for the Internet of Vehicles (IoV)

Cloud-edge computing is a collaborative approach that combines the capabilities of cloud and edge computing for the Internet of Vehicles (IoV). It involves offloading computational tasks from vehicles to cloud data centers or edge devices to improve resource utilization and minimize energy consumption. Various computational offloading schemes have been proposed to manage the coordination among IoT devices, the cloud, and edge computing in the context of IoV. Serverless vehicular edge computing (VEC) has emerged as a promising execution paradigm for time-sensitive applications in connected vehicles. It aims to offload computational load to edge devices, such as roadside units, using serverless computing technology. Additionally, a joint communication and computational resource allocation mechanism has been proposed to enhance resource utilization and scalability in VEC-enhanced IoV.

1.8 Conclusion

This chapter introduces the main concepts addressed in this thesis. These include Machine Learning, along with various classic algorithms, Deep Learning and its subtypes with their use cases, Reinforcement Learning, and its integration with Deep Learning to create more powerful algorithms for solving complex dynamic systems. Additionally, we delve into the technology of WSNs, Internet of Vehicles and Computing. We also provide an overview of Cloud, Fog, and Edge Computing.

The Intelligent Transportation System (ITS) is a vast domain where road safety is a crucial challenge. The application of artificial intelligence techniques in this domain has seen significant growth recently, especially with the emergence of new machine learning algorithms such as Deep Learning and Reinforcement Learning.

For this reason, in the following chapter, we aim to provide an overview of the state-of-the-art works that combine AI and Intelligent Transportation Systems in general and specifically focus on machine learning applied to road accident prediction.

Chapter (2): AI in accident risk prediction- State of the art

2.1 Introduction

Many researchers have explored the application of machine learning in traffic safety and accident risk prediction over the past few decades. This thesis primarily focuses on two key machine learning algorithms, namely deep learning and deep reinforcement learning, which are primarily applied to accident risk prediction in addition to classical methods.

In this chapter, we provide an overview of the relevant state-of-the-art for each type of algorithm and how they are utilized in the accident risk prediction problem. We categorize previous works in each sub-field based on the techniques used, and summarize all the relevant state-of-the-art in a table at the end of each section.

2.2 Traffic Accident risk Prediction Using Classical Techniques

Numerous researchers have approached the prediction of traffic accidents risk by considering it as either a classification problem or a regression problem. In this section, we will explore several studies that have utilized classical machine learning techniques to address this problem.

The research presented in [51] discusses the development of accident prediction models using linear and logistic regression. It identifies suitable regression models for predicting fatalities based on the total number of accidents and analyzes specific scenarios like accidents at T-junctions and those involving alcohol. A logistic regression model is created for accidents from 2014 to 2019 to forecast fatalities. The research concludes by validating the effectiveness of the regression models in road safety modeling and accident prediction. In the same context, the paper [52] introduces a new framework for assessing car accident risk using ordinal regression, focusing on factors like weather and road conditions. It addresses the challenge of including data where accidents did not happen, using multinomial logistic regression. The study compares four ordinal regression methods using real accident data from the US and UK, finding that adjusting data quantity and using a specific approach improves risk assessment accuracy. The authors in [53] employed diverse analytical methods to bolster road safety measures. They utilized Kernel Density Estimation to pinpoint blackspots based on injury severity levels and develops a Multinomial Logistic Regression model to predict crashes involving vulnerable road users. Spatial and temporal analysis uncovers patterns in VRU-involved crashes, aiding

in the formulation of effective safety policies. Factors influencing injury severity of VRUs are assessed, alongside the identification of peak traffic hours impacting VRU injuries. Overall, the research offers insights crucial for targeted interventions and proactive accident prevention strategies, emphasizing the importance of nuanced spatial and temporal considerations in road safety initiatives. This paper [54] addresses the underreporting of African Road Traffic Injuries (RTIs) and identifies contributing factors to accidents in Addis Ababa, Ethiopia, such as pedestrian faults, illegal driver behaviors, and nighttime crashes. It utilizes an ordinal logistic regression model to analyze injury severity levels, providing insights into key determinants of crash outcomes. The findings emphasize the need for measures like strict law enforcement and driver training to enhance road safety and reduce injury severity. In the paper [55] the authors developed a predictive model for traffic accident severity on Indian highways using the random forest algorithm. a multi-step methodology was employed, involving data collection and preparation, feature selection, training a random forest model, tuning parameters, and evaluating the model using accuracy and F1 score, Their model achieved an overall accuracy of 67% on the training set and 41.47% on the test set.

Support Vector Machine (SVM) has been used in multiple studies for road accident risk prediction. The paper [56] proposes a practical model using a mixed-support vector machine (SVM) with heuristic algorithms to forecast traffic casualties. Ten variables including time characteristics, weather factors, accident types, collision characteristics, and road environment conditions are considered. The SSA-SVM model is found to be most effective compared to other algorithms (GA-SVM, GWO-SVM, PSO-SVM) in casualty prediction. The study focuses on urban road traffic accidents, aiming to enhance prediction accuracy using SVM. Various attributes such as Week, Period, Weather, Road Conditions, Alarm Categories, Active Hit, Passive Hit, Collision Type, Road Section, and Road Type are analyzed for this purpose. This study [57] develops a severity causation network using information entropy and Bayesian network to analyze the relationship between risk factors and crash severity. Key factors for severity prediction are identified and used to predict injury and property damage levels. Selective factor utilization enhances prediction accuracy and operational efficiency in crash response, aiding in traffic safety improvement and casualty reduction. The findings offer valuable insights for crash severity analysis and response planning. Authors in this research [58] focused on analyzing road accident severity through the application of Support Vector Machine (SVM) and Artificial Neural Network (ANN) algorithms. Leveraging a Traffic-crashes dataset, the study aims to predict the severity of road accidents. Findings indicate a

notable discrepancy in accuracy rates, with the SVM algorithm achieving a significantly higher accuracy of 83.6% compared to the ANN algorithm's 73.3% in predicting accident severity. This highlights the SVM algorithm's potential in accurately forecasting road accident severity, offering valuable insights for the development of effective accident prevention strategies.

Moreover, SVM was employed in an image classification approach to discern vehicles susceptible to accidents, showcasing its efficacy in accident analysis. In this study [59], the authors investigate the accident's impact and identify vehicles prone to accidents through image classification facilitated by machine learning. This method endows the system with the capability to autonomously learn and refine from the provided dataset without requiring human intervention or assistance.

Another study [60] explores the use of Support Vector Machine (SVM) in safety critical applications, particularly in car crash scenarios like airbag control systems. Its objective is to demonstrate how modern products utilize machine learning (ML) to prevent unnecessary airbag deployment. Car safety applications heavily rely on detection algorithms and sensor systems for timely and accurate decision-making based on sensor signals. The study employs a multiclass SVM to enhance classification accuracy in full frontal crashes, comparing two methods: One-Versus-Rest and One-Versus-One, with the latter performing better. This improved classification facilitates the implementation of active and passive occupant safety features in the automotive industry. Overall, SVM has shown promise in road accident risk prediction, but its performance may vary depending on the specific context and factors considered in the analysis.

By using several classic methods at the same time, the authors in the paper [61] tackle a pressing concern in road safety by employing machine learning techniques to predict the causes of traffic accidents. Their study is centered on analyzing the frequency of accidents and exploring the interplay between traffic incidents, road conditions, and environmental factors. To achieve this, the paper develops a robust accident prediction model leveraging machine learning algorithms such as Decision Tree, Random Forest, and Logistic Regression. By incorporating various elements such as weather, vehicle condition, road surface condition, and light condition, the model aims to accurately predict road accidents and forecast their severity. The evaluation of these models is conducted using key performance metrics including accuracy, precision, recall, and F-score, providing a comprehensive assessment of their predictive capabilities and effectiveness in addressing the challenges of road safety.

This study [62] investigates the impact of traffic accidents and injury severity considering three aspects: characteristics of daily travelers (e.g., age, sex), vehicle attributes (e.g., type, transmission), and road conditions (e.g., pavement quality, intersection type). A proposed model [63] utilizes Arduino boards (ATmega328) and IR sensors to detect and prevent traffic accidents in two stages: initial accident detection using IR sensors and subsequent alerting of individuals via GSM (SIM900D) module messages. The study [64] compares injury severity prediction methods, including statistical models (multinomial logit and ordered probit) and machine learning algorithms (k-nearest neighbors, decision tree, random forest, support vector machine), using accident data from Florida. The evaluation aims to identify effective approaches for predicting accident injury severities. In [65] a strategy is proposed to address traffic congestion and reduce accidents through speed and lane-changing control systems. This includes measures such as lane management, speed regulation, and the dissemination of traffic information to enhance collision prevention and traffic flow management.

Road traffic accidents (RTA) are a major global concern, especially in low and middle-income countries, and are a leading cause of death in Rwanda. This study [66] used Random Forest (RF) and Support Vector Machine (SVM) models to predict short-term road accidents using police data. The models' performance was compared using MAE, MSE, RMSE, and the coefficient of determination (R^2). The RF model outperformed the SVM model, showing higher R^2 values. Advancements in traffic sensor technology have enhanced short-term crash risk prediction. This study [67] improves existing models using a random multinomial logit model for key predictors and a Bayesian belief net (BBN) for real-time prediction. Tested on Tokyo expressways, it forecasts hazardous conditions within 4-9 minutes for specific segments, correctly predicting 66% of crashes with a false alarm rate under 20%.

Table 2. 1. Related work of application of Classical algorithms for accident risk prediction

<i>Authors</i>	<i>DATA</i>	<i>Methods</i>	<i>Performances</i>
[1]	New Zealand road accident data from 2016 - 2020 (378820 rows)	RF , DT, AdaBoost, XGBoost, LGBM, CatBoost	Accuracy : RF = 81.45 DT = 74.12 AdaBoost = 65.61 XGBoost = 78.52 LGBM = 76.94 CatBoost = 69.68
[51]	India road accident data from 2014–2019 (118 rows)	Binary Logistic Regression	Cut-off probability = 0.33
[52]	US: road accident data from 2015 - 2016 (22380 rows and 17 features) UK: 2018 (14593 rows and 10 features)	LR, DT, L. AT, L.SVR, K-NN	Uk: MSE (DT) = 0.199 MSE (L. AT)= 0.396 MSE (LR)= 0.43 MSE (L.SVR) = 0.387 US : MSE (DT)= 1.145 MSE(K-NN)= 3.084 MSE (L . AT)= 1.338 MSE (L.SVR) = 2.247
[53]	4,439 road accidents in Portugal (2012-2015)	Kernel Density Estimation (KDE) Multinomial Logistic Regression (MLR)	Nagelkerke's R2 values of 37, 39, and 29 for Aveiro, Porto, and Lisbon
[54]	8458 road accidents of Addis Ababa, Ethiopia, from (2017 to 2020)	Ordinal logistic regression	15.1% fatal, 46.7% severe, and 38.3% slight injuries
[55]	3257 road accident in India with 32 attributes	Random forest	Accuracy of 67% , F1-score of 0.64
[56]	5000 road accidents of Wuhan	SVM with heuristic algorithms (GA, SSA, GWO, PSO)	ACC: SVM = 0.60 GA-SVM = 0.84 SSA-SVM = 0.86 GWO-SVM = 0.85 PSO-SVM = 0.85

[57]	>10,000 traffic accident datasets of USA	Naïve bayes, RF, MLP, Ada boost	NB – 74% RF – 77% MLP – 77% AB – 75%
[58]	Dataset of traffic accidents LasVegas region in the year 2016	SVM, ANN	ACC: SVM = 83.6% ANN= 73.3%
[62]	>300 datasets of traffic accidents	KNN, DT	KNN : 80.26% , DT :73.68%
[63]	5,000 datasets of traffic accidents	NN, DT, RF, SVM	Overall accuracy ranges from 44.7% to 80.5%
[64]	> 6,000 datasets of Michigan traffic Accidents	RF, LR, NB and Ad Boost	LR: 74.5% N B:73% Ad boost:74.5% RF: 75%
[65]	3,643 traffic accident data of China	Bayesian network and information entropy	overall accuracy ranges from 50 to 90%
[66]	Accident data from 2016-2019 in Rwanda whith an average of 5595 cases per year.	RF, SVM	RF: R2 = 0.918, MAE = 11.925, MSE = 268.386 and RMSE = 16.382 SVM: R2 of 0.866, MAE = 14.175, MSE = 440.268 and RMSE = 20.982.
[67]	6478 normal traffic condition data, Japan.	Random multinomial logit (variable selection), Bayesian network (model)	66% crash detection rate with less than 20% false alarm.

2.3 Deep Learning For Traffic Accident Prediction

Artificial Neural Networks (ANN) offer superior precision and forecasting capacity for road accident prediction, outperforming traditional statistical models. They provide flexibility and can model non-linear functions without statistical simulation. Deep learning approaches like Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN) have gained traction for their effectiveness in predicting road traffic accidents, attracting considerable interest from researchers across diverse scientific fields. Recently, with the rapid advancements and remarkable achievements in machine learning technologies, several recent studies have embraced deep learning methods for predicting traffic accidents.

The study [68] contributes to the existing literature on motorcycle crash severity by investigating various deep learning (DL) algorithms to forecast the severity of accidents involving motorcycles. It aims to determine the most effective DL algorithm for this purpose. The research focuses on predicting the severity of injuries resulting from motorcycle crashes in Wyoming between 2006 and 2016, spanning a decade. This study overview and summarizes the different forms of neural network models such as the single layer perceptron (SLP) neural network, the multilayer layer perceptron (MLP) neural network, the radial basis function (RBF) neural network, the recurrent neural network, and the convolutional neural network used as a prediction method for the severity of road crash injuries.

In order to enhance prediction accuracy, [69] introduced a novel model, the Traffic Accident Severity Prediction-Convolutional Neural Network (TASP-CNN), designed to consider interrelationships among various features of traffic accidents. The model incorporates the Feature Matrix to Gray Image (FM2GI) algorithm, which utilizes the weights of traffic accident features. This algorithm transforms individual feature relationships within traffic accident data into gray images, incorporating parallel combination relationships. These gray images serve as input variables for the TASP-CNN model. The Authors in [70] explores the efficacy of deep learning in predicting injury severity resulting from traffic accidents on Malaysian highways. Three distinct network architectures utilizing simple feedforward Neural Networks (NN), Recurrent Neural Networks (RNN), and Convolutional Neural Networks (CNN) were introduced. These architectures were optimized through grid search optimization to finely adjust the hyperparameters, aiming for models that provide accurate predictions with reduced computational costs. The findings revealed that, among the evaluated algorithms, the RNN model demonstrated superior performance. A novel Road Traffic Accident Prediction model

(TAP-CNN) is introduced in [71], which leverages factors such as traffic flow, weather, and light to construct a state matrix describing the traffic conditions and utilizes a Convolutional Neural Network (CNN) model. The accuracy of this new model is evaluated using samples, and the experimental findings indicate that the TAP-CNN model outperforms traditional neural network models in predicting traffic accidents.

The work in [72] employed human mobility data and historical traffic accident records in Japan to create a Stack Denoising Auto-Encoder (SdAE) model for assessing real-time traffic accident risk. This marked the initial application of deep learning for estimating traffic accident risk on a national level. However, the model generated only real-time accident risk maps, lacking suitability for near-future accident prediction. Additionally, it solely relied on GPS data from mobile phones, neglecting various other relevant factors. Long Short-Term Memory (LSTM) and the Fully Connected (FC) network were utilized in [73] to predict the probability of future traffic accidents in Beijing. They considered a number of significant variables, such as the direction and speed of traffic, as well as the weather. However, the road level did not allow for the specification of the prediction's granularity. The study [74] introduced the Hetero-Convolutional Long Short-Term Memory (HeteroConvLSTM) model for predicting traffic accident amounts in Iowa. This model incorporated spatial and temporal features, including time-invariant, time-variant, and spatial-graph features. However, the prediction was limited to a coarse-grained approach, constrained within grid cells spatially and daily prediction temporally. This limitation is not reflective of real-world situations, as traffic accidents typically occur near road networks. While the authors attempted to address this by using a road network mask layer to map the model's final outputs, the predicting process remained confined to grid levels due to the simplicity of the filter mask layer, which only performed a basic AND operation.

A novel Deep Spatio-Temporal Graph Convolutional Network (DSTGCN) is proposed in [75]. The model consists of three components: a spatial learning layer, a spatio-temporal learning layer, and an embedding layer. The DSTGCN outperforms traditional and state-of-the-art methods in predicting traffic accidents, as demonstrated by experimental results on real-world datasets. In [76] IOT and computer vision were used for the development of a driver safety monitoring system that aims to alert drivers when they become drowsy and update their condition in a centralized database. The system measures the driver's concentration level by analyzing factors such as drowsiness and the position of their face in the frame. To address the limitations of existing methods, robust facial landmark detectors are utilized to accurately

detect the driver's face and eyes. Algorithms are then employed to detect driver drowsiness and face position. The system averages the collected data to assess the driver's concentration level and utilizes their previous data to make predictions. This paper [77] presents an innovative data architecture influenced by images, designed to effectively capture intricate details of vehicular behavior at the microscopic level. To achieve this, a model for predicting accidents is developed for a specific segment of the Autopista Central urban highway in Santiago, Chile. The model is constructed by amalgamating multiple-input Convolutional Neural Networks and employs a combination of aggregated standard traffic data and the newly proposed image-inspired architecture.

The main objective in [78] is to develop a hierarchical deep learning-based model for estimating the risk of urban traffic accidents by considering factors such as aggressive driving, land use, and traffic facilities. The proposed model aims to overcome challenges in previous studies, such as the data imbalance problem and neglecting road environmental factors. The goal is to improve the estimation performance and provide a more accurate risk assessment, which can be used in traffic warning systems to prevent accidents and predict traffic accidents. Deep Convolutional Neural Networks (CNNs) in [79] trained on OpenStreetMap (OSM) images can accurately predict whether an area is high-risk or low-risk for road traffic accidents in Okayama Prefecture, Japan, based on the OSM image. Grad-CAM is also used to visualize the decision-making process of the trained CNN. A significance of traffic accident prediction for travel route design and urban safety is covered in the study [80]. It introduces the Spatial Gated Memory Network (SGMN), a proposed deep learning model to predict accident risk in urban areas. The model integrates real-time accident risk, traffic flow, and weather data to forecast high-risk sub-regions. The evaluation of the model's performance using real datasets demonstrates that SGMN surpasses commonly used memory neural networks such as RNN, LSTM, GRU, Convulsion, and Hereto-ConvLSTM.

The study [81] presents a combination of machine learning and deep learning models, specifically Random Forest and Convolutional Neural Network, referred to as RFCNN. Its purpose is to predict the severity of road accidents. The proposed approach is compared to various base learner classifiers. The analysis utilizes accident records from the USA spanning from February 2016 to June 2020. The results obtained demonstrate that RFCNN greatly improves the decision-making process and outperforms other models.

This article [82] presents a forecast of road accident severity using a novel deep learning approach that combines a deep-Convolutional Neural Network (D-CNN) and a Deep Recurrent

Neural Network (D-RNN). The identification of important features is accomplished through the utilization of CNN, enhancing the effectiveness of the models in terms of accuracy, specificity, and sensitivity. D-RNN stands as the most efficient and productive model, consistently yielding precise predictions in accident severity forecasting. The spacing between vehicles is identified as the most influential factor impacting the severity of accidents. The papers discussed in this subsection are summarized in *Table2.2*.

The primary drawback of deep learning lies in its dependence on stable historical data, which is not renewable with each new event. This reliance on fixed datasets can limit the adaptability of deep learning models in dynamic environments where conditions may change rapidly. Furthermore, deep learning algorithms necessitate considerable storage and processing resources to handle the large volumes of data involved, adding complexity and costs to implementation.

Table2. 2: Related work of application of DL for accident risk prediction

<i>Authors</i>	<i>DATA</i>	<i>Input (Independent variable)</i>	<i>Output (Dependent variable)</i>	<i>Method</i>	<i>Performances</i>
[68]	2,430 motorcycle crashes in a mountainous area in the United States over a 10-year period (2006 to 2016)	AADT, geometric characteristics, winter conditions, area types	Severity of motorbike accident	RNN , MLP LSP	RNN=0.74 MLP=0.58 SLP=0.53 Error rate: RNN=29 MLP=35 SLP=37
[69]	Traffic accident's data for an 8-year period (2009–2016) from the Leeds City Council	Street category, time of accident, number and type of vehicles, road surface, lighting conditions, climate conditions, casualty class, sex and age of casualty	Severity of traffic accident	TASP-CNN	Slight accident = 0.893 (Average Precision) serious accident = 0.248 (Average Precision) fatal accident = 0.063).
[72]	300 thousand records of traffic accidents throughout Japan from January 1, 2013 to July 31, 2013.	Traffic accident data includes attributes such as occurrence location, hourly occurrence time, GPS	Severity can be graded as three levels, that is, slight injury (level 1), heavy injury (level 2) and fatal (level 3).	Stacked denoise Autoencoder (SdAE),	MAE =0.96 ; MRE= 0.39 ; RMSE=1.00 .

[73]	The traffic accident records of Beijing in the year 2016	Traffic flow , traffic accident , geographical position weather , air quality , holiday , time period,	Severity level can be graded as slight injury (level 1), heavy injury (level 2) or fatal (level 3).	LSTM	MAE =0.58 ; MRE= 0.38 ; RMSE=0.69.
[75]	Traffic accident records of Beijing from 2018/08/01 to 2018/10/31	Traffic Accident Data (timestamps and locations), GPS, POI Data, Meteorological Data , Road Network Data	risk of next-period	DSTGCN	RMSE= 0.3439; PCC= 0.7445 ; Precision= 0.8213; Recall=0.8968; F1-Score= 0.8573 ; AUC= 0.8508 .
[78]	Data from 2019-01-1 to 2019-12-31 to do the experiments. The dataset consists of 3,650,000 cells.	Flow of lane, speed of lane, vehicle type	Binary classification (0 for no accident and 1 for accident).	CNN	Accuracy = 0.935 ; Precision= 0.848 ; Recall= 0.631 ; F-Score= 0.724.
[70]	The 2009–2015 traffic accident data for the North–South Expressway (NSE),Malaysia	Accident time, zone and location, collision type, surface and lighting condition, accident reporting	Injury severities	RNN ,CNN FFNN	Accuracy: RNN=73.76 CNN=70.30 FFNN=68.79
[79]	OpenStreetMap images of traffic accident data Okayama 2010 to 2021	Open Street Map images	Areas : "high-risk" or "low-risk".	CNN : VGGNet and ResNet	Accuracy: 90%
[80]	Datasets from New York City and Chicago	The input data includes weather data, historical traffic accidents, and traffic flow data.	Accident risk graph for the next hour.	SGMN	MSE=0.4418 RMSE=0.6646 ACC=64.88 SPR=27.42

[81]	4.2 million US car accident dataset from Feb 2016 to Jun 2020	Temperature, wind, humidity, visibility, wind direction, and other variables	Severity	RFCNN	• Accuracy: 0.991 • Precision: 0.974 • Recall: 0.986 F-Score: 0.980
------	---	--	----------	-------	---

2.4 Deep Reinforcement Learning for Traffic Safety

In recent years, deep reinforcement learning (DRL), an advanced form of artificial intelligence, has gained significant importance in intelligent decision-making across various domains. DRL has found applications in robotics [83], healthcare [84], Natural Language Processing [86], and sentiment analysis [85]. In the field of transportation systems, DRL algorithms have been widely utilized, particularly in traffic control tasks. For example, DRL has emerged as the most popular machine learning methodology for traffic signal control [87]. This paper examines the issues of persistent congestion in dense traffic networks and the limitations of traditional Adaptive Traffic Signal Control (ATSC) in addressing them. It presents Reinforcement Learning (RL) and Deep Reinforcement Learning (DRL) as potential solutions, emphasizing their use of on-line learning and Deep Neural Networks (DNN) to tackle the dimensionality curse and approximate quality functions. The authors in [88] study proposes using the double actors and regularized critics (DARC) method, a state-of-the-art reinforcement learning algorithm, for early accident prediction in autonomous driving. The DARC method outperforms existing state-of-the-art accident anticipation models. This work utilizes dashcam video as input and proposes the use of the DARC method for accident forecasting, resulting in predictions 5% earlier on average with improved precision.

A novel technique named DRIVE (Deep ReInforced accident anticipation with Visual Explanation) is presented in [89] which simulates both bottom-up and top-down visual attention mechanisms in a dashcam observation environment. The proposed stochastic multi-task agent can be visually explained by attentive regions, and the method uses dense anticipation reward and sparse fixation reward to train the model with an improved reinforcement learning algorithm. Experimental results show that the DRIVE model achieves state-of-the-art performance on multiple real-world traffic accident datasets.

The authors in [90] explore the use of an end-to-end learning technique for self-driving cars through reinforcement learning. A framework is created to test various reinforcement learning methods in a simulated setting, using simulated front camera images as input and generating steering angle, accelerator, and brake pedal position as output. The intended behavior is acquired through interaction with the environment, and the reward function is fine-tuned to prioritize staying in a lane at maximum speed. Several deep reinforcement learning methods are evaluated, with Soft Actor-Critic identified as the most effective in terms of learning speed and adaptability to new environments.

The work in [91] focuses on modeling the decision-making and interactions among different vehicles on highways. The authors utilized Double DQN (DDQN) to train the host vehicle, and the study was carried out using the open-source simulation platform "SUMO - Simulation of Urban Mobility". The driving environment consisted of three lanes and 20 cars randomly distributed on the highway. The host vehicle continuously monitors the distance between itself and any obstacles (such as other moving vehicles) ahead as it travels. If there is a decrease in the distance between successive measurements, the host vehicle begins to apply the brakes to avoid a collision. The algorithm then adjusts the host vehicle's speed accordingly.

To increase the safety of urban expressways, [92] suggested implementing a variable speed limit (VSL) system. It uses traffic data to assess crash risk in real time. When risk is high, the system activates VSL control to normalize traffic. This study addresses limitations in current VSL-based safety interventions. A CTM simulates traffic states based on existing detector spacing. A DBN is used in the RTCPM. Instead of predefined strategies, we use a deep Q-network for VSL control. The system reduced crash risk by 19% in the Tokyo Metropolitan Expressway.

The significance of traffic accident prediction for travel route design and urban safety is covered in the study [93]. Unexpected critical situations can lead to severe collisions. This work reviews chain collisions and proposes a reinforcement learning-based strategy to prevent and mitigate crash severity. The safety efficiency of existing methods in driving security is also analyzed. A perception network structure is used to enhance decision-making, and simulations assess algorithm efficiency in different traffic situations. The findings aim to highlight the benefits of reliable autonomous traffic systems.

In this research [94] the authors propose to train an agent to drive a simulated car in Unity ML-agents Highway by utilizing Policy Gradient. The approach produced good results and successfully trained the agent to navigate an environment that mimicked an autonomous car.

This paper [95] studied a DRL agent for highway driving by training an EV to learn a driving policy through interaction with simulated traffic. The DRL agent is based on a modified version of the DDQN algorithm which is considered a state-of-the-art RL algorithm for discrete state and continuous action space optimal problems.

Numerous accidents and fatalities occur due to reckless driving and increasing number of vehicles. Autonomous cars are seen as a potential solution. They need to be intelligent agents and capable of making decisions to prioritize people's lives. A proposed algorithm in [96] trains autonomous

vehicles to drive, overtake, and avoid collisions. The work is simulated using a TORC simulator and has shown expected performance.

In order to improve the deep reinforcement learning method for traffic light control and to perform information modeling, this paper [97] first uses deep learning to estimate municipal traffic flow. The results show that algorithm-optimized timing control of traffic signals has a far bigger effect than conventional timing control, lowering the probability of traffic accidents and bringing municipal traffic engineering closer to the level of intellectualization.

The DRL algorithm has proven highly efficient in solving complex decision-making problems that were previously beyond the capability of traditional machine learning techniques. However, when operating in a dynamic environment, such as in the case of traffic prediction and prevention, the algorithm requires frequent updates of the data being exploited in order to provide reliable predictions. Additionally, due to the significant storage and computing resources required, its application is best suited for deployment on a cloud platform.

Table2. 3: Related work of application of DRL for accident risk prediction

<i>Authors</i>	<i>Description</i>	<i>Framework/Environment</i>	<i>Algorithm</i>
[89]	Utilizes the Soft Actor-Critic (SAC) algorithm for accident anticipation with visual explanations.	PyTorch	Soft Actor-Critic (SAC)
[90]	Implements an end-to-end learning technique for self-driving cars through reinforcement learning.	Town03 map from Carla with Python	Q-Network (DQN)
[91]	Develops a safe driving policy for a highway scenario with randomly distributed vehicles.	SUMO	Deep Q-Network (DDQN)

[92]	Introduces a variable speed limit (VSL) system for enhancing the safety of urban expressways in real time.	NA	Deep Q-Network (DQN)
[93]	Proposes a reinforcement learning-based strategy to prevent and mitigate crash severity.	Unity3D	Proximal Policy Optimization (PPO), Soft Actor-Critic (SAC), Deep Deterministic Policy Gradient (DDPG)
[94]	For building an autonomous vehicle on highways through reinforcement learning.	Unity ML-Agents Toolkit	Policy Gradient
[95]	Proposes an RL method for ego cars (autonomous vehicles) to learn decision-making through simulated traffic.	NA	Deep Q-Network (DDQN)
[97]	Utilizes deep learning to forecast traffic flow and enhances the deep reinforcement learning algorithm for traffic signal control.	SCATS and SCOOT	Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM), Deep Q-Network (DQN)

2.5 Conclusion

The prediction of traffic accident risks is paramount for minimizing hazards and saving lives. Across the various research works cited above, a multitude of machine learning algorithms have been employed to establish an efficient traffic security system, spanning from classical machine learning models to newer algorithms inspired by neural network technologies such as deep learning and reinforcement learning.

The Deep Reinforcement Learning (DRL) algorithm has shown exceptional efficacy in addressing complex decision-making challenges previously beyond the capabilities of traditional machine learning techniques. However, in dynamic environments like traffic prediction and prevention, the algorithm requires frequent data updates to ensure reliable predictions, leading to a substantial amount of data that needs to be stored.

Our primary challenge lies in overcoming the constraints of computing and storage resources within vehicles to effectively used machine learning models. Moreover, the integration of Internet of

Vehicles (IOV), edge computing and cloud computing technologies can provide significant resources and real-time responses to drivers, thereby enhancing the Quality of Service (QoS) of the system. Nevertheless, combining complex concepts such as deep learning, reinforcement learning, IoV, and edge and cloud computing presents a significant challenge that demands careful consideration and innovative solutions.

Chapter (3): Collaborative Cloud.DRL - V. Edge.DL approach for Predicting Traffic Accident Risk

3.1 Introduction

In this chapter, we introduce our innovative approach employing machine learning techniques to aid drivers in adopting safer driving practices. Our primary objective is to anticipate accident risks in advance and notify drivers promptly to take appropriate action to prevent accidents. To achieve this goal, we have employed machine learning methodologies, specifically deep learning and deep reinforcement learning algorithms. Although deep learning and deep reinforcement learning are highly proficient in tackling intricate challenges within transportation systems, but they demand substantial resources for optimal performance. While the prevailing machine learning and Internet-of-vehicles (IOV) solutions rely predominantly on cloud infrastructure due to its virtually unlimited storage and processing capabilities, challenges such as internet disconnection and latency hinder its practicality. Vehicular edge computing (V.Edge.C) emerges as a solution to mitigate these limitations by leveraging resources in proximity to end users. Our research proposes an advanced system that harnesses the power of Deep Reinforcement Learning (DRL) to forecast traffic accident risks. The proposed approach entails conducting the training phase in the cloud, utilizing its vast storage and computational resources alongside a comprehensive historical accident dataset. Subsequently, the trained model is deployed to vehicular edge devices for real-time inference during accident scenarios. Further details on the adopted approach will be provided below.

3.2 Proposed System Architecture

Our main goal is to develop a framework that leverages machine learning techniques to help drivers in safe driving practices. We plan to achieve this by analyzing large amounts of data from previous accidents. The proposed framework consists of an intelligent and collaborative driver assistance system, called *Intelligent and Collaborative Cloud-V.Edge Driver Assistance System (ICEDAS)* that operates between the cloud and a vehicle's edge. Figure 3.1 illustrates the two layers in this framework, which work together to safeguard drivers and minimize the risk of road accidents.

3.2.1 Cloud Layer

Cloud computing is one of the most significant trends in the information technology evolution, as it has created new opportunities that were never possible before [98]. Due to its storage capacity and

computing power, we consider it the suitable location to generate the two machine learning models in our system. The first model is DRL, which is the main component in our framework. It runs on the cloud to predict accident risks. The second model is a DL model, also generated in the cloud and then deploys it to the V.Edge device for inference when needed, to cover the absence of prediction by the cloud.

3.2.2 V.Edge Layer

Vehicular Edge Computing (VEC), based on the edge computing motivation and fundamentals, is a promising technology supporting ITS services, and smart city applications [99]. In our system, the V.Edge is used to replace the cloud in certain cases, such as internet disconnection or bandwidth overload. Vehicles equipped with cameras, radars, GPS, and other devices can sense both the internal and external environment and collect various information such as speed, road quality, position, and more. These data are either sent to the cloud in real-time for prediction by DRL model, or used by the edge itself to replace cloud prediction in generating accident risk alerts using the inferred DL model.

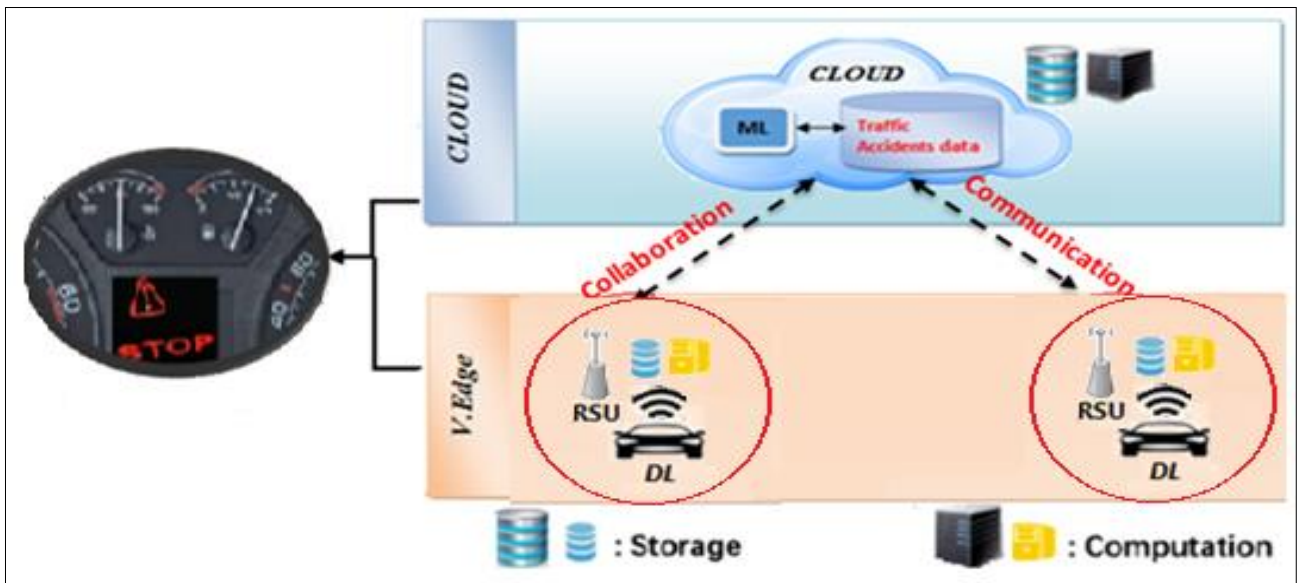


Figure 3.1: Cloud_DRL and V.Edge_DL system architecture

3.3 Cloud_DRL For Accident Risk Prediction

In this section, we present the details of the proposed Cloud_DRL based risk prediction system. We first define possible scenarios then we present the structure of DQN and explain in detail how it works to train the learning model based on accidents data available in the cloud.

3.3.1 Scenarios

Many accidents occur when driving conditions suddenly change. ICEDAS must detect the potentially accident-causing events in advance and help the driver take the appropriate actions to avoid them. To predict a traffic accident risk, we focus on many contributing factors that frequently cause traffic accidents. They are often related to Drivers, Roads or Vehicles such as Driver's age, Driver's Sex, Driver's experience, Road condition, Light condition, Weather condition, Type of vehicle, Service year, etc. When a sudden change in any of the car's normal conditions is accurately detected, it may be difficult to adapt properly to this change, which may become a threat to the car. In this case, we need an intelligent risk prediction system that adapts to different situations of this risk. Markov Decision Process (MDP) is a powerful technique for modeling sequential decision-making problems. We used MDP to formulate our problem. In the MDP framework, an agent interacts with a given environment state by taking actions at discrete time steps. In our system, we assume that the traffic environment follows the discrete-state. Figure 3.2 describes this Markov process. The state (SN-risk) implies that the system did not detect any risk. Once a risk is detected, the state (SN-risk) is changed to the state (S-risk). In practical scenarios, it is difficult to know the transition probabilities of the Markov process and the distribution of the environment states. Therefore, reinforcement-learning approach can be applied to learn the risk prediction policy through the interaction with the environment.

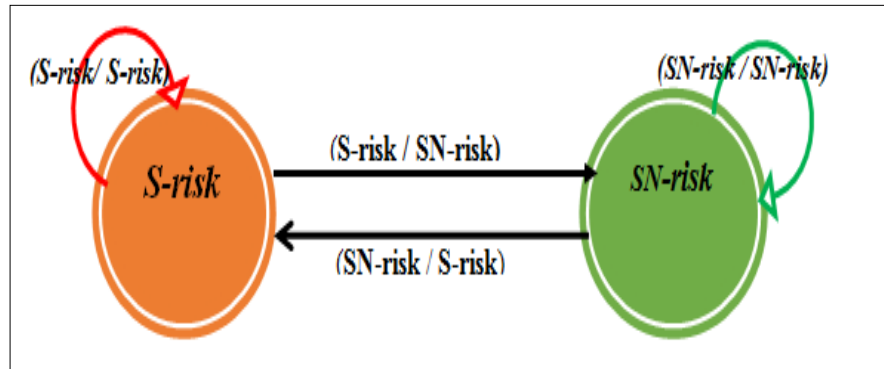


Figure 3.2: Traffic environment-state description by the discrete-state Markov process.

3.3.2 Key Elements of Cloud _DRL

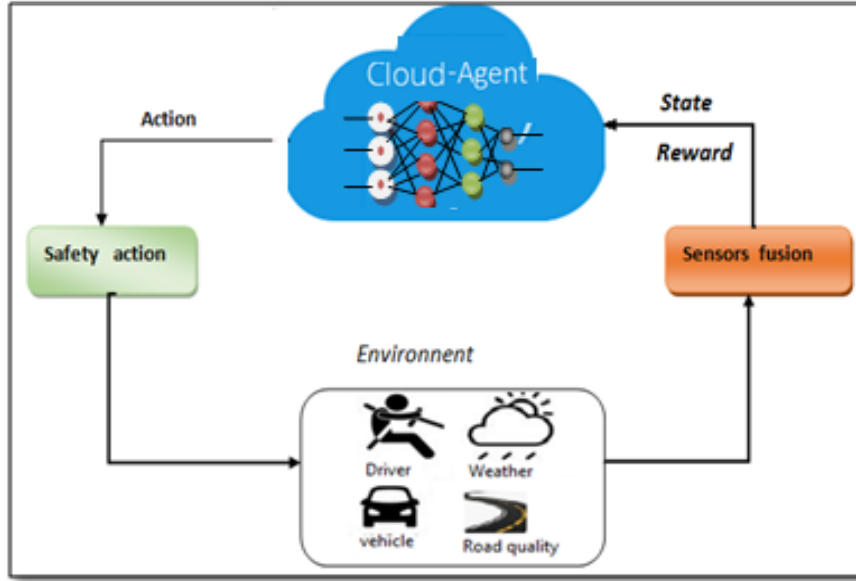


Figure 3.3: Proposed Cloud DRL based accident risk

There are four key elements in this DRL system: Cloud-Agent, observation/state, action, and reward scheme. We formulate traffic accident risk prediction problem as a reinforcement learning problem shown in Figure 3.3, where the Cloud-Agent interacts with the vehicle traffic environment in discrete time steps $(t_0, t_1, t_2 \dots t_N)$. The agent's objective is to reduce the number of accidents.

- **Cloud –Agent:** the agent observes the state of each vehicle, in its environment, defined by S_{t_i} at the beginning of time step t_i , then selects an action $A_{t_i} \in \mathbf{A}$ to perform. The use of a deep neural network (DNN) model in this case is very appropriate due to the large number of states. The DNN take input observations about traffic accidents and produces action decisions that should be taken as its output. The DNN architecture is a multilayer-network where the Cloud–Agent explores the information (available in the Cloud) about various accidents that have occurred previously and recommend the best actions that must be applied to avoid similar accidents from happening again.
- **Action:** refers to the decision recommended by a Cloud -Agent. It is a feedback on a state of risk accident, which is one of the following actions (*Stop, Deceleration, and No-Change of lane*) as an output to avoid this risk of accident.
- **State:** is an efficient representation of current road traffic condition. The representation variables contain multiple parameters reflecting the circumstances of a specific zone of an

urban transportation network to precisely describe the complexity of its dynamics. The agent learns through interacting with the environment episode by episode, where each episode ends with the prediction of accident risk for a vehicle, and the next episode starts.

- *Reward (penalty)*: the agent gets a reward Rt_i at the end of time step t_i as a result of the applied action At_i . The key requirement for a successful application of reinforcement learning is to design a reward function that frames the goal of an application and guides the learning towards a desirable behavior [100]. To reduce the traffic accident risk, it is reasonable to reward the agent at each time step for choosing an action that led to the avoidance of accidents [101].

In this system, the reward indicates the degree of risk encountered by the drivers. If there is no risk (negligible), the reward is set to '0', encouraging the maintenance of safe conditions through the action (No-Change). However, if a risk is detected and the appropriate action (such as stopping or decelerating) is not applied, a negative reward (penalty) is given to discourage unsafe behavior. Conversely, if the appropriate action (stopping or decelerating) is taken in the presence of a risk, the reward also remains '0', as the situation is managed correctly, but no additional positive reward is given to ensure that the main goal remains avoiding risk situations altogether. This approach ensures that actions failing to manage risks are penalized. Accordingly, we define the reward (or penalty) Rt_i for the agent choosing an action At_i at time step t_i as follows:

$$Rt_i = \begin{cases} 0 & \rightarrow \text{for : No Risk, action (No - Change /appropriate action)} \\ 1 * (N) \text{ or } 2 * (N) & \rightarrow \text{for : Risk, action (No - appropriate action)} \end{cases} \quad (3.1)$$

Where N is a negative integer, which represents the severity of an accident. The agent can perform one of these actions (*No-Change, Deceleration or Stop*) according to accident severity:

{(0) Negligible risk, (1) Serious risk and (2) Fatal risk}

The goal of reinforcement learning system is to achieve a safe road traffic system with no accident risk rate during the evaluation time (T). This is represented by the Total-Reward ($T_Rt_i \approx 0$):

$$T_Rt_i = \sum_{i=1}^T |Rt_i| \quad (3.2)$$

3.3.3 Deep Q-Network (DQN)

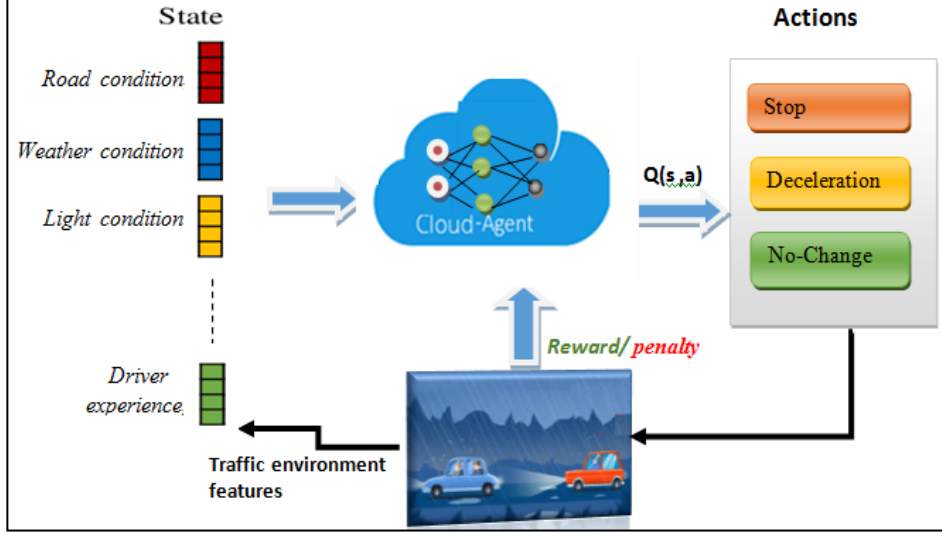


Figure 3.4: Cloud _DQL accident risk prediction

There are classical RL algorithms such as Q-learning, Policy Gradient (PG), Actor Critic, etc. Q-learning is one of the popular RL methods, which search for the optimal policy in an iterative fashion [102]. This algorithm is not suitable when we have a huge number of states and complex state transitions. In this work, a DQN algorithm that uses a DNN is utilized for predicting accident risks, with the aim of enhancing both the speed and accuracy of predictions. For each episode, the Cloud-Agent observes state St_i at the beginning of time step t_i , then makes action decision according to vehicle state, and receives a sequence of rewards (Rt_i) after time steps. If the cloud agent aims to reduce vehicle road accidents, it is sufficient to choose an action that maximizes the immediate reward Rt_i .

Since the agent aims to reduce the number of accidents in the long run, it needs to find an optimal policy noted (π^*) at every possible state-action pair. To find the optimal policy π^* , we need to find the optimal Q-value:

$$\begin{aligned} Q\pi^*(s, a) &= \max Q\pi(s, a) \\ &= Q^*(s, a) \end{aligned} \quad (3.3)$$

When the state space is continuous, it is impossible to find the optimal value of the state-action pair $Q^*(s, a)$ for all possible states. To deal with this problem, the DQN method was proposed, which approximates the state-action value function $Q(s, a)$ using the DNN, i.e, $Q(s,$

$a) \approx Q_{\theta}(s, a)$, where θ are parameters of the DNN that will be learned from raw traffic accident data.

We construct such a DNN network, where the network input is the observed traffic environment state S_{t_i} and the output is a vector of estimated Q-values $Q(s, a, \theta)$ for all actions $a \in A$ under observed state S_{t_i} . Figure 3.4 illustrates the Cloud_DQN module for traffic accidents prediction. Real-traffic accident data was collected in a buffer called a replay buffer to train our network. We build a neural network connected to several layers so that DNN approaches the Q-value. The agent learns parameters θ by training the DNN network to minimize the following mean squared error (MSE) as the loss function. MSE can be defined as the average squared difference between the target value and the predicted value [29], as shown in Equation (3.4):

$$MSE = \frac{1}{K} \sum_{i=1}^K (y_i - \hat{y}_i)^2 \quad (3.4)$$

Where y is the target value, $\hat{y}$ is the predicted value, and K is the number of training samples. Our target value should be the optimal Q value; the optimal Q value can be obtained by using the Bellman optimality Equation (3.5), where its Q value is just the sum of the reward (r) and the discounted maximum Q value of the next state-action pair [29]:

$$Q^*(s, a) = r + \gamma \max_{a'} Q^*(s', a') \quad (3.5)$$

Therefore, we can define our loss as the difference between the target value (the optimal Q value) and the predicted value (the Q value predicted by the DQN) and express the loss function L as (3.6) [29]:

$$L(\theta) = Q^*(s, a) - Q_{\theta}(s, a) \quad (3.6)$$

Substituting Equation (5) in Equation (6), we get Equation (3.7).

$$L(\theta) = r + \gamma \max_{a'} Q(s', a') - Q_{\theta}(s, a) \quad (3.7)$$

The Q value of the next state-action pair in the target is computed by the target network parameterized by θ' and the predicted Q value is computed by the main network parameterized

by θ . The loss function is represented by Equation (3.8).

$$L(\theta) = \frac{1}{K} \sum_{i=1}^K (r_i + \gamma \max_{a'} Q_{\theta'}(s', a') - Q_{\theta}(s, a))^2 \quad (3.8)$$

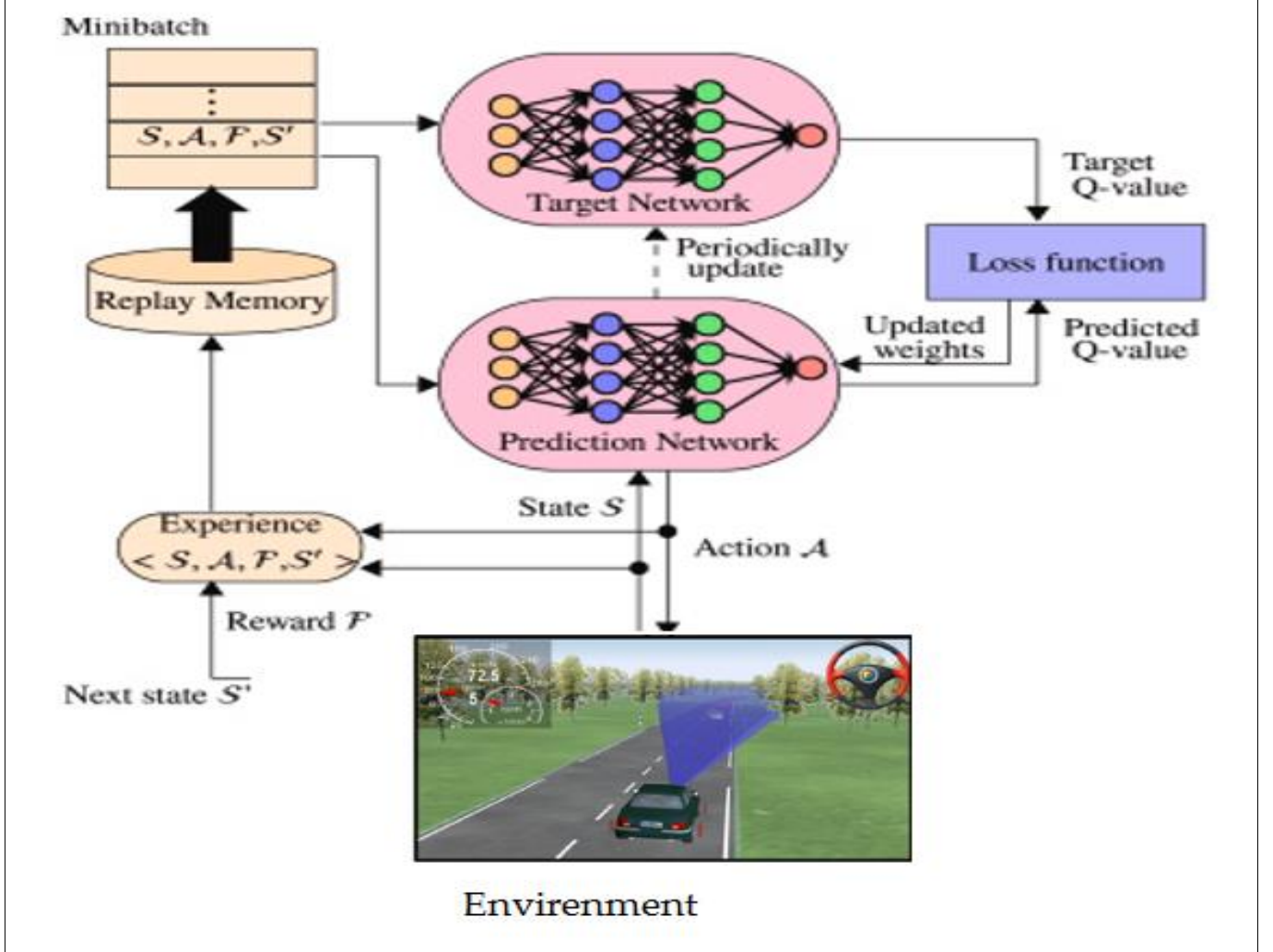


Figure 3.5: deep reinforcement learning process

The target network has the same architecture as the main network but different weights. Every N step, the weights from the main network are copied to the target network, where N is a hyperparameter that can be set by the user. Using both networks leads to more stability in the learning process and helps the algorithm to learn more effectively. To find the optimal parameter θ , we use gradient descent. We compute the gradient of our loss function $\nabla_{\theta} L(\theta)$ and update the network parameter θ as:

$$\theta = \theta - \alpha \nabla_{\theta} L(\theta) \quad (3.9)$$

The algorithm for training the Cloud_DQN is defined on next page.

Algorithm DQN algorithm

Initialize the main network with random weights θ ;

Initialize the target network with random weights θ' by copying the main network parameter θ ;

Initialize the replay memory capacity as D ;

For episodes = 1, M **do**

(1) Randomly select the initial road traffic state S ;

For $t = 1, T$ **do**

(2) Observe the state S and select random action a with probability ϵ otherwise select

$$a = \arg\max Q_{\theta}(s, a);$$

(3) Execute action a and move to the next state s' and obtain the reward r ;

(4) Store experience(s, a, r, s') in replay memory D ;

(5) Randomly sample a minibatch of K transitions from the replay memory D ;

(6) Compute the target value, that is, $y = r + \gamma \max_{a'} Q_{\theta'}(s', a')$;

(7) Compute the loss function value $L(\theta)$ with (3.8) ;

(8) Compute the gradients of the loss and update the main network parameter θ using (3.9) ;

(9) Every N step update $\theta' \leftarrow \theta$;

End For

End For

3.4 V.Edge_Deep Learning For Accident Risk Prediction

In the realm of road safety, the fusion of Vehicular Edge Computing (V.Edge) with deep learning holds significant promise, particularly in the context of accident risk prediction. This innovative approach involves a two-step process: the initial training of a deep learning model in the cloud, leveraging expansive storage and computational resources, followed by the deployment and inferencing of the trained model on V.Edge devices situated within vehicles. The decision to offload inferencing to the edge is motivated by the need for real-time responsiveness and the constraints of space and computing resources in vehicular environments.

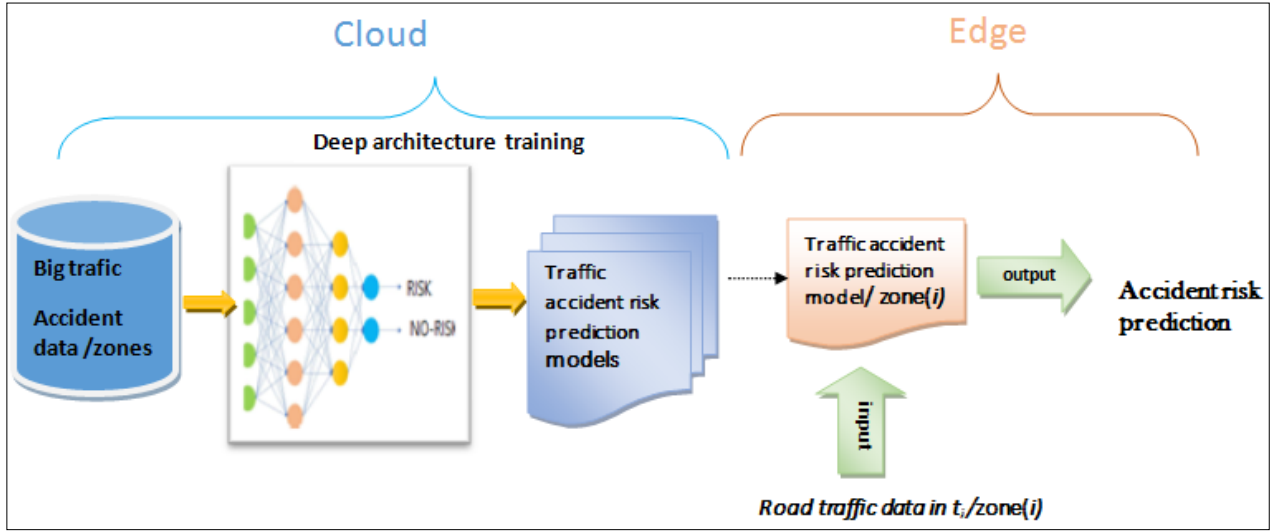


Figure 3.6: Regional V.Edge_DL traffic accident risk prediction

3.4.1 Data Collection at the Edge

Data collection at the edge, within the context of Intelligent Transportation Systems (ITS), represents a pivotal advancement in enhancing road safety. In the modern transportation landscape, vehicles have transformed into sophisticated hubs of technology, equipped with advanced sensors and communication capabilities. These sensors, including cameras, radars, and GPS devices, enable vehicles to gather diverse information crucial for preventing accidents. They assess road quality in real-time, monitor weather conditions, analyze driver behavior, and engage in Vehicle-to-Everything (V2X) communication for sharing critical data with the broader traffic infrastructure. This rich and varied information empowers vehicles to proactively contribute to road safety, serving as the foundation for subsequent stages, where the data is transmitted to the cloud for further analysis and model training. The collaboration between edge and cloud computing marks the beginning of an innovative approach to accident risk prediction, leveraging the strengths of both domains.

3.4.2 Cloud Processing and Training

Since DL often requires high-performance computing resources (GPUs, CPUs and storage devices) for model training and execution on massive data [103], the resources available in a vehicle may not fulfill this stringent requirement. Meanwhile, there is an imprecise trend: the more layers and parameters of a deep neural network, the more accurate the decision-making,

which would undoubtedly increase the training and running cost of deep learning models (DLMs) [104]. In this case, the cloud is the best solution to handle massy traffic accident data due to its scalability, availability of resources, and cost-effectiveness.

3.4.3 Model Deployment to the Edge (Vehicle)

In some situations, like sudden accidents, where fast response is the most important variable in the accident risk prediction problem, it is not always effective to rely on the cloud to send risk forecasting. This is because predictions sent from the cloud to the driver may be lost due to an internet disconnection or a high latency. The best solution is to build a deep learning model based on massy data for traffic accidents risk prediction in cloud platform, and then transferred it to the V.Edge to cover this cloud prediction absence. Once the training is complete, the trained deep learning model is deployed back to the edge (vehicle). The model is optimized for efficient inference and adapted to the limited resources (memory and computation) available on the edge device.

The deployed model on the edge (vehicle) performs real-time inference using the locally collected data. This inference involves predicting the likelihood of accidents based on the trained patterns and learned correlations. The edge (vehicle) can generate predictions about potential accidents in real-time. If a significant risk is detected, the system can trigger alerts or warnings to the driver, suggesting preventive actions. The V.Edge_DL can learn deep connections between traffic accidents and their spatial-temporal patterns. It aims to exploit historical traffic accident data to prevent future occurrences.

3.5 Collaboration V.Edge_DL / Cloud _DRL

In the pursuit of optimizing traffic safety, the collaboration between Cloud and V.Edge platforms emerges as a strategic approach. Acknowledging the inherent limitations of V.Edge in storing and processing vast amounts of Internet of Vehicle (IOV) data and generating Deep Learning (DL) models, a distributed learning system has been adopted. This collaborative effort aims to leverage the strengths of both platforms for an optimal prediction of traffic risks, ensuring a proactive and effective safety mechanism. The cooperation-communication between these platforms can have a vertical” *V2C*” (V.Edge- Cloud) or horizontal “*V2V*” (V.Edge– V.Edge) type.

3.5.1 V2C (V.Edge-Cloud)

Vertical Collaboration for Enhanced Predictions; within this collaborative framework, the vertical cooperation between V.Edge and Cloud, known as V2C (V.Edge-Cloud), plays a pivotal role. The V.Edge device, facing constraints in storage and processing capacity, relies on models generated at the cloud level to enhance the quality of traffic safety predictions. Figure 3.7 illustrates the flow of information, emphasizing the symbiotic relationship between the Cloud and V.Edge. This vertical collaboration ensures that the cloud's expansive resources contribute to refining models, ultimately leading to more accurate and efficient traffic risk predictions. In the realm of V.Edge and Cloud collaboration, two distinctive forms emerge, each contributing uniquely to the synergy between these platforms:

a) Communication (V.Edge - Cloud)

Within the Internet of Vehicles (IOV) technology, sensors embedded in vehicles serve as data custodians, capturing intricate details about the road, vehicle dynamics, and driver behavior. This wealth of information traverses through the digital highway, conveyed to the cloud through the medium of V2C (Vehicle to Cloud) communication. In this context, the data becomes a potent tool for predictive analytics, empowering the Cloud_DRL entity to discern potential accident risks. In the event of an impending threat, the cloud swiftly dispatches alert messages to the driver, fostering timely responsiveness and accident prevention.

b) Cooperation (Cloud - V.Edge)

The collaboration between Cloud and V.Edge unfolds in a second form, where the cloud assumes the role of a mentor. High-end computing resources in the cloud become the training ground for deep learning models, refining their intelligence with vast datasets. Once honed, these models descend from the cloud and take residence in the V.Edge device. This orchestrated intelligence empowers V.Edge with the capability for real-time inference. By distributing the cognitive load, this collaborative approach ensures swift decision-making and processing, mitigating the risk of delayed responses during crucial moments. This dual perspective encapsulates the versatility of (V.Edge - Cloud) collaboration, showcasing its adaptability in both communication-driven predictive insights and intelligence-enhancing model deployment scenarios.

3.5.2 V2V (V.Edge-V.Edge)

In addition to vertical collaboration, the distributed learning system introduces horizontal cooperation denoted as V2V (V.Edge-V.Edge). This type of collaboration emphasizes the communication and information exchange between multiple V.Edge platforms. By fostering a network of interconnected edge devices, this approach facilitates seamless connectivity and information sharing among vehicles. The horizontal collaboration empowers V.Edge devices to collectively contribute to traffic safety predictions, creating a dynamic and responsive ecosystem at the vehicular edge. Embedded a deep learning model into different vehicles enables effective collaboration and communication among them for accurate prediction of road accidents.

a) Communication (V.Edge - V.Edge)

In this scenario, vehicles transcend their individual entities and engage in collaborative communication through Vehicle-to-Vehicle (V2V) interactions. Enabled by wireless communication protocols, vehicles share crucial data such as speed, direction, and location with their counterparts on the road. This real-time exchange of information establishes a dynamic network where vehicles become aware of each other's status, fostering a collective effort to prevent potential collisions and address hazardous behaviors. The interconnected V.Edge communication ensures a synchronized response to the evolving conditions on the road.

b) Cooperation (V.Edge – V.Edge)

Beyond communication, V.Edge collaboration takes the form of cooperation between vehicles in scenarios where cloud connectivity faces challenges. In this collaborative model, a vehicle possesses the capability to refer to another vehicle for importing a deep learning model specific to its current zone. This is particularly valuable in situations of internet connectivity issues with the cloud. By sharing and leveraging local knowledge, vehicles collaboratively contribute to maintaining a robust and responsive system for accident risk prediction. This decentralized approach ensures continuous functionality even in the absence of direct cloud interaction.

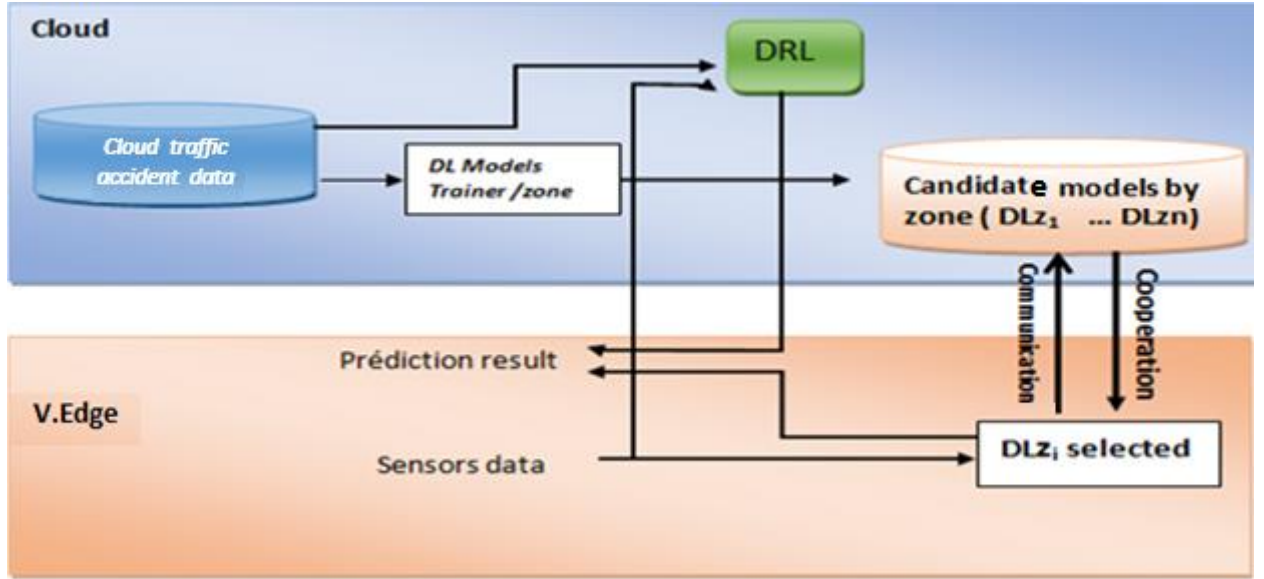


Figure 3.7: Collaboration / communication (V. Edge -cloud)

3.6 Conclusion

In conclusion, our chapter has delineated an innovative approach that harnesses the power of collaboration between cloud machine learning techniques and vehicular edge computing (V.Edge.C) to bolster driver safety practices. Our primary objective has been to anticipate accident risks proactively and swiftly alert drivers to avert potential collisions. By synergizing deep learning and deep reinforcement learning algorithms with the capabilities of cloud computing and vehicular edge computing, we've endeavored to address the intricate challenges within transportation systems more effectively. While these methodologies necessitate significant resources, the collaborative integration of cloud and edge platforms offers promising solutions to overcome limitations associated with standalone systems. Our proposed advanced system, driven by Deep Reinforcement Learning (DRL), is poised to accurately forecast traffic accident risks. By conducting training in the cloud and deploying models to vehicular edge devices for real-time inference, we aim to significantly enhance crash prediction and prevention.

In this chapter, we have outlined the design of our idea and explored various scenarios of collaboration between the two platforms. In the subsequent chapter, we will advance to the implementation phase of this approach.

Chapter (4): Experiments and results

4.1 Introduction

The study is structured around a series of phases designed to guide the process from the initial selection of data to the ultimate goal of prediction. These phases emphasize a hands-on approach, particularly suitable for issues pertaining to detection and prediction rather than traditional systems engineering. The workflow consists of six key stages: data selection, data pre-processing, exploratory data analysis, transformation, modeling, and evaluation. The figure below visually represents the systematic flow through these essential phases, outlining a comprehensive journey from the deliberate choice of data to the final prediction stage.

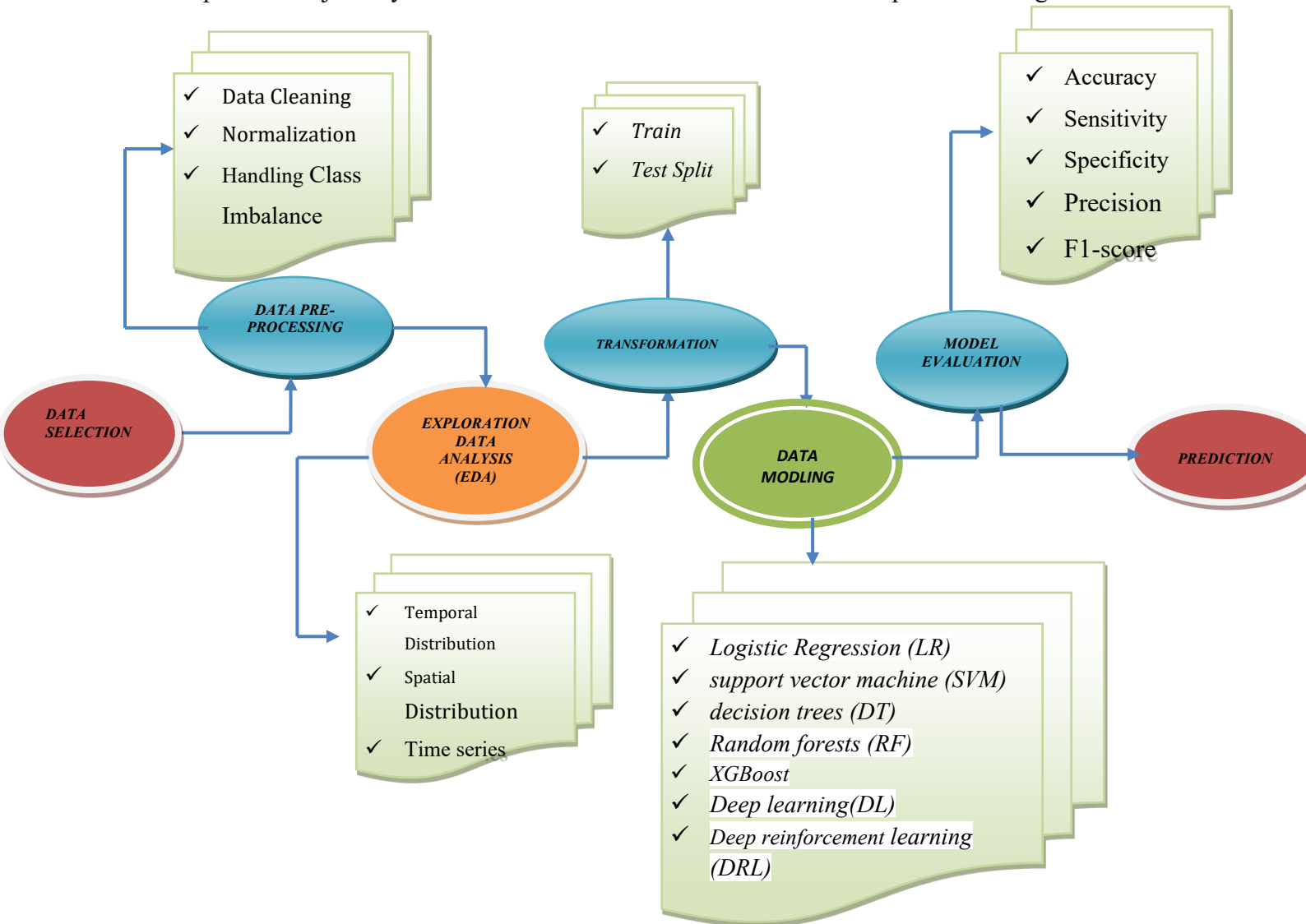


Figure 4.1: Flow-Based Methodology for Data-Driven Prediction

4.2 Data Selection

The datasets utilized in this study are sourced from the publicly available Open Data website of the UK government (<http://data.gov.uk/>), specifically published by the Department of Transport. This dataset encompasses accidents information spanning from 2005 to 2015. Three CSV-formatted files constitute this dataset (Accidents, Casualties, and Vehicles) in (.csv) format. Accidents is the main file, and through the Accident Index column, it has links to Casualties and Vehicles. The Accidents file comprises 1780653 rows and 31 columns. The Casualties file contains 2216720 rows and 14 columns, and the Vehicles file has 3004425 rows and 21 columns. The Big Data presenting a significant challenge when loading it into CPU memory and addressing non-numeric entries during the cleaning process. With over three million entries, the dataset proves substantial enough for robust machine learning model training.

The term "accident severity" in this dataset refers to the level of seriousness of the consequences of a traffic accident. Each recorded accident is categorized into one of three predetermined groups based on its outcomes: 1) *Fatal*, 2) *Serious*, or 3) *Slight*. Fatal accidents involve the death of individuals involved, while serious accidents result in severe injuries but not death. Slight accidents are those causing minor injuries or no injuries at all.

4.3 Programming Language, Libraries And Used Environment

The programming language chosen for our methods presented previously is Python. Python is a powerful and easy-to-learn programming language known for its high-level data structures and simple yet efficient approach to object-oriented programming. The decision to use Python offers several advantages, including:

- **Cost-effectiveness:** Python is completely free, making it an economical choice for development.
- **Ease of Learning and Use:** Python is renowned for its simplicity, making it easy for developers to learn, read, understand, and write code. This characteristic promotes a smooth learning curve for beginners and enhances productivity for experienced programmers.
- **Object-Oriented Flexibility:** Python supports object-oriented programming but does not impose it, providing flexibility in coding styles based on project requirements.

- **Cross-Platform Compatibility:** Python is compatible with all major operating systems and computer platforms, ensuring the portability of code across diverse environments.
- **Professional Quality Software:** Python is widely used for developing professional-grade software, thanks to its extensive libraries, frameworks, and community support.

Python offers several libraries (packages) for data processing, creating machine learning models, matrix calculations, analysis, and data visualization, as illustrated in the table below:

Table4.1: Python Libraries Utilized.

Library	Main Role
Numpy (np)	Efficient manipulation of arrays and numerical computations.
Pandas (pd)	Data manipulation and analysis using structured data, like DataFrames.
Matplotlib.pyplot (plt)	Creating plots and data visualizations.
Seaborn (sns)	Advanced data visualization based on Matplotlib, with a focus on statistical graphics.
Sklearn	Comprehensive machine learning tools, including preprocessing, models, and metrics.
Tensorflow (tf)	A deep learning framework for building and training neural networks.
Keras	A high-level API integrated with TensorFlow for easily building deep learning models.
Gym	Toolkit for developing and comparing reinforcement learning (RL) algorithms.
Baselines	Implementations of RL algorithms (DQN) and tools for benchmarking and logging.
Random	Generating random numbers, making random selections, and shuffling data.

- **Google Colab environment**

Google Colab, also known as "the Colaboratory," serves as a complimentary cloud service by Google aimed at fostering research in Machine Learning and Artificial Intelligence. It addresses the common hurdle in these fields, which is the need for substantial computational resources. The platform offers free GPU support, enabling users to enhance their Python programming skills while developing deep learning applications utilizing popular libraries such as Keras, TensorFlow, PyTorch, and OpenCV. Additionally, it seamlessly integrates with Google Drive for project storage. What sets Colab apart from other free cloud services is its

provision of GPU resources without any cost. Our work centers on a deep convolutional network, demanding significant training data points, which we managed within the Google Colab notebook environment.

4.4 Data Pre-Processing

Several preprocessing steps were meticulously executed on the dataset to ensure a robust and unbiased analysis.

4.4.1 Data Loading

The data loading process involved importing the dataset into a Pandas Data Frame utilizing the `'pd.read_csv()'` function. This function is a part of the Pandas library in Python, specifically designed to read comma-separated values (CSV) files. By passing the file path as an argument to this function, the data was efficiently loaded into memory, ready for further analysis and manipulation.

4.4.2 Data Cleaning

Data cleaning involves identifying and rectifying errors, eliminating duplicates, handling missing values, and correcting inconsistencies. Techniques such as `'drop_duplicates ()'` and `'dropna()'` were employed to ensure data completeness and quality. To handle missing values, both 'NaN' (Not a Number) and '-1' were considered as indicators of missing data.

4.4.3 Data Scaling

Data Scaling, also known as feature scaling or normalization, played a crucial role in our dataset preprocessing, particularly due to the varied scales of attributes present. To ensure uniformity across all features, we opted for the Min-Max scaling technique. This approach was instrumental in standardizing the range of feature values, constraining them within the 0 to 1 interval. By doing so, we not only achieved consistency but also set the stage for enhanced convergence and effectiveness of subsequent classification algorithms. This transformation effectively harmonized the disparate scales of attributes, laying a solid foundation for accurate and reliable model training.

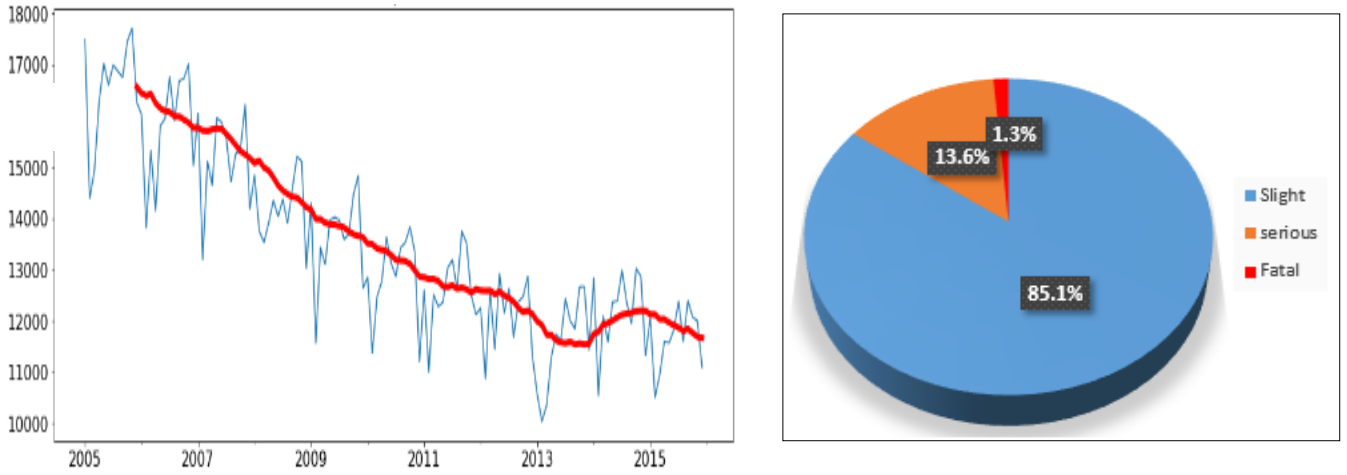
4.4.4 Handling Class Imbalance

Upon initial examination of the dataset, it became evident that there was a significant class imbalance within the target attribute, *Accident_Severity*. The majority of accidents, accounting for 85.1%, were categorized as '*Slight*.' In contrast, *serious* accidents represented 13.6%, while *fatal* accidents constituted a mere 1.3% of the data. Such a skewed distribution poses a challenge as it can introduce bias during model training and diminish the predictive capacity, particularly for the underrepresented classes. To mitigate this imbalance, two techniques were employed:

- a. *Under-sampling*: In the first step, under-sampling was implemented using the `RandomUnderSampler` method from the `imblearn` library. This method randomly removes instances from the over-represented classes, thereby achieving a more balanced class distribution.
- b. *Over-sampling*: Following under-sampling, the Synthetic Minority Over-sampling Technique (SMOTE) from the `imblearn` library was applied to further balance the dataset. SMOTE operates by generating synthetic examples in the feature space, effectively reinforcing the minority class. By augmenting the dataset with synthetically generated data points, SMOTE ensures a more equitable class distribution, thereby enhancing the model's ability to generalize effectively across all classes.

4.5 Exploratory Data Analysis (EDA)

In this analysis process, an investigation of the dataset was conducted through the utilization of exploratory data analysis (EDA) techniques. This involved the utilization of various plots, graphs to visually illustrate the data, revealing underlying patterns and correlations among the variables present in the accident data. Through the EDA process, a more profound understanding of the factors impacting accidents was achieved, encompassing aspects such as weather conditions, types of roads, and the specific times of day when accidents occur.



(a): Time series plot of RTAs; (b): Accident Severity Distribution

Figure 4.2: Accident Distribution

Figure 4.2 (a) depicts a notable decline in road accidents across the UK spanning from 2005 to 2015. Over this timeframe, there was a significant decrease in the total number of accidents. This decline underscores the efficacy of road safety initiatives and signals substantial progress in curbing accident rates throughout the specified period. Such a trend reflects positively on the effectiveness of implemented measures aimed at enhancing road safety and mitigating the occurrence of accidents. In Figure 4.2(b) a dataset analysis represented reveals that most accidents (85.1%) were classified as '*Slight*.' In contrast, *serious* accidents comprised (13.6%) of the total, while *fatal* accidents were the least frequent, with a count of (1.3%)

In the following section, we aim to visually represent the distribution of accidents across various factors, including human factors such as age and gender of drivers, environmental conditions such as light conditions, weather conditions, and road surface conditions, and finally, distribution across time by occurrences per month, day, and hour. To improve the clarity and effectiveness of our visualizations, we will create insightful graphs that provide a comprehensive understanding of the patterns and trends within each category. By analyzing these visual representations, stakeholders can gain valuable insights into the contributing factors and circumstances surrounding accidents, which can inform targeted interventions and preventive measures to enhance road safety.

4.5.1 Accident Distribution Across human factors

By studying the human factors and impact, we can design new rules to the drivers depending on their age, sex, and many more individual or combined characteristics.

An interesting finding from our analysis of this data is the age_of_driver. The analysis shows that the age band of the driver plays an important role in causing traffic accidents. Figure 4.3(a) shows the distribution of age driver, it can be observed that the majority of drivers lie in the age band of [26-35, 36-45, 46-55]. Figure 4.3(b) shows the distribution for fatal, serious and slight severity accident over the sex of the driver. Our analysis shows that male drivers are higher than female drivers.

Previous studies in the field of traffic flow and incidents have modeled and simulated human factors and their impacts using mathematical modeling techniques. Although mathematical models are powerful, they often struggle to capture the complex and diverse nature of human behavior, limiting their ability to generalize effectively. The unpredictable interactions among various human elements on the road give rise to a wide range of scenarios that traditional models cannot fully account for.

In this context, leveraging big data mining techniques emerges as a promising alternative for studying human factors and interactions. By extracting insights from large volumes of

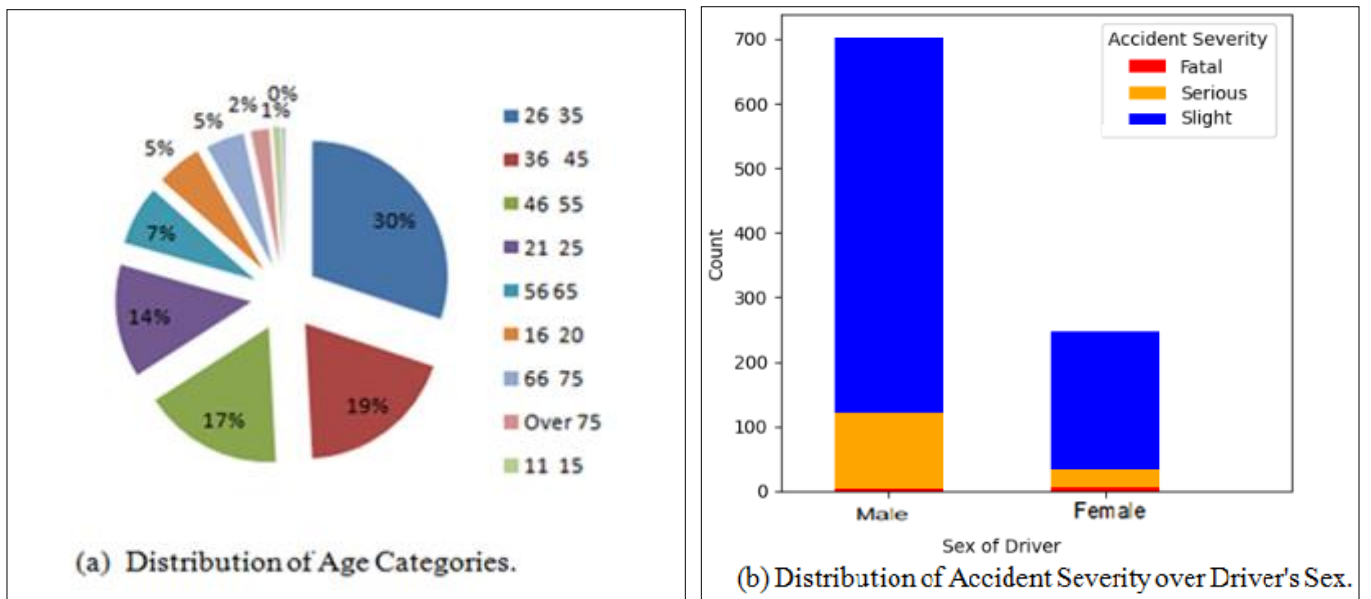


Figure 4.3: Accident Distribution Across human factors.

historical data, data mining provides a practical approach for forecasting and anticipating future traffic-related scenarios.

4.5.2 Accident Distribution Across Environmental Conditions

All of environmental conditions considered by this study showed a similar distribution (Figure 4.4). Contrary to common belief, our analysis reveals that a significant 80% of accidents occur under favorable weather conditions with no winds. Intriguingly, the lowest incidence of accidents is observed in snowy conditions and high winds. It's noteworthy that this data pertains to the UK, where adverse weather, such as fog, typically prevails throughout the year.

The same issue with the light condition, Contrary to the common belief that accidents predominantly occur in low-light or nighttime conditions, our analysis reveals a surprising trend. A significant 73% of the total accidents in our dataset occurred during daylight hours. Moreover, of the accidents that took place in daylight. This challenges preconceived notions about accident patterns and emphasizes the importance of addressing factors contributing to daytime incidents. The majority of accidents occurred under dry road surface conditions, comprising 69% of all incidents, followed by wet or damp conditions at 28%. Notably, a significant proportion of accidents also transpired in urban areas. This trend can be attributed to the dense traffic and prevalent congestion characterizing urban roadways, which elevate the risk of collisions.

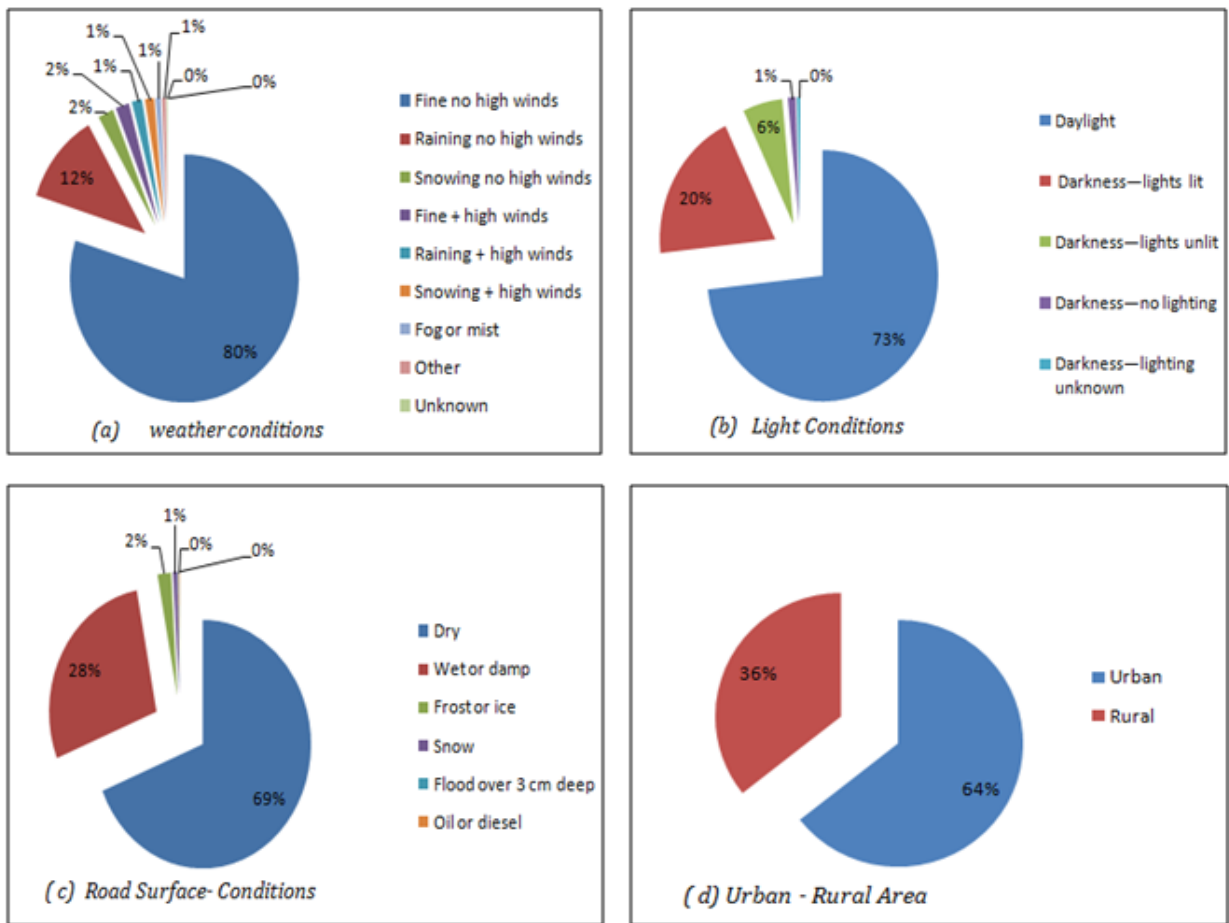
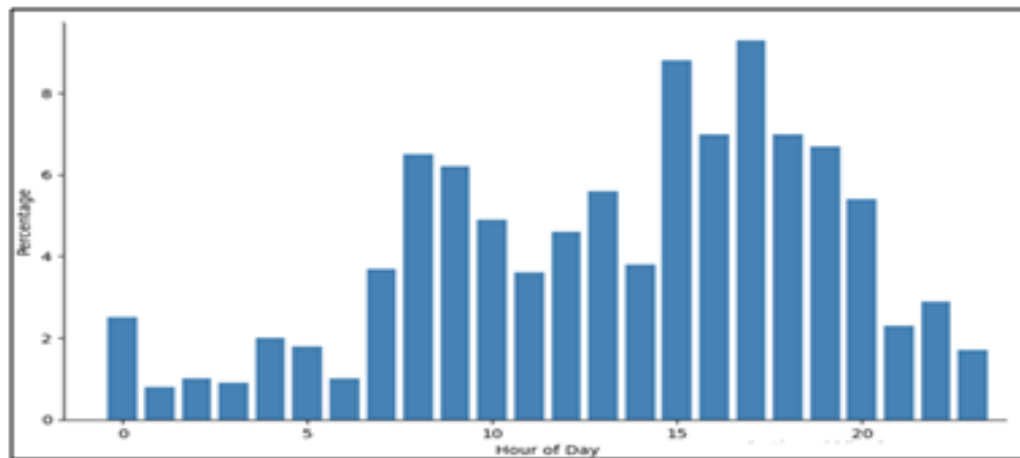


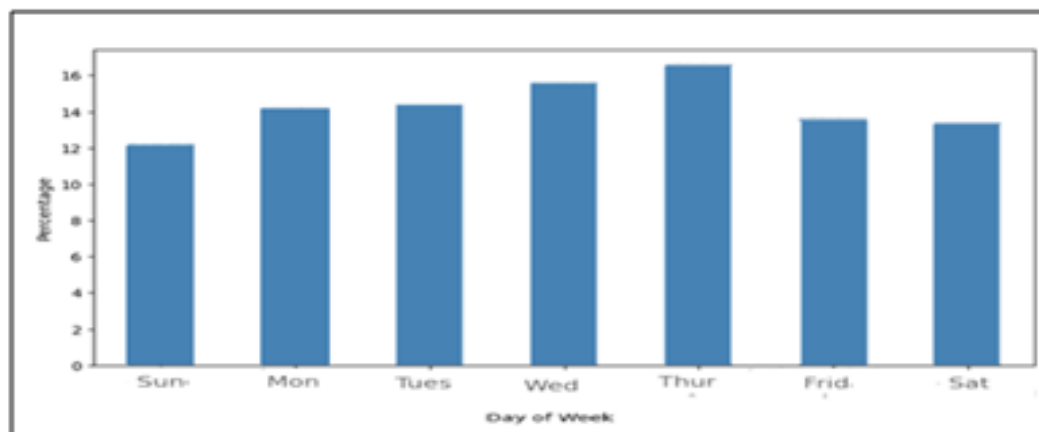
Figure 4. 4: Accident Distribution Across Environmental Conditions

4.5.3 Accident Distribution Across Time

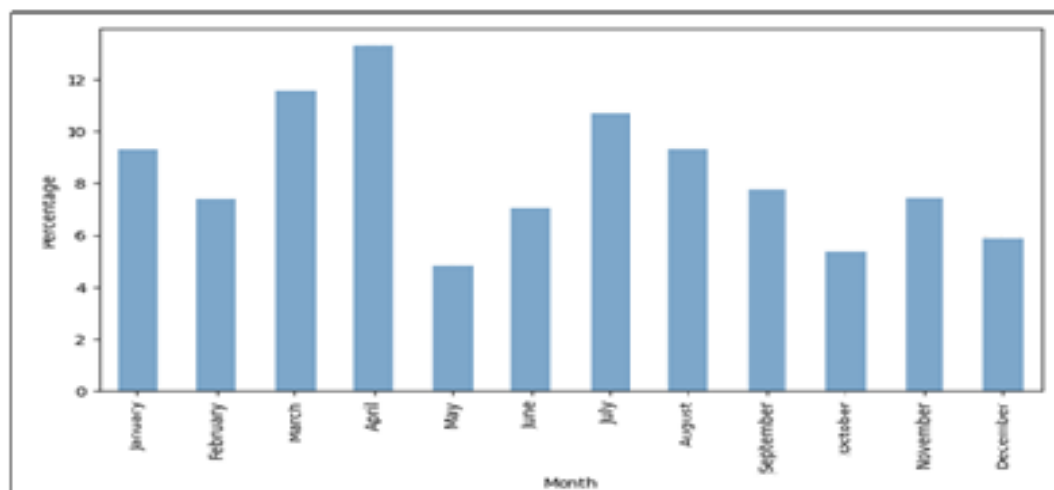
Figure 4.5 depicts crash frequency by month, by day of the week, and by hour of the day, revealing intriguing accident data concerning time patterns. In Figure 4.5(a), the accident rate is observed to increase at 8 a.m. and between 3 p.m. and 5 p.m., particularly during rush hours when traffic is at its peak, such as people commuting to and from work. In Figure 4.5(b), weekdays tend to have more incidents compared to weekends, possibly due to higher travel volume on workdays, which raises the risk of traffic accidents. Figure 4.5(c) summarizes the annual traffic accidents. The chart illustrates the accident rate by year. It appears that accidents fluctuate significantly throughout the year, peaking in the middle of the year. March and April have consistently had the highest number of accidents over the years. In contrast, May has the fewest accidents, with a decrease towards the end of the year.



(a) Distribution of Accidents by hour of day.



(b) Distribution of Accidents by day of week.



(c) Distribution of Accidents by Month.

Figure 4. 5: Accident Distribution Across Time

4.6 Feature Importance

In this section, an analysis is conducted on the feature space to determine feature importance and feature correlation within the generated dataset. Initially, a correlation measure is employed to explore the relationship between each feature and accidents. Subsequently, a machine learning feature selection method, such as the Scikit-learn Random Forest library, is utilized to identify the most relevant and correlated attributes that influence the learning process. These findings are illustrated in Figure 4.6.

Table 4.2 presents the (20) important features description of this dataset, which will form the input vector of our ML models.

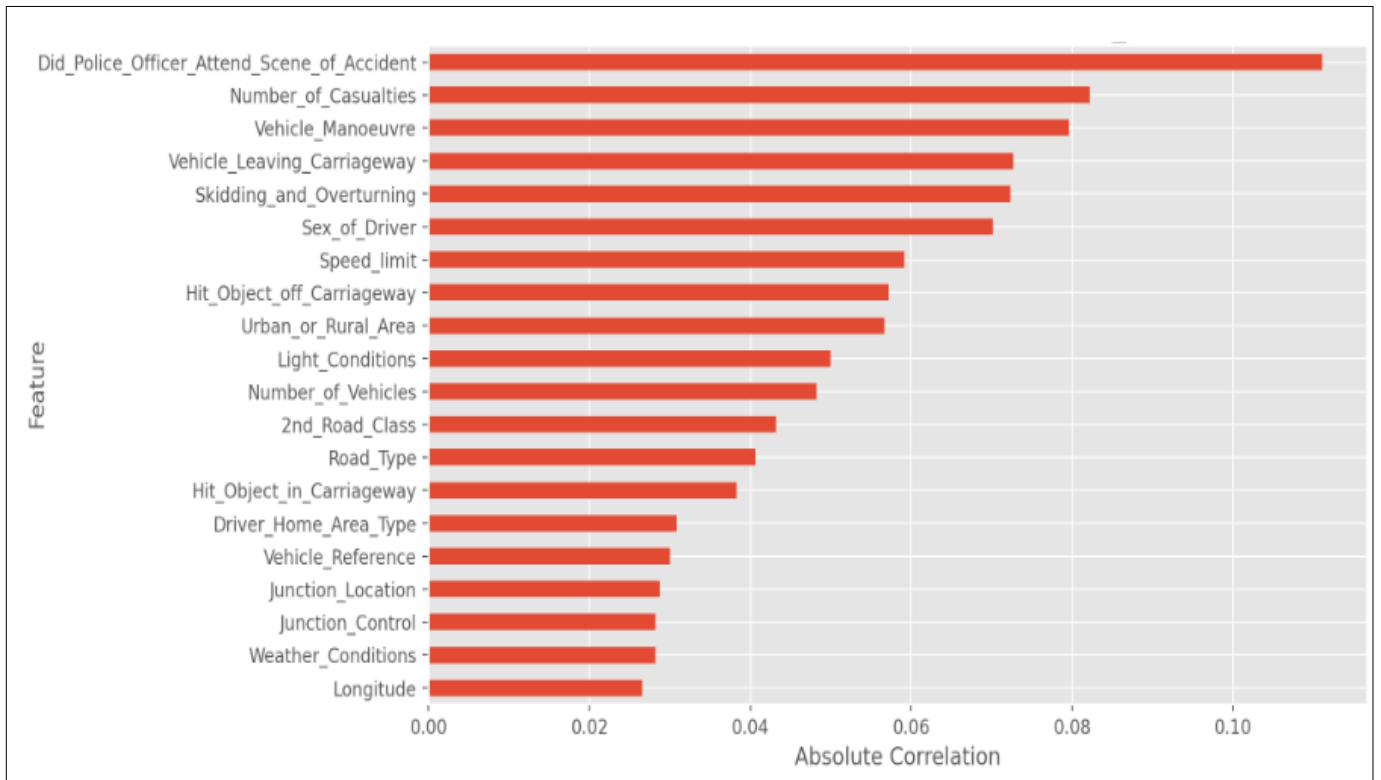


Figure 4. 6: Attribute importance scores.

Table 4. 2: Input factors

<i>Feature</i>	<i>Description</i>
<i>Did Police Officer Attend Scene of Accident</i>	<i>This feature indicates whether a police officer attended the scene of the accident, which may correlate with the severity or type of accident.</i>
<i>Number_of_Casualties</i>	<i>The number of casualties involved in the accident may be correlated with the occurrence of the accident, as more casualties may indicate a more severe accident.</i>
<i>Vehicle Manoeuvre</i>	<i>The type of vehicle manoeuvre performed at the time of the accident may be correlated with the likelihood of an accident occurring, depending on the manoeuvre executed.</i>
<i>Vehicle Leaving Carriageway</i>	<i>Vehicles leaving the carriageway may be more likely to be involved in accidents, correlating with the occurrence of accidents.</i>
<i>Skidding_and_Overturning</i>	<i>Skidding or overturning vehicles may indicate loss of control, potentially correlating with accident occurrence.</i>
<i>Sex of Driver</i>	<i>The gender of the driver may correlate with accident occurrence due to differences in driving behaviour or risk-taking tendencies.</i>
<i>Speed_limit</i>	<i>Higher speed limits may be correlated with a higher likelihood of accidents occurring due to increased risk of loss of control or collision severity.</i>
<i>Hit_Object_off_Carriageway</i>	<i>Objects struck off the carriageway may indicate erratic vehicle behaviour, correlating with accident occurrence.</i>

<i>Urban_or_Rural_Area</i>	<i>The type of area where the accident occurs (urban or rural) may correlate with accident occurrence due to differences in traffic density and road conditions.</i>
<i>Light Conditions</i>	<i>Poor lighting may influence the severity of accidents, especially at night or in tunnels, where reduced visibility can lead to more serious outcomes.</i>
<i>Number of Vehicles</i>	<i>The number of vehicles involved in an accident may be correlated with the occurrence of the accident, potentially indicating multi-vehicle collisions.</i>
<i>2nd_Road_Class</i>	<i>The classification of the second road involved in the accident may correlate with the likelihood of accidents occurring, depending on road characteristics.</i>
<i>Road Type</i>	<i>The type of road where the accident occurs may correlate with accident occurrence, as different road types have varying risk factors.</i>
<i>Hit_Object_in_Carriageway</i>	<i>Objects struck within the carriageway may indicate collision patterns, correlating with accident occurrence.</i>
<i>Driver_Home_Area_Type</i>	<i>The type of area where the driver resides may correlate with accident occurrence due to differences in driving habits or road familiarity.</i>
<i>Vehicle Reference</i>	<i>Vehicle reference information may correlate with the occurrence of the accident, depending on vehicle characteristics or condition.</i>
<i>Junction Location</i>	<i>Accidents occurring at junctions may be more common and correlate with accident occurrence due to the complexity of traffic movements.</i>

<i>Junction Control</i>	<i>The type of control at junctions may correlate with accident occurrence, as different control types affect traffic flow and collision risk.</i>
<i>Weather Conditions</i>	<i>Adverse weather conditions may increase the likelihood of accidents occurring, correlating with accident occurrence.</i>
<i>Longitude</i>	<i>The geographical location (longitude) of the accident may correlate with accident occurrence due to varying road and traffic conditions.</i>

4.7 Evaluation Metrics

It is necessary to identify and estimate the efficiency and effectiveness of Cloud_DRL and V.Edge_DL in predicting traffic accidents with the dataset. Our models are validated in terms of:

- Learning curves (Accuracy and Loss) for both of *Cloud_DRL* and *V.Edge_DL*.
- Comparison with other algorithms in terms of evaluation metrics.
- Efficiency and effectiveness in reducing the risk of road accidents with or without cooperation.

The calculation of evaluation metrics is mainly based on (N x N) confusion matrix (shown in Figure 4.7) that is used to display the performance of the algorithm, where N is the number of target classes. This matrix compares the actual target values with those predicted by the machine learning model. To comprehensively measure the performance of the proposed models, accuracy, sensitivity, F1 score, and other indicators are used. The concept and formula for calculating each of these indicators are shown in Table 4.3 Where TP denotes true positive, FP denotes false positive, TN denotes true negative, and FN denotes false negative.

		<i>Predicted values</i>	
		<i>Positive</i>	<i>Negative</i>
<i>Real values</i>	<i>Positive</i>	TP	FN
	<i>Negative</i>	FP	TN

Figure 4. 7: Confusion matrix.

Table 4.3: Main metrics for classification

Metric	Formula	Interpretation
<i>Accuracy (Acc)</i>	$\frac{(TP + TN)}{TP + TN + FP + FN} * 100\%$	Gives the proportion of the total number of predictions that were correct
<i>Precision (Pre)</i>	$\frac{TP}{(TP + FP)} * 100\%$	How accurate the positive predictions are
<i>Recall (Sensitivity)</i>	$\frac{TP}{(TP + FN)} * 100\%$	Gives information about the True Positives that are correctly classified during the test
<i>Specificity</i>	$\frac{TN}{(TN + FP)} * 100\%$	Gives information about of True Negatives that are correctly classified during the test
<i>F1-score</i>	$\frac{2 * TP}{(2 * TP + FN + FP)} * 100\%$	Hybrid metric useful for unbalanced classes

4.8 Results And Discussion

4.8.1 Data Splitting

Training and testing on the same dataset are not good approaches. If we train and test on the same dataset, it might give high performance but this might lead to overfitting and the model might fail to generalize well. To avoid such problems, we split the datasets into training and

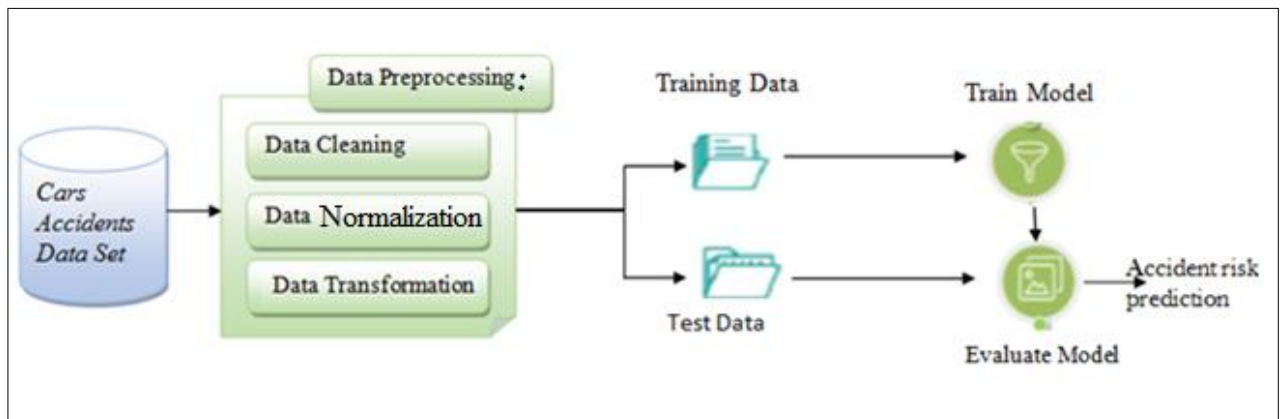


Figure 4.8: Experiment procedure.

test data. During the construction of our machine learning models, the dataset was divided into training dataset (80%) and test dataset (20%).

4.8.2 Cloud_DRL Vs V.Edge_DI learning curves

Epsilon-Greedy Strategy: as illustrated in the transformations of deep reinforcement learning curves, the exploration-exploitation strategy requires a delicate balance between discovering new actions and exploiting known ones to maximize cumulative rewards. At its core, this strategy revolves around a parameter ‘epsilon (ϵ)’, which determines the probability of choosing a random action versus selecting the action with the highest estimated value based on previous experience. During exploration (when ‘epsilon’ dictates random action selection), the agent gathers information about the environment, ensuring it doesn't prematurely settle on suboptimal actions. In contrast, exploitation (when ‘epsilon’ favors choosing the best-known action) utilizes accumulated knowledge to maximize immediate rewards. This dynamic ensures that over time, the agent refines its decision-making by gradually shifting focus from exploration to exploitation as it learns more about the environment and the rewards associated with different actions. By striking a balance between these two approaches, the agent can effectively navigate uncertain environments while striving for optimal long-term performance. In Deep Q-Learning (DQL), the choice of hyperparameters is critical for guiding the learning process and ensuring the model's effectiveness, which are defined as follows:

Table4.4: DRL hyperparameters.

Parameter	Description/Value
<i>State_size</i>	Input layer size: number of features
<i>Action_size</i>	Output layer size: 3 (possible actions: stop, decelerate, No-change)
<i>Hidden Layers</i>	3 layers
<i>Activation functions</i>	ReLU for hidden layers; Linear for output layer
<i>Optimiser</i>	Adam optimizer
<i>Memory_size</i>	Replay memory capacity: 512
<i>Gamma</i>	Discount rate for future rewards: 0.99.

<i>Epsilon</i>	Initial exploration rate: 1.0.
<i>Epsilon_min</i>	Minimum exploration rate: 0.005.
<i>Epsilon_decay</i>	Decay factor for exploration rate: 0.995.

- To build the best traffic accidents predictive framework, we used a Convolution Neural Network (CNN), which is one of the best classification algorithms based on artificial neural networks. CNN is designed to learn automatically and adaptively using multiple building blocks such as convolution layers, pooling layers, and fully connected layers. The set of deep learning model hyperparameters is described in the following table.

Table 4.5: DL hyperparameters

Parameter	Value
<i>Input Layer</i>	number of features.
<i>Output Layer</i>	3 neurons (1 per Casualty Severity class)
<i>Conv1D Layers</i>	3 layers: 64 filters
<i>Kernel Size</i>	4 for both Conv1D layers
<i>Pooling</i>	MaxPooling1D with pool size 2 after each Conv1D layer
<i>Dense Units</i>	64 $\rightarrow$ 32 $\rightarrow$ 3
<i>Activation Functions</i>	ReLU for hidden layers, Softmax for output
<i>Dropout</i>	0.5 after first Dense, 0.3 after second Dense
<i>Optimizer</i>	Adam (learning rate = 0.001)
<i>Batch Size</i>	64
<i>Epochs</i>	20 (with early stopping, patience=5)
<i>Loss Function</i>	Categorical Crossentropy

The results obtained in terms of accuracy and loss for both models are displayed in Figure 4.9 and Figure 4.10.

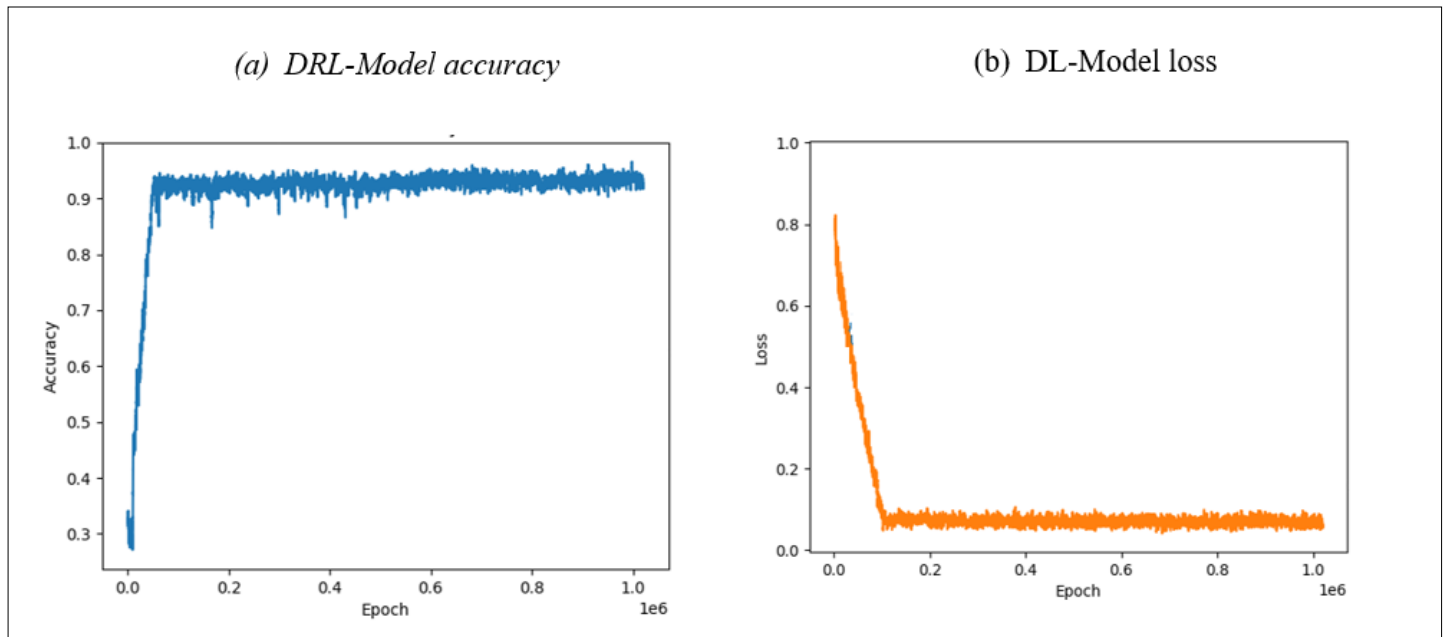


Figure 4.9: Learning curves for DRL Model

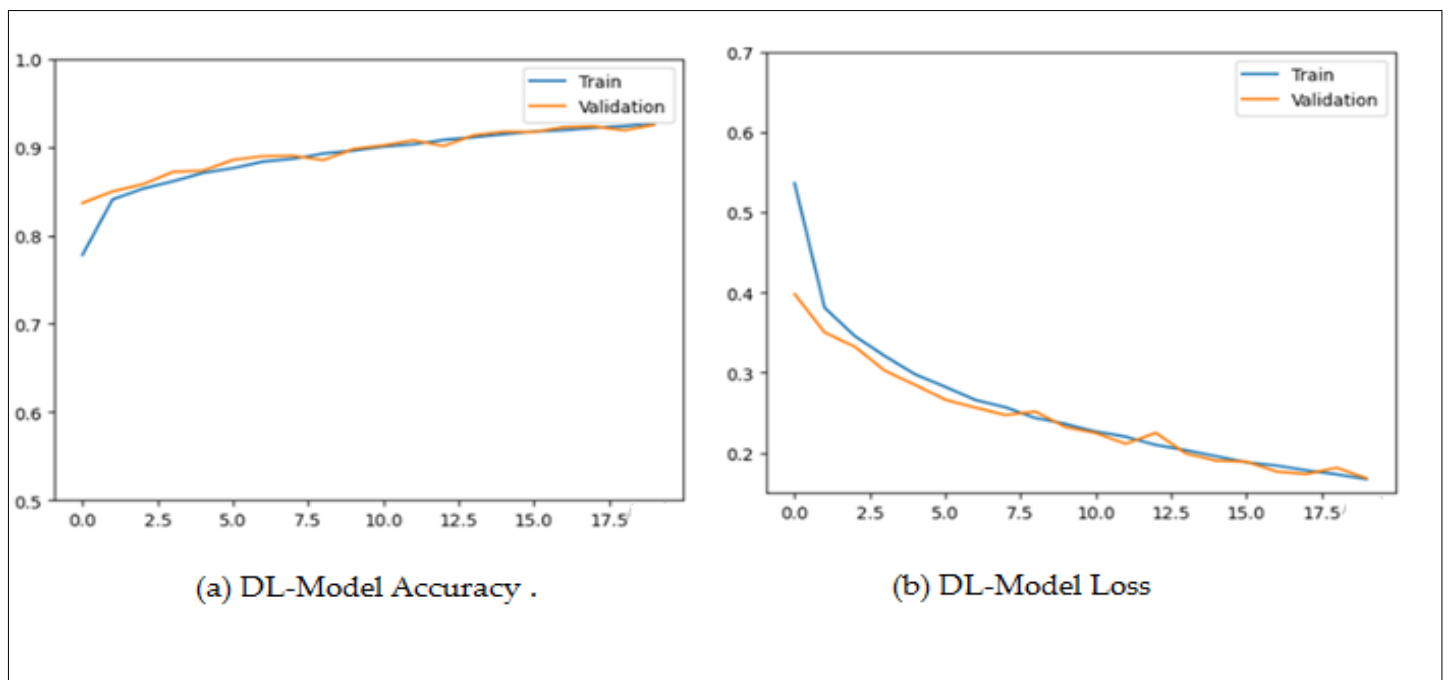


Figure 4.10: Learning curves for DL Model

4.8.3 Performance comparison

The proposed models Cloud_DRL and V.Edge_DL are compared to other well know algorithms [104], such as Logistic Regression (*LR*), support vector machine (*SVM*), decision trees (*DT*), Random forests (*RF*), and *XGBoost* in terms of Accuracy, Sensitivity, Specificity, Precision, and F1-score measures. The experiment results are summarized in Table 4.6. Figure 4.11 visualized the results in Table 4.6.

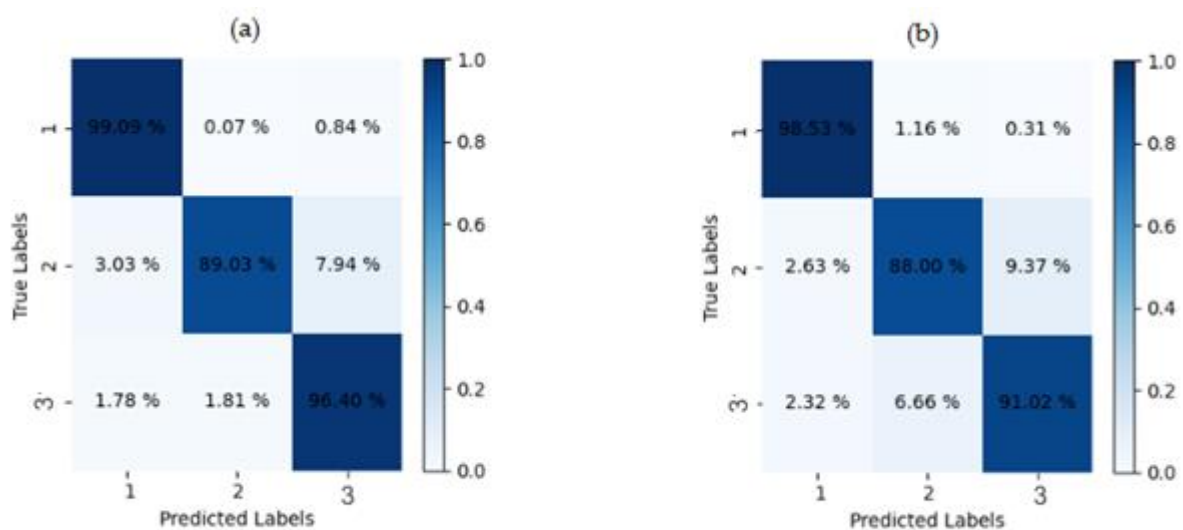
Table 4.6: Comparison of (Cloud_DRL, V.Edge_DL) with baseline.

<i>Classification Techniques</i>	<i>Accuracy</i>	<i>Sensitivity</i>	<i>Specificity</i>	<i>Precision</i>	<i>F1 score</i>
<i>Cloud_DRL</i>	0.94	0.94	0.97	0.94	0.94
<i>V.Edge_DL</i>	0.92	0.92	0.96	0.92	0.92
<i>LR</i>	0.79	0.79	0.89	0.79	0.79
<i>SVM</i>	0.80	0.80	0.84	0.79	0.79
<i>DT</i>	0.88	0.88	0.94	0.89	0.88
<i>RF</i>	0.91	0.91	0.95	0.92	0.91
<i>XGBoost</i>	0.91	0.91	0.94	0.91	0.91



Figure 4.11: Visual comparison with baselines.

- The confusion matrix is one of the most often used metrics for evaluating classification performance using predicted and actual or real labels. It is used to assess the model's capacity to handle ambiguous labels. Figure 4.12 DRL(a), DL(b), LR(c), SVM(d), DT(e), RF(f), XGboost(g) depict the confusion matrix (CM) of the various techniques utilized.



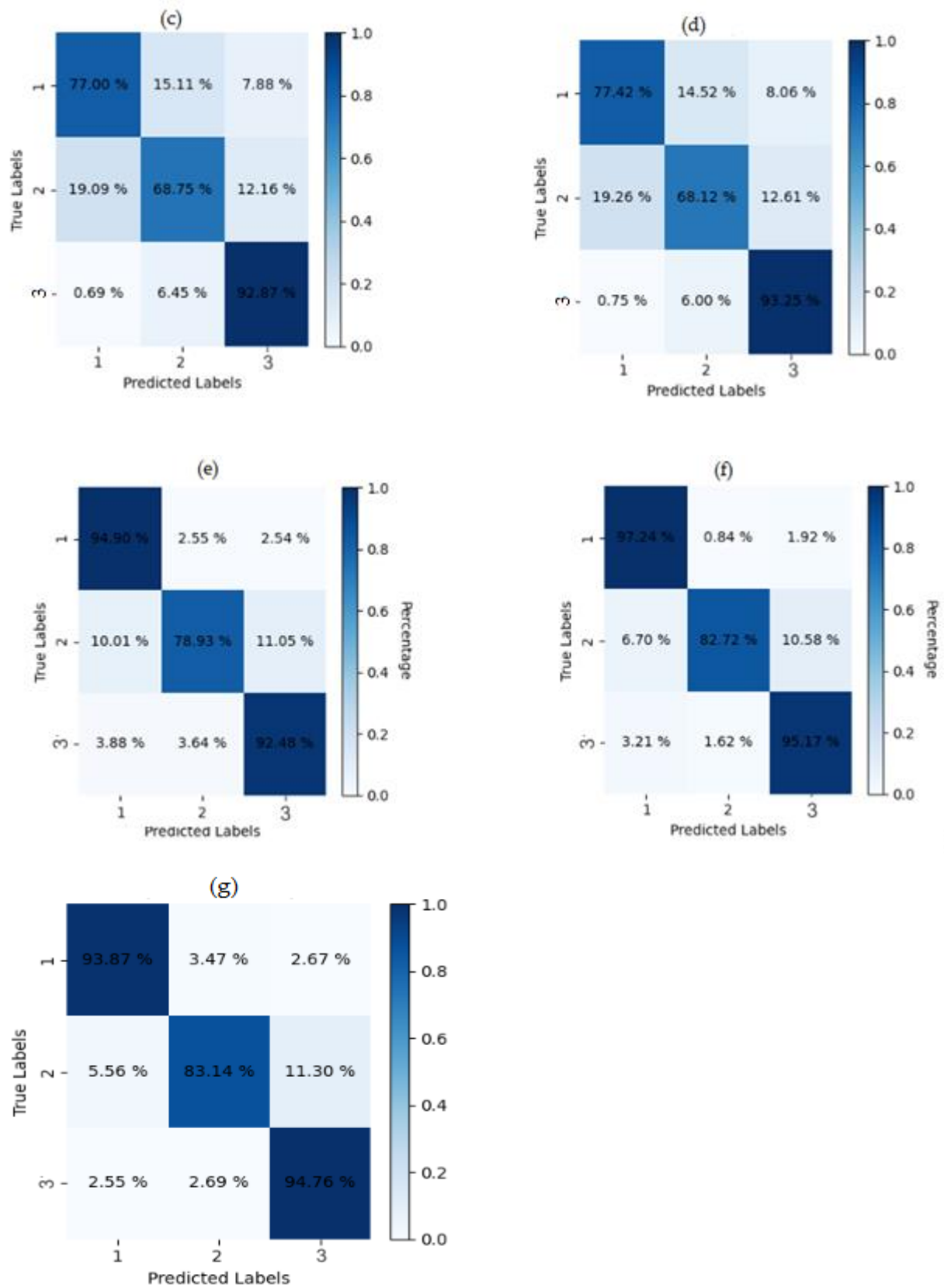


Figure 4.12: Confusion Matrix Results of: DRL(a), DL(b), LR(c), SVM(d), DT (e), RF(f), XGboost(g).

4.8.4 Cloud-Edge Collaboration: A Flask-Based Implementation

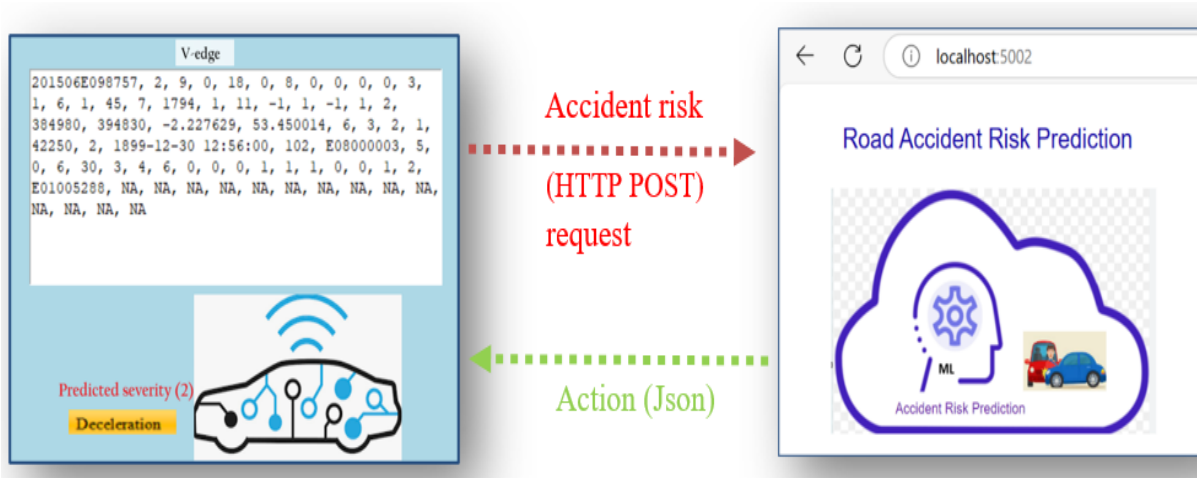


Figure 4.13: Cloud-Edge Communication Architecture for Accident Risk Prediction – Flask Implementation.

This research proposes a novel cloud-edge collaboration framework for real-time vehicular accident risk prediction, leveraging the synergy between distributed edge computing and centralized cloud resources. The system integrates real-time data acquisition from vehicular edge nodes equipped with embedded sensors and advanced predictive analytics executed on a cloud server. By employing a deep reinforcement learning (DRL) algorithm, the framework predicts accident severity (classified as fatal, serious, or slight) and generates actionable recommendations (e.g., stop, decelerate, or maintain speed) to mitigate risks. This hybrid architecture ensures low-latency decision-making at the edge while harnessing the cloud's computational capabilities for complex model inference, addressing the critical need for timely and accurate risk assessment in intelligent transportation systems (ITS).

The proposed system is structured into two core components: the vehicular edge node and the cloud server (localhost). These components are seamlessly integrated using a RESTful API implemented with Flask, a lightweight Python web framework, facilitating efficient communication via HTTP POST requests. The following sections detail the design, implementation, and operational mechanisms of each component [105].

1) Vehicular Edge Node

The vehicular edge node serves as the frontline data acquisition unit, simulating the real-time collection of sensor data from an instrumented vehicle. In this implementation, sensor data—

such as speed, acceleration, proximity to obstacles, and environmental conditions—are sourced from a CSV file to emulate real-world vehicular telemetry. This data is transmitted to the cloud server (Localhost) through an HTTP POST request, a method chosen for its ability to securely encapsulate large datasets within the request body, unlike the URL-based GET method. Upon receiving the cloud server’s response a JSON object containing the predicted accident severity and recommended action, the edge node displays this information via a Tkinter-based graphical user interface (GUI) and executes the prescribed action (e.g., deceleration or stopping).

- **HTTP POST Request Mechanics:**
An HTTP POST request is a client-server communication protocol wherein data is embedded in the request body and sent to a designated server endpoint (in this case, `http://localhost:5002/predict`). Unlike GET requests, which append data to the URL and are thus limited in size and security, POST requests offer enhanced privacy and capacity, making them ideal for transmitting sensitive or voluminous sensor data. The edge node’s POST request triggers the cloud server to process the data and return a response, enabling real-time interaction critical for accident prevention.

2) Cloud Server

The cloud server, implemented using Flask, constitutes the analytical backbone of the system. It receives sensor data from vehicular edge nodes via the `/predict` endpoint, preprocesses it into a structured format (e.g., a Pandas Data Frame), and applies a pre-trained DRL model to predict accident severity. The DRL algorithm, trained on historical accident data, dynamically learns optimal action policies to minimize risk, aligning with recent advancements in reinforcement learning for ITS. The predicted severity and corresponding action recommendation are encapsulated in a JSON response and transmitted back to the edge node, ensuring compatibility with heterogeneous systems.

- **Role of Flask:**
Flask facilitates the development of a lightweight, scalable RESTful API that handles HTTP requests efficiently. Upon receiving a POST request, the server extracts the sensor data, performs necessary preprocessing (e.g., normalization, feature extraction), and invokes the DRL model for inference. The use of JSON as the data interchange format enhances interoperability, enabling seamless integration with diverse edge devices. The server’s ability to process multiple requests concurrently underscores its scalability, a critical attribute for deployment in large-scale vehicular networks.

3) Operational Workflow

The collaboration between edge and cloud unfolds as follows:

1. The vehicular edge node collects and transmits sensor data to the cloud server via an HTTP POST request.
2. The Flask-based cloud server processes the request, preprocesses the data, and applies the DRL model to predict accident severity.
3. The server returns a JSON response containing the severity level and recommended action.
4. The edge node interprets the response and executes the action.

4.8.5 Cloud_DRL – V.Edge_DL collaboration

Cloud_DRL, *V.Edge_DL* collaboration leads to efficient and effective prediction of traffic accident risk. The results obtained by each model individually and then together are shown in Figure 4.14.

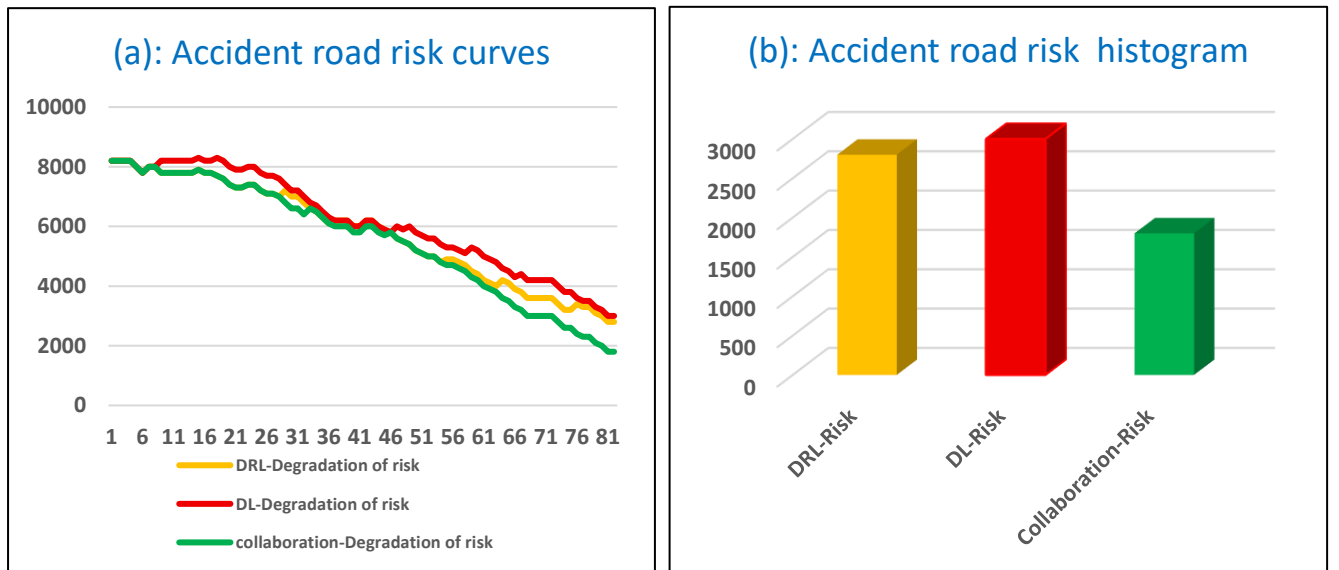


Figure 4. 14: Cloud _DRL - V.Edge_DL Collaboration

4.8.6 Discussion

Figure 4.9 and Figure 4.10 represents the accuracy and the loss of both models *Cloud_DRL* and *V.Edge_DL*. Figure 4.9(a) plots the increment of *Cloud_DRL* accuracy in function of .epoch's number; its accuracy starts very low and ends very high. The main reason of this

distinction is due to a balance between the two explorations and exploitation strategies. At the beginning of the algorithm, each action is performed randomly, which is useful for helping the agent learn more about its environment. Whenever the agent takes more steps, the exploration decreases, and the agent starts to exploit more of the good actions that it has detected. Towards the end of the training process, the search space becomes very limited. Therefore, the agent concentrate more on the exploitation step. This leads to this significant increase in accuracy. It is the same similar justification for the loss curve Figure 4.9(b), which reduces the error to the minimum.

DL has also shown better performance (Figure 4.10) , but DRL remains the strongest in solving dynamic problems where the environment changes over time and the optimal decision-making strategy may vary depending on the state of the environment. This poses challenges for traditional DL algorithm that lack the ability to adapt to changing conditions.

To present how well our accident risk prediction models are performing, we compared them with other algorithms that use the same performance measures. Table 4.6 summarizes the obtained results when applying these machine learning algorithms including *LR*, *SVM*, *DT*, *RF* and *XGBoost*. We note that *Cloud_DRL* and *V.Edge_DL* give a high performance in term of Accuracy, Sensitivity, Specificity, Precision, F1-score measures. We can see that the *Cloud_DRL* achieved the highest degrees of accuracy 94%, Sensitivity94%, Specificity 97%, Precision 94% and F1-score 94%. After *Cloud_DRL* algorithm, the *V.Edge_DL* classifier generates a good result with 92 percent accuracy, 92 percent Sensitivity, 96 percent Specificity, and 92 percent Precision and 92 percent F1-score ; where all the implemented ML methods also perform excellently. Only LR performs relatively poorly with accuracy of less than 80%. In analyzing the classifier outcomes via the confusion matrix (Figure 4.12), the discernment of each model in predicting accident severities became more transparent. The Deep Reinforcement Learning (DRL) algorithm classifier exhibited high accuracy in identifying accident severities, correctly classifying 99% of fatal accidents, 89% of severe accidents, and 96% of slight accidents. These results highlight the effectiveness of the classifier in accurately recognizing different levels of accident severity. Similarly, an alternative Deep Learning (DL) model achieved promising results, with an accuracy of 98% for detecting fatal accidents, 88% for identifying severe accidents, and 91% for recognizing slight accidents. These findings demonstrate strong performance across various levels of accident severity.

In Figure 4.14, we tested our framework on a sample of past road accidents to evaluate its effectiveness in reducing the risk of traffic accidents by using equations (1) and (2) with ($N = 100$), running it through 80 epochs. We started by evaluating each model individually and then combined the two models to demonstrate the importance of their collaboration. As shown in Figure 4.14. The red curve in the graph shows the decrease in the risk rate of road accidents when only *V.Edge_DL* was applied, The risk value decreased from 8200 to 3000 over time; In contrast, when *Cloud_DRL* was used, the risk value decreased further to 2800, as shown by the orange curve. However, the best solution for reducing traffic accident risk was achieved by combining the two models, as demonstrated by the blue curve. With their collaboration, the risk level decreased to almost zero (1800). The same objective is represented by the histogram in Figure 4.14 (b), which shows the level of traffic safety that has been achieved by each model individually and by their collaboration.

4.9 Conclusion

In this chapter, we have demonstrated the effectiveness of our proposal, which involves collaboration between cloud and edge platforms to make a smart city safer and minimize accident rates. We utilized two main machine learning algorithms deep learning and deep reinforcement learning based on neural network concepts. The results obtained are highly satisfactory, as evidenced by the accuracy and loss curves of the two main algorithms, as well as in terms of accuracy, sensitivity, specificity, precision, and F1-score measures. They provide better results compared to classical machine learning algorithms such as logistic regression, decision trees, random forests, and XGBoost. The collaboration between the two algorithms, applied to historical accident data in the UK, also results in a significant reduction in the accident rate.

General Conclusion

This part provides a summary of the contributions made in this thesis and outlines the attained results. Additionally, it proposes directions for future research and improvement.

1. Summary

Ensuring traffic safety in smart cities is a primary focus in the development of urban areas, necessitating innovative methods for managing and preventing traffic accidents.

With recent advancements in Internet-of-Vehicle (IOV) technology, wireless communications, and computational systems, new opportunities have emerged for enhancing road traffic safety. Predicting traffic accident risk plays a pivotal role in achieving this objective, machine learning (ML) is considered as a promising tool for forecasting this risk. These all technologies offer drivers early warnings and valuable information to help them avoid potential hazards on the road. In this framework, we have employed machine learning methodologies, specifically deep learning and deep reinforcement learning algorithms. While deep learning and deep reinforcement learning are highly proficient in addressing intricate challenges within transportation systems, they require substantial resources for optimal performance. Therefore, exploring efficient resource utilization strategies is essential for their effective implementation in enhancing road traffic safety. On the other hand, the data generated by the ensemble of IOV devices on the road is highly voluminous, making their collection and processing within the vehicle impossible. To address this challenge, a collaborative Cloud-V.Edge driver assistance system (ICEDAS) that utilizes machine learning-based IOV has been proposed. To leverage the advantages and mitigate the drawbacks of both platforms, the proposed framework includes two models.

- The first model, CLOUD_DRL, achieves an accuracy of 94% and utilizes a substantial amount of crash data stored in the cloud. It suggests various preventive actions, including stopping, decelerating, or not changing lanes in cases of negligible risk. The cloud serves as a centralized location to aggregate all continuously generated data by IOV devices inside or outside of vehicles, making it an ideal location for training the two proposed ML algorithms (DL and DRL), leveraging the storage and computing capacity provided by this platform.

- The second model, V.Edge_DL, achieves an accuracy of 92% and is deployed at the edge (vehicle) to perform real-time inference. It provides a solution for cases of internet disconnection or bandwidth overload by utilizing locally collected data. This inference involves predicting the likelihood of accidents based on trained patterns and learned correlations. By leveraging edge computing, this model enables faster response times and reduces dependency on cloud resources, thereby enhancing the overall efficiency and effectiveness of the driver assistance system.

To evaluate the effectiveness of the collaborative framework in reducing the risk of accidents, we conducted tests on a randomly selected sample of past road accidents, running it through 80 epochs. The results obtained indicate that the collaboration between the two models significantly reduces the risk (from 8200 to less than 1800), surpassing the performance of either model alone. This demonstrates its capability to decrease the incidence of traffic accidents.

2. Perspective

The future work will also explore the synergies between machine learning and Cloud-Edge Computing to develop advanced systems for preventing accidents and enhancing road safety. By incorporating computer vision technologies, the system will be able to analyze images and videos captured from various sources such as vehicle-mounted cameras, traffic cameras, and drones. This analysis will enable the detection of hazardous situations, identification of risky behaviors among road users, and assessment of real-time road conditions.

Furthermore, the integration of computer vision will enhance the system's ability to detect and respond to complex scenarios on the road, such as identifying pedestrians, cyclists, and other vehicles in challenging environments like low-light conditions or adverse weather. By leveraging the power of machine learning algorithms, the system can continuously learn and improve its accuracy in recognizing and predicting potential risks on the road.

This approach holds great promise for further advancements, particularly in the field of self-driving vehicles. By equipping autonomous vehicles with advanced computer vision capabilities, they can better perceive and understand their surroundings, leading to safer navigation and reduced accident rates.

Moreover, beyond road safety, the system's capabilities can extend to other areas such as fire detection and industrial risk management. By adapting the algorithms and methodologies

General conclusion

developed for road safety applications, the system can be applied to analyze images and videos in industrial settings to detect potential hazards, monitor equipment performance, and ensure workplace safety.

Bibliography

- [1] Ahmed, S., Hossain, M. A., Ray, S. K., Bhuiyan, M. M. I., & Sabuj, S. R. (2023). *A study on road accident prediction and contributing factors using explainable machine learning models: Analysis and performance. Transportation research interdisciplinary perspectives*, 19, 100814.
- [2] Cheng, W., & Washington, S. P. (2005). *Experimental evaluation of hotspot identification methods. Accident Analysis & Prevention*, 37(5), 870-881.
- [3] Cui, H., Dong, J., Zhu, M., Li, X., & Wang, Q. (2022). *Identifying accident black spots based on the accident spacing distribution. Journal of traffic and transportation engineering (English edition)*, 9(6), 1017-1026.
- [4] Elvik, R. (2007). *State-of-the-art approaches to road accident black spot management and safety analysis of road networks. Institute of Transport Economics Norwegian Centre for Transport Research*, Oslo.
- [5] Kumar Thangavel, R., Athithan, S., Sarumathi, S., Aruna, M., & Nithila, B. (2019, July). *Blackspot alert and accident prevention system. In 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)* (pp. 1-6). IEEE.
- [6] Yang, H., Wang, Y., Zhao, H., Zhu, J., & Wang, D. (2020, September). *Real-time traffic incident detection using an autoencoder model. In 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)* (pp. 1-6). IEEE.
- [7] Labib, M. F., Rifat, A. S., Hossain, M. M., Das, A. K., & Nawrine, F. (2019, June). *Road accident analysis and prediction of accident severity by using machine learning in Bangladesh. In 2019 7th international conference on smart computing & communications (ICSCC)* (pp. 1-5). IEEE.
- [8] Paul, J., Jahan, Z., Lateef, K. F., Islam, M. R., & Bakchy, S. C. December, (2020). *Prediction of road accident and severity of bangladesh applying machine learning techniques. In 2020 IEEE 8th R10 Humanitarian Technology Conference (R10-HTC)* (pp. 1-6). IEEE.
- [9] Thaduri, A., Polepally, V., & Vodithala, S. (2021). *Traffic accident prediction based on CNN model. In 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 1590-1594). IEEE.
- [10] Ren, H., Song, Y., Liu, J., Hu, Y., & Lei, J. (2017). *A deep learning approach to the prediction of short-term traffic accident risk. arXiv preprint arXiv:1710.09543*.

- [11] R. Arunachalam, S. Peararulsevi, M. Saraswathi, M. Subasri. 2023. *Road Accident Severity Prediction using Machine Learning*, IJARSCT.
- [12] Yang, K., Sun, P., Yang, D., Lin, J., Boukerche, A., & Song, L. (2024). *A novel hierarchical distributed vehicular edge computing framework for supporting intelligent driving*. *Ad Hoc Networks*, 153, 103343.
- [13] Mahesh, B. (2020). *Machine learning algorithms-a review*. International Journal of Science and Research (IJSR). [Internet], 9(1), 381-386.
- [14] Ghosh, P., Kiran, S., Mahalakshmi, J., & Basha, S. A. H. (2023). *Understanding machine learning*. Ag Publishing House (AGPH Books).
- [15] Hiran, K. K., Jain, R. K., Lakhwani, K., & Doshi, R. (2021). *Machine Learning: Master Supervised and Unsupervised Learning Algorithms with Real Examples (English Edition)*. BPB Publications.
- [16] Shukla, A., Agarwal, A., Pant, H., & Mishra, P. (2020). *Flower classification using supervised learning*. *Int. J. Eng. Res*, 9(05), 757-762.
- [17] Gupta, V., T Mishra, V. K., Singhal, P., & Kumar, A. (2022, December). *An overview of supervised machine learning algorithm*. In *2022 11th International Conference on System Modeling & Advancement in Research Trends (SMART)* (pp. 87-92). IEEE.
- [18] Yu, F. R., & He, Y. (2019). *Deep reinforcement learning for wireless networks*. Springer.
- [19] Saravanan, R., & Sujatha, P. (2018, June). *A state of art techniques on machine learning algorithms: a perspective of supervised learning approaches in data classification*. *2018 Second international conference on intelligent computing and control systems (ICICCS)* (pp. 945-949). IEEE.
- [20] Mahesh, B. (2020). *Machine learning algorithms-a review*. *International Journal of Science and Research (IJSR)*. [Internet], 9(1), 381-386.
- [21] Debbadi, R. K., & Boateng, O. (2025). *Developing intelligent automation workflows in Microsoft Power Automate by embedding deep learning algorithms for real-time process adaptation*. *Int J Sci Res Arch*, 14(2), 802-820.
- [22] Ke, K. C., & Huang, M. S. (2020). *Quality prediction for injection molding by using a multilayer perceptron neural network*. *Polymers*, 12(8), 1812.
- [23] Harthik, Jayakanna, S., Manoj, Raju. (2022). *'A Study on Deep Learning*. *International Journal For Science Technology And Engineering'*.

- [24] Shiri, F. M., Perumal, T., Mustapha, N., & Mohamed, R. (2023). *A comprehensive overview and comparative analysis on deep learning models: CNN, RNN, LSTM, GRU*. *arXiv preprint arXiv:2305.17473*.
- [25] Han, X. (2018). *A mathematical introduction to reinforcement learning*. *Semantic Scholar*, 1-4.
- [26] van der Wilk, M. M. (2021). *Deep reinforcement learning for global network management*.
- [27] Van Otterlo, M. (2009). *Markov decision processes: Concepts and algorithms*. *Course on 'Learning and Reasoning'*.
- [28] Zheng, W. (2022). *Deep Reinforcement Learning for Building Control: A comparative study for applying Deep Reinforcement Learning to Building Energy Management*.
- [29] Ravichandiran, S. (2020). *Deep Reinforcement Learning with Python: Master classic RL, deep RL, distributional RL, inverse RL, and more with OpenAI Gym and TensorFlow*. Packt Publishing Ltd.
- [30] Nack, F. (2010). 'An overview on wireless sensor networks. *Institute of Computer Science (ICS)*', Freie Universität Berlin, 6.
- [31] Senouci, M. R., & Mellouk, A. (2016). 'Deploying wireless sensor networks: theory and practice'. Elsevier .
- [32] Ahmad, R., Wazirali, R., & Abu-Ain, T. (2022). 'Machine learning for wireless sensor networks security: An overview of challenges and issues, *Sensors*, 22(13), 4730.
- [33] M.M. Ahsan, M. Hasanuzzaman, A.G. Olabi, M.S.J. Hashmi, 2014, 'Review of the Reliability and Connectivity of Wireless Sensor Technology ', *Comprehensive Materials Processing*, Elsevier.
- [34] Majid, M., Habib, S., Javed, A. R., Rizwan, M., Srivastava, G., Gadekallu, T. R., & Lin, J. C. W. (2022). 'Applications of wireless sensor networks and internet of things frameworks in the industry revolution 4.0: A systematic literature review'. *Sensors*, 22(6), 2087.
- [35] Hassan, K., Madkour, M. A., & Nouh, S. A. (2023). 'A Review Of Security Challenges And Solutions In Wireless Sensor Networks. *Journal of Al-Azhar University Engineering Sector*', 18(69), 914-938.
- [36] Sharma, H., Haque, A., & Blaabjerg, F. (2021). *Machine learning in wireless sensor networks for smart cities: a survey*. *Electronics*, 10(9), 1012.
- [37] Hankare, P., Babar, S., & Mahalle, P. (2021). *Trust management approach for detection of malicious devices in siot*. *Tehnički glasnik*, 15(1), 43-50.

- [38] Alsheikh, M. A., Lin, S., Niyato, D., & Tan, H. P. (2014). ' *Machine learning in wireless sensor networks: Algorithms, strategies, and applications* '. IEEE Communications Surveys & Tutorials, 16(4), 1996-2018.
- [39] Gupta, N., Prakash, A., & Tripathi, R. (Eds.). (2021). *Internet of vehicles and its applications in autonomous driving*. Berlin/Heidelberg, Germany, Springer.
- [40] Chen, M., Jiang, Y., Huang, J., Ou, W., Han, W., & Zhang, Q. (2024). *An attribute-encryption-based cross-chain model in urban internet of vehicles*. *Computers and Electrical Engineering*, 115, 109136.
- [41] Amadeo, M., Campolo, C., & Molinaro, A. (2016). *Priority-based content delivery in the Internet of vehicles through named data networking*. *Journal of Sensor and Actuator Networks*, 5(4), 17.
- [42] Magaia, N., Mastorakis, G., Mavromoustakis, C. X., Pallis, E., & Markakis, E. K. (Eds.). (2021). *Intelligent technologies for internet of vehicles*. Berlin/Heidelberg, Germany: Springer
- [43] Kim, S., & Shrestha, R. (2020). *Automotive cyber security: introduction, challenges, and standardization* (pp. 149-181). Singapore, Springer.
- [44] Sadiku, M. N., Tembely, M., & Musa, S. M. (2018). *Internet of vehicles: An introduction*. *International Journal of Advanced Research in Computer Science and Software Engineering*, 8(1), 11.
- [45] Patil, R. R., Kulkarni, G., & Dongare, A. (2018). *Cloud computing an overview*. *International Journal of Engineering and Technology*, 7(4), 2743.
- [46] Chandrasekaran, K. (2014). *Essentials of cloud computing*. CrC Press.
- [47] Srinivas, J., Reddy, K. V. S., & Qyser, A. M. (2012). *Cloud computing basics*. *International journal of advanced research in computer and communication engineering*, 1(5), 343-347.
- [48] Cao, J., Zhang, Q., & Shi, W. (2018). *Edge computing: a primer*. Berlin/Heidelberg, Germany: Springer International Publishing.
- [49] Liu, L., Chen, C., Pei, Q., Maharjan, S., & Zhang, Y. (2021). *Vehicular edge computing and networking: A survey*. *Mobile networks and applications*, 26, 1145-1168.
- [50] Qi, Q., & Tao, F. (2019). *A smart manufacturing service system based on edge computing, fog computing, and cloud computing*. *IEEE access*, 7, 86769-86777.
- [51] Mhetre, K. V., & Thube, A. D. (2022). *Development of Road Safety Models by Using Linear and Logistic Regression Modeling Techniques*. In *Recent Trends in Construction*

Technology and Management: Select Proceedings of ACTM 2021 (pp. 1205-1225). Singapore: Springer Nature Singapore.

[52] Alicioglu, G., Sun, B., & Ho, S. S. (2020, July). *Assessing accident risk using ordinal regression and multinomial logistic regression data generation*. In *2020 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-8). IEEE.

[53] Vilaça, M., Macedo, E., Tafidis, P., & Coelho, M. C. (2019). *Multinomial logistic regression for prediction of vulnerable road users risk injuries based on spatial and temporal assessment*. *International journal of injury control and safety promotion*, 26(4), 379-390.

[54] Bekelcho, T., Olani, A. B., Woldemeskel, A., Alemayehu, M., & Guta, G. (2023). *Identification of determinant factors for crash severity levels occurred in Addis Ababa City, Ethiopia, from 2017 to 2020: using ordinal logistic regression model approach*. *BMC public health*, 23(1), 1884.

[55] Khanum, H., Garg, A., & Faheem, M. I. (2023). *Accident severity prediction modeling for road safety using random forest algorithm: an analysis of Indian highways*. *F1000 Research*, 12.

[56] Zhong, W., & Du, L. (2023). *Predicting Traffic Casualties Using Support Vector Machines with Heuristic Algorithms: A Study Based on Collision Data of Urban Roads*. *Sustainability*, 15(4), 2944.

[57] Zong, F., Chen, X., Tang, J., Yu, P. & Wu, T., (2019). *Analyzing Traffic Crash Severity With Combination of Information Entropy and Bayesian Network*. *IEEE Access*, pp. 63288–63302.

[58] Kumar, U. S., & Jeyavathana, R. B. (2021). *An efficient analysis of road accidents severity using novel support vector machines over artificial neural networks with improved accuracy rate*. *Revista Geintec-Gestao Inovacao E Tecnologias*, 11(2), 1109-1122.

[59] Vishnu, A., Sushmitha, S., Jacob, T. S., David Maxim Gururaj, A., & Dhanasekar, S. (2022, March). *Automated Road Surveillance System Using Machine Learning*. In *International Conference on Big data and Cloud Computing* (pp. 67-80). Singapore: Springer Nature Singapore.

[60] Schwarz, M., & Buhmann, A. (2021, June). *Safety application car crash detection using multiclass support vector machine*. In *2021 28th International Conference on Mixed Design of Integrated Circuits and System* (pp. 238-248). IEEE.

[61] Vanitha, R. (2023). *Prediction of Road Accidents Using Machine Learning Algorithms*. *Middle East Journal of Applied Science & Technology (MEJAST)*, 6(2), 64-75.

- [62] Masoumi, k., Forouzan, A., Barzegari, H., Asgari Darian, A., Rahim, F., Zohrevandi, B. & Nabi, S., (2018). *Effective Factors in Severity of Traffic accident-related Traumas an Epidemiologic Study Based on the Haddon Matrix*. Emergency (Tehran, Iran), pp. 78–82.
- [63] Wadhahi, N.T.S.A., Hussain, S.M., Yosof, k.M., Hussain, S.A., & Singh, A.V., (2018). *Accidents Detection and Prevention System to reduce Traffic Hazards using IR Sensors. 20187th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, pp. 254-257.
- [64] Zhang, J., Li, Z., Pu, Z., & Xu, C., (2018). *Comparing Prediction Performance for Crash Injury Severity among Various Machine Learning and Statistical Methods*. *IEEE Access*, pp.1–1.
- [65] Guang Yu, Shao lu, (2019). *Speed and lane changing control optimization research on accidentand congestion concentrated section of guang-shen freeway*, *IEEE Access*, pp. 1-8.
- [66] Gatera, A., Kuradusenge, M., Bajpai, G., Mikeka, C., & Shrivastava, S. (2023). *Comparison of random forest and support vector machine regression models for forecasting road accidents*. *Scientific African*, 21, e01739.
- [67] Hossain, M., & Muromachi, Y. (2012). *A Bayesian network based framework for real-time crash prediction on the basic freeway segments of urban expressways*. *Accident Analysis & Prevention*, 45, 373-381.
- [68] Rezapour, M., Nazneen, S., & Ksaibati, K. (2020). *Application of deep learning techniques in predicting motorcycle crash severity*. *Engineering Reports*, 2(7), e12175.
- [69] Zheng, M., Li, T., Zhu, R., Chen, J., Ma, Z., Tang, M., ... & Wang, Z. (2019). *Traffic accident's severity prediction: A deep-learning approach-based CNN network*. *IEEE Access*, 7, 39897-39910.
- [70] Sameen, M. I., Pradhan, B., Shafri, H. Z. M., & Hamid, H. B. (2019). *Applications of deep learning in severity prediction of traffic accidents*. In *GCEC 2017: Proceedings of the 1st Global Civil Engineering Conference 1* (pp. 793-808). Springer Singapore.
- [71] Wenqi, L., Dongyu, L., & Menghua, Y. (2017, September). *A model of traffic accident prediction based on convolutional neural network*. In *2017 2nd IEEE international conference on intelligent transportation engineering (ICITE)* (pp. 198-202). IEEE.
- [72] Chen, Q., Song, X., Yamada, H., & Shibasaki, R. (2016, February). *Learning deep representation from big and heterogeneous data for traffic accident inference*. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 30, No. 1).

- [73] Ren, H., Song, Y., Liu, J., Hu, Y., & Lei, J. (2017). *A deep learning approach to the prediction of short-term traffic accident risk*. *arXiv preprint arXiv:1710.09543*.
- [74] Yuan, Z., Zhou, X., & Yang, T. (2018, July). *Hetero-convlstm: A deep learning approach to traffic accident prediction on heterogeneous spatio-temporal data*. In Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining (pp. 984-992).
- [75] Yu, L., Du, B., Hu, X., Sun, L., Han, L., & Lv, W. (2021). *Deep spatio-temporal graph convolutional network for traffic accident prediction*. *Neurocomputing*, 423, 135-147.
- [76] Shan, S. E., Faisal, M. F., Haque, S. R., & Saha, P. (2018, February). *IoT and computer vision based driver safety monitoring system with risk prediction*. In 2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2) (pp. 1-4). IEEE.
- [77] Basso, F., Pezoa, R., Varas, M., & Villalobos, M. (2021). *A deep learning approach for real-time crash prediction using vehicle-by-vehicle data*. *Accident Analysis & Prevention*, 162, 106409.
- [78] Jin, Z., Noh, B., Cho, H., & Yeo, H. (2022, October). *Deep Learning-based Approach on Risk Estimation of Urban Traffic Accidents*. In 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC) (pp. 1446-1451). IEEE.
- [79] Arase, K., Wu, Z., Migita, T., & Takahashi, N. (2022, November). *Deep Learning of OpenStreetMap Images Labeled Using Road Traffic Accident Data*. In TENCON 2022-2022 IEEE Region 10 Conference (TENCON) (pp. 1-6). IEEE.
- [80] Lai, S., Zhou, C., Song, X., & Zhang, X. (2021, September). *A traffic accident risk prediction model based on spatial gated memory network*. In 2021 International Conference on Electronic Information Engineering and Computer Science (EIECS) (pp. 331-335). IEEE.
- [81] Manzoor, M., Umer, M., Sadiq, S., Ishaq, A., Ullah, S., Madni, H. A., & Bisogni, C. (2021). *RFCNN: Traffic accident severity prediction based on decision level fusion of machine and deep learning model*. *IEEE Access*, 9, 128359-128371.
- [82] Kanakala, R., & Reddy, K. (2023). *Modelling a deep network using CNN and RNN for accident classification*. *Measurement: Sensors*, 27, 100794.
- [83] Wan, S., Gu, Z., & Ni, Q. (2020). *Cognitive computing and wireless communications on the edge for healthcare service robots*. *Computer Communications*, 149, 99-106.

- [84] Tolba, A., Said, O., & Al-Makhadmeh, Z. (2019). *MDS: Multi-level decision system for patient behavior analysis based on wearable device information*. *Computer Communications*, 147, 180-187.
- [85] Usama, M., Ahmad, B., Yang, J., Qamar, S., Ahmad, P., Zhang, Y., ... & Guna, J. (2019). *REMOVED: Equipping recurrent neural network with CNN-style attention mechanisms for sentiment analysis of network reviews*. *Computer Communications*, 148, 98.
- [86] Al-Ayyoub, M., Nuseir, A., Alsmearat, K., Jararweh, Y., & Gupta, B. (2018). *Deep learning for Arabic NLP: A survey*. *Journal of computational science*, 26, 522-531.
- [87] Gregurić, M., Vujić, M., Alexopoulos, C., & Miletić, M. (2020). *Application of deep reinforcement learning in traffic signal control: An overview and impact of open traffic data*. *Applied Sciences*, 10(11), 4011.
- [88] Cho, I., Rajendran, P. K., Kim, T., & Har, D. (2023, February). *Reinforcement learning for predicting traffic accidents*. In 2023 International Conference on Artificial Intelligence in Information and Communication (ICAIIC) (pp. 684-688). IEEE.
- [89] Bao, W., Yu, Q., & Kong, Y. (2021). *Drive: Deep reinforced accident anticipation with visual explanation*. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 7619-7628).
- [90] Sharif, A., & Marijan, D. (2022, December). *Evaluating the robustness of deep reinforcement learning for autonomous policies in a multi-agent urban driving environment*. In 2022 IEEE 22nd International Conference on Software Quality, Reliability and Security (QRS) (pp. 785-796). IEEE.
- [91] Zhao J, Qu T, Xu F (2020) A deep reinforcement learning approach for autonomous highway driving. *IFAC-PapersOnLine* 53(5):542–546.
- [92] Roy, A., Hossain, M., & Muromachi, Y. (2022). *A deep reinforcement learning-based intelligent intervention framework for real-time proactive road safety management*. *Accident Analysis & Prevention*, 165, 106512.
- [93] Muzahid, A. J. M., Kamarulzaman, S. F., Rahman, M. A., & Alenezi, A. H. (2022). *Deep reinforcement learning-based driving strategy for avoidance of chain collisions and its safety efficiency analysis in autonomous vehicles*. *IEEE Access*, 10, 43303-43319.
- [94] Huynh, A. T., Nguyen, B. T., Nguyen, H. T., Vu, S., & Nguyen, H. D. (2021). *A Method of Deep Reinforcement Learning for Simulation of Autonomous Vehicle Control*. In ENASE (pp. 372-379).

- [95] Nagesh Rao, S., Tseng, H. E., & Filev, D. (2019, October). *Autonomous highway driving using deep reinforcement learning*. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)* (pp. 2326-2331). IEEE.
- [96] Ashwin, S. H., & Naveen Raj, R. (2023). *Deep reinforcement learning for autonomous vehicles: lane keep and overtaking scenarios with collision avoidance*. *International Journal of Information Technology*, 15(7), 3541-3553.
- [97] Ma, H. (2022, October). *Algorithm Optimization of Deep Reinforcement Learning for Traffic Signal Control of Municipal Road Engineering*. In *2022 4th International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM)* (pp. 829-833). IEEE.
- [98] Kiswani, J. H., Dascalu, S. M., & Harris Jr, F. C. (2021). *Cloud computing and its applications: A comprehensive survey*. *International Journal of Computer Applications IJCA*, 28(1), 3-24.
- [99] Meneguette, R., De Grande, R., Ueyama, J., Filho, G. P. R., & Madeira, E. (2021). *Vehicular edge computing: Architecture, resource management, security, and challenges*. *ACM Computing Surveys (CSUR)*, 55(1), 1-46.
- [100] Murad, A., Kraemer, F. A., Bach, K., & Taylor, G. (2020, October). *Information-driven adaptive sensing based on deep reinforcement learning*. In *Proceedings of the 10th International Conference on the Internet of Things* (pp. 1-8).
- [101] Gao, J., Shen, Y., Liu, J., Ito, M., & Shiratori, N. (2017). *Adaptive traffic signal control: Deep reinforcement learning algorithm with experience replay and target network*. arXiv preprint arXiv:1705.02755.
- [102] Chae, H., Kang, C. M., Kim, B., Kim, J., Chung, C. C., & Choi, J. W. (2017, October). *Autonomous braking system via deep reinforcement learning*. In *2017 IEEE 20th International conference on intelligent transportation systems (ITSC)* (pp. 1-6). IEEE.
- [103] Yang, C., Wang, Y., Lan, S., Wang, L., Shen, W., & Huang, G. Q. (2022). *Cloud-edge-device collaboration mechanisms of deep learning models for smart robots in mass personalization*. *Robotics and Computer-Integrated Manufacturing*, 77, 102351.
- [104] Tatsat, H., Puri, S., & Lookabaugh, B. (2020). *Machine Learning and Data Science Blueprints for Finance*. O'Reilly media.
- [105] Conroy, K., Horn, R., & Stratton, F. (2020). *Flask-RESTful Documentation* (Release 0.3.7).