Titre: Title:	Congestion Prediction and Avoidance Mechanisms for Heterogeneous Vehicular Networks
Auteur: Author:	Farnoush Falahatraftar
Date:	2021
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Falahatraftar, F. (2021). Congestion Prediction and Avoidance Mechanisms for Heterogeneous Vehicular Networks [Thèse de doctorat, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/9744/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/9744/
Directeurs de recherche: Advisors:	Samuel Pierre, & Steven Chamberland
Programme: Program:	Génie informatique

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Congestion Prediction and Avoidance Mechanisms for Heterogeneous Vehicular Networks

FARNOUSH FALAHATRAFTAR
Département de génie informatique et génie logiciel

Thèse présentée en vue de l'obtention du diplôme de *Philosophiæ Doctor*
Génie informatique

Novembre 2021

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Cette thèse intitulée :

Congestion Prediction and Avoidance Mechanisms for Heterogeneous Vehicular Networks

présentée par **Farnoush FALAHATRAFTAR**
en vue de l'obtention du diplôme de *Philosophiæ Doctor*
a été dûment acceptée par le jury d'examen constitué de :

Alejandro QUINTERO, président
Samuel PIERRE, membre et directeur de recherche
Steven CHAMBERLAND, membre et codirecteur de recherche
Soumaya CHERKAOUI, membre
Abderrezak RACHEDI, membre externe

DEDICATION

To my beloved mother soul Farah,
for all the self-sacrifices that gladly made out of love for me...

ACKNOWLEDGEMENTS

From the bottom of my heart I would like to express my gratitude and appreciation for my research director Professor Samuel Pierre whose guidance, continued support and encouragement has been invaluable throughout my Ph.D. study. Thank you for giving me the opportunity of being your student, to grow and learn day by day in Mobile Computing and Network Research Laboratory (LARIM). Your humanity, kindness and attitude as well as your big achievements have been a source of inspiration for me.

I would like to specially thank my co-supervisor Professor Steven Chamberland for his guidance, the insightful comments and suggestions on this dissertation. Moreover, I am grateful to Professor Alejandro Quintero, Professor Soumaya Cherkaoui, and Professor Abderrezak Rachedi for kindly accepting to participate in this jury.

I would like to say big thank you to my colleagues and friends at LARIM for the friendly and collaborative environment that we had. My sincere gratitude goes to Dre. Franjieh El Khoury for her kindness, energy, collaboration and help throughout my dissertation. Many thanks to Marième, Eric, Amir, Sepehr, Lamia, Nasrin, Sanaz, Parvin, Olson, Claudy, Jean, Loïc, Marc, Carria and Nick for their helpful comments on my research and their contribution to make a friendly and affirmative laboratory. I truly had a very very good time in LARIM.

I wish to express my deepest gratitude to my angels, my parents Farah and Mehdi for their infinite support, sacrifice and efforts. I owe to them all that I am. My brother Farhoud and my sister Farnaz, I do not know how to say thank you for all your love, support and attention that you always sent to me during the years of my study. I could feel all of them with my heart even miles away from you, thank you for everything.

Most importantly, I would like to say a special thank you to my dearest, my companion, my source of happiness, my friend and my spouse Mehrdad for his unconditional love, endless support and understanding.

RÉSUMÉ

Le réseau hétérogène de véhicules « Heterogeneous Vehicular Network » (HetVNET) est conçu pour fournir des services de sécurité et de non-sécurité en utilisant la communication dédiée à courte portée (DSRC) pour les communications de véhicule à véhicule « Dedicated Short-Range Communication » (V2V) et l'évolution à long terme « Long Term Evolution » (LTE) pour les communications de véhicule à infrastructure « Vehicle-to-Infrastructure » (V2I). HetVNET peut fournir une large couverture avec un débit de données élevé via LTE et un échange de données en temps réel via DSRC. D'après la littérature, HetVNET peut fournir divers services nécessaires au système de transport intelligent (ITS). Toutefois, dans un réseau HetVNET dense, les énormes données générées par les véhicules doivent disposer de suffisamment de ressources pour être acheminées correctement et avec une faible latence. Dans le cas contraire, les données sont reçues à destination avec un retard important ou peuvent même être abandonnées par les dispositifs du réseau. Cette situation conduit à la congestion du réseau. Par conséquent, le problème de congestion du réseau affecte négativement la qualité des services « Quality of Services » (QoS) et les performances du réseau.

Dans les réseaux véhiculaires, la gestion de la congestion comporte deux phases principales : la détection de la congestion et le contrôle de la congestion. La plupart des détections de congestion sont basées sur des seuils prédéfinis pour le niveau d'utilisation des canaux ou le nombre de véhicules. Dans la littérature, les chercheurs ont fait des efforts significatifs pour la phase de contrôle de la congestion et pour la phase de détection de la congestion. Ils ont utilisé des hypothèses, par exemple, lorsque plus de 70% du canal de communication est utilisé ou lorsque le taux de génération de données dépasse une valeur prédéfinie. Cependant, trouver la valeur optimale pour le seuil reste un défi car il peut y avoir un risque de sous-utilisation du canal. De plus, l'utilisation d'un seuil élevé pour la détection de la congestion pourrait augmenter la perte de paquets dans un environnement véhiculaire très dense.

Dans la littérature, les méthodes proposées pour contrôler la congestion du réseau sont centralisées ou distribuées. Dans la méthode centralisée, une unité centrale est chargée de prendre les décisions de contrôle et les véhicules doivent obéir aux commandes de l'unité centrale. Dans les méthodes distribuées, les véhicules décident de la manière de contrôler la congestion du réseau, en se basant principalement sur les informations reçues des autres véhicules environnants. La surveillance, la collecte et l'analyse des données nécessitent des ressources de calcul et de stockage suffisamment robustes. Sinon, des frais généraux et des

retards élevés sont inévitables. Néanmoins, les méthodes distribuées sont largement considérées dans la littérature. Cependant, l'émergence de technologies récentes tels que le réseau défini par logiciel (SDN), l'informatique en brouillard et la virtualisation des réseaux, qui peuvent fournir une programmabilité, une vue globale et des ressources de calcul, de réseau et de stockage appropriées, encouragent les auteurs à considérer les méthodes centralisées plus qu'auparavant.

Dans cette thèse, la première phase de la gestion de la congestion est considérée mais d'une manière différente. Des méthodes d'intelligence artificielle (IA) sont envisagées pour générer des méthodes de prédiction de la congestion pour HetVNET. Le problème de prédiction est défini dans les deux styles de problème de classification et de régression. Les données simulées extraites de la simulation de la mobilité urbaine « Simulation of Urban Mobility » (SUMO) et du simulateur Veins LTE sont utilisées comme données d'entrée des méthodes d'apprentissage automatique « Machine Learning » (ML) supervisées.

Les classes d'alerte et de non-alerte sont définies à l'aide du taux de livraison des données « Data Delivery Ratio » (DDR) et de l'intensité du signal reçu « Received Signal Strength » (RSS). L'algorithme Naïve Bayes est appliqué pour prédire les états d'alerte et de non-alerte de congestion du réseau HetVNET. En outre, une architecture centralisée et dynamique de nuage-brouillard est proposée. Cette architecture comporte deux composants principaux : une unité de gestion centralisée « Centralized Management Unit » (CMU) et des unités de prédiction de la congestion du brouillard « Fog Congestion Predictor Units » (FCPU). La méthode de classification d'alerte de congestion Naïve Bayes proposée peut être appliquée dans la FCPU. De plus, un mécanisme de classification centralisée de la congestion du réseau « Network Congestion Classification » (CNCC) est proposé pour prouver l'efficacité de l'approche de classification proposée pour éviter la congestion du réseau dans HetVNET. Les résultats montrent que la méthode proposée peut améliorer les performances du réseau en termes de taux de perte de paquets, de retard moyen et de débit moyen.

De plus, une fonction d'utilité est proposée dans cette thèse. Dans cette étape, l'objectif de la méthode de prédiction par régression est de prédire la valeur de la fonction d'utilité. Un modèle de prédiction par régression linéaire multiple « Multiple Linear Regression » (MLR) et une méthode de prédiction par réseau neuronal de régression généralisée « Generalized Regression Neural Network » (GRNN) sont proposés pour prédire la valeur de la fonction d'utilité. La valeur prédite fournit une vision de l'avenir et peut être utilisée comme un indice pour la gestion de congestion des réseaux en prenant des décisions et en appliquant des politiques. Cette approche permet de créer une gestion de congestion intelligente pour un HetVNET adaptatif et autonome. En outre, le mécanisme de prévention de la congestion intelligente « Intelligent Congestion Avoidance Mechanism » (ICAM), qui est une technique de transmission adaptative puissante utilisant la méthode de prédiction GRNN, a été proposé. Les résultats obtenus prouvent que GRNN est une méthode précise, fiable et stable. De plus, les résultats de simulation montrent que l'ICAM peut améliorer les performances du réseau en termes de taux de perte de paquets et de délai moyen.

En ce qui concerne le concept de découpage du réseau dans la 5g et au-delà, une méthode « Conditional Generative Adversarial Network » (CGAN) est appliquée pour générer des tranches de réseau dans HetVNET. Dans cette partie de la thèse, une architecture hybride est proposée en utilisant les technologies SDN et « Network Function Virtualization » (NFV). L'objectif de cette partie de la thèse est de générer dynamiquement des configurations pour les tranches de réseau qui ont un faible potentiel d'occurrence de congestion du réseau.

Les méthodes proposées basées sur l'IA sont entraînées et testées à l'aide du langage de programmation Python. La précision et la fiabilité des méthodes proposées ont ensuite été évaluées. Les résultats obtenus montrent que les méthodes proposées basées sur l'IA (Naïve Bayes, GRNN et CGAN) peuvent accomplir leurs tâches avec précision. En outre, les résultats de la simulation montrent que l'application du CNCC et de l'ICAM proposés pourrait améliorer les performances du réseau.

ABSTRACT

Heterogeneous Vehicular Network (HetVNET) is designed to provide safety and non-safety services using Dedicated Short Range Communication (DSRC) for Vehicle-to-Vehicle (V2V) and Long Term Evolution (LTE) for Vehicle-to-Infrastructure (V2I) communications. HetVNET can provide a wide coverage range with high data rate via LTE and real-time data exchange via DSRC. Based on the literature, HetVNET may well provide various required services of the Intelligent Transportation System (ITS). However, in a dense HetVNET, enormous data generated by vehicles need enough available resources to be successfully delivered with low latency. Otherwise, the data received in destinations with notably delay or even it might be dropped by the network devices. This situation leads to network congestion in the network. Network congestion problem consequently affects Quality of Services (QoS) and network performance negatively.

In vehicular networks, congestion management has two main phases: congestion detection and congestion control. Most of the congestion detection is based on the predefined thresholds for channel usage level or number of vehicles. Regarding the literature, researchers significantly made efforts for the congestion control phase and for the congestion detection phase they used assumptions (e.g., when more than 70% of the communication channel is used or when data generation rate exceeds a predefined value). However, finding the optimal value for the threshold still is a challenge since it may have a risk of channel under-utilization. Moreover, using a high amount of threshold for congestion detection could increase packet loss in a highly dense vehicular environment.

In the literature, the proposed controlling network congestion methods are centralized or distributed. In a centralized method, a central unit is responsible for making controlling decisions and vehicles must obey the commands of the central unit. In distributed methods, the vehicles decide how to control the network congestion, mostly based on the information receives from other vehicles around. Monitoring, data gathering and data analyzing required robust enough computation resources and storage. Otherwise, high overheads and delay are inevitable. Nevertheless, distributed methods are widely considered in the literature. However, emerging recent technologies such as Software Defined Network (SDN), fog computing and network virtualization which can provide programmability, global view and suitable computing, networking and storage resources encourage the authors to consider centralized methods more than before.

In this dissertation, the first phase of congestion management is considered but in a different

way. Artificial Intelligence (AI) methods are considered to generate congestion prediction methods for HetVNET. The prediction problem is defined in both styles of classification problem and regression problem. Simulated data extracted from Simulation of Urban Mobility (SUMO) and Veins LTE simulator is used as input of the supervised Machine Learning (ML) methods.

The warning and non-warning classes are defined using Data Delivery Ratio (DDR) and Received Signal Strength (RSS). Then Naive Bayes algorithm is applied to predict congestion warning/non-warning states of the HetVNET. Moreover, a centralized and dynamic cloudy-fog architecture is proposed. This architecture has two main components: a Centralized Management Unit (CMU) and Fog Congestion Predictor Units (FCPUs). The proposed Naive Bayes congestion warning classification method can be applied in the FCPU. Additionally, a Centralized Network Congestion Classification (CNCC) mechanism is proposed to show how the proposed classification approach is effective to avoid network congestion in HetVNET. The results show that the proposed method could improve performance of the network in terms of packet loss ratio, average delay and average throughput.

Moreover, a utility function is proposed in this dissertation. In this step, the aim of the regression prediction method is to predict the value of the utility function. A Multiple Linear Regression (MLR) prediction model and a Generalized Regression Neural Network (GRNN) prediction method are proposed to predict the value of the utility function. Predicted value provides a vision about the future and can be used as a clue for congestion network management in making decisions and running policies. This approach helps create intelligent congestion management for an adaptive and autonomous HetVNET. Additionally, the Intelligent Congestion Avoidance Mechanism (ICAM) which is an adaptive transmission power technique that uses GRNN prediction method has been proposed. Obtained results show that GRNN is an accurate, reliable and stable method. Moreover, simulation results show that ICAM could improve network performance in terms of packet loss ratio and average delay.

Regarding the Network Slicing concept in 5g and beyond, a Conditional Generative Adversarial Network (CGAN) method is applied to generate network slices in HetVNET. In this part of the dissertation, a hybrid architecture is proposed using SDN and Network Function Virtualization (NFV). The aim of this part of the dissertation is to dynamically generate configurations for network slices which have the low potential of occurring network congestion in them.

The proposed AI-based methods are trained and tested using the Python programming language. Then accuracy and reliability of the proposed methods were evaluated. The obtained results show that the AI-based proposed methods (Naive Bayes, GRNN and CGAN) could accurately perform their tasks. Moreover, the simulation results show that applying the proposed CNCC and ICAM could improve network performance.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ACRONYMS

AC	Access Category
ACK	Acknowledgement
AI	Artificial Intelligence
AIFS	Arbitration Interframe Spacing
ANN	Artificial Neural Network
API	Application Programming Interface
AUC	Area Under the Curve
AWS	Amazon Web Services
BBR	Bottleneck Bandwidth and Round-trip propagation time
BS	Base Station
BSM	Basic Safety Message
CABS	Context Awareness Beacon Scheduling
CACC	Channel Aware Congestion Control
CAM	Cooperative Awareness Messages
CART	Classification and Regression Tree
CBR	Channel Busy Ratio
CCH	Control Channel
CCID	Congestion Control Identifier
CE	Congestion Encounter
CGAN	Conditional Generative Adversarial Network
CGAN-SDN	CGAN-Software-Defined Network
CMU	Centralized Management Unit
CNCC	Centralized Network Congestion Classification
CNN	Convolutional Neural Network
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CW	Contention Window
C-ITS	Cooperative Intelligent Transportation System
DCC	Decentralized Congestion Control
DCCP	Datagram Congestion Control Protocol
DDR	Data Delivery Ratio
DENM	Decentralized Environmental Notification Messages
DNN	Deep Neural Network
DH	Density Histogram

DnyB	Dynamic Beaconing
DSRC	Dedicated Short Range Communication
DTR	Decision Tree Regression
D-FPAV	Distributed-Fair Power Adjustment for Vehicular environment
EC	Congestion Encounter
ECN	Explicit Congestion Notification
ECT	ECN Capable Transport
EDCA	Enhanced Distributed Channel Access
EDF	Earlier Deadline First
ERD	Random Early Detection
ETSI	European Telecommunications Standards Institute
FCPU	Fog Congestion Predictor Unit
FD2C	Fully Distributed Congestion Control
FIFO	First-In First-Out
FN	False Negative
FP	False Positive
GAN	Generative Adversarial Network
GCP	Google Cloud Platform
GIS	Geographic Information System
GPS	Global Position System
GRNN	Generalized Regression Neural Network
HALL	High Availability Low Latency
HetVNET	Heterogeneous Vehicular Network
HPLR	High Power Long Range
ICAM	Intelligent Congestion Avoidance Mechanism
IoE	Intelligent offloading Engine
IoT	Internet of Things
IoV	Internet of Vehicle
IP	Internet Protocol
ISP	Internet Service Provider
ITS	Intelligent Transportation System
KNN	K Nearest Neighbor
LIMERIC	Linear Message Rate Integrated Control
LSTM	Long Short-Term Memory
LTE	Long-Term Evolution
MAC	Medium Access Control

MBL	Maximum Beaconing Load
MDR	Message Delivery Ratio
MD-DCC	Message-rate and Data-rate Congestion Control
ML	Machine Learning
MLR	Multiple Linear Regression
ML-CC	Machine Learning Congestion Control
MOTabu	Multi-Objective Tabu Search
MSE	Mean Square Error
MSS	Maximum Segment Size
NFV	Network Functions Virtualization
NGMN	Next Generation Mobile Network Alliance
OSM	OpenStreetMap
QoE	Quality of Experience
QoS	Quality of Service
RAN	Radio Access Network
RAT	Radio Access Technology
RED	Random Early Detection
RF	Random Forest
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristics
RSE	Residual Standard Error
RSS	Received Signal Strength
RoS	Requirement of Safety
RSU	Road Side Unit
RTT	Round-Trip Time
SBCC	Statistical Beaconing Congestion Control
SCH	Service Channel
SDN	Software-Defined Network
SLA	Service Level Agreement
SPCR	Synchronized Persistent Coded Repetition
SINR	Signal to Interference plus Noise Ratio
SPCR	Synchronized Persistent Coded Repetition
SPFS	Shared Proportional Fairness Scheme
SUMO	Simulation of Urban Mobility
SVM	Support Vector Machine

TCP	Transmission Control Protocol
TCP Sack	TCP Selective Acknowledgments
TFRC	TCP Friendly Rate Control
TN	True Negative
TNR	True Negative Rate
TP	True Positive
TPR	True Positive Rate
UM2D	Unicast Multi-hop Data Dissemination
UOTabu	Uni-Objective Tabu Search
VANET	Vehicular Ad hoc Network
VNP	Virtual Network Provider
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
V2X	Vehicle to every thing
WAVE	Wireless Access in Vehicular Environments
WHO	World Health Organization

LIST OF APPENDICES

CHAPTER 1 INTRODUCTION

In vehicular networks, connected vehicles can communicate with other devices via various wireless technologies. Providing safe, efficient and comfortable experiences for road travelers are the main objective to equip vehicles in order to initiate and use wireless communications. Beside all of the advantages that vehicular networks have for the users, occurring congestion in the network is a significant concern for researchers. Increasing number of vehicular users and consequently generating huge data in the network, make the communication resources saturated. This situation plummets the network performance and Quality of Service (QoS) by significantly increasing packet loss and latency in the network.

Congestion detection mechanism and congestion control mechanism are the main strategies to solve congestion problem in vehicular networks [1]. In closed-loop congestion control mechanisms, initially the congestion occurs and then it is detected and controlled in the network. However, in open-loop congestion controlling mechanisms, the measurements and strategies run in order to cope with congestion problem before it occurs in the network [1]. Controlling the transmission rate and controlling the transmission power are the key parts of network congestion solutions [1].

Network dynamicity, fast and huge data generation and strict and diverse service requirements are the characteristics of the vehicular networks. Regarding these features, intelligent based congestion management strategies could be an approach to meet the dynamic and various requirements of the vehicular networks. Although significant progress has been made by researchers to regrade congestion problem in vehicular networks, insufficient intelligent congestion management strategies (congestion prediction and congestion control) is sensible in the literature [1].

1.1 Definitions and Basic Concepts

1.1.1 Heterogeneous Vehicular Network (HetVNET)

Architecture of HetVNET: Communications, Components and Scenarios

Dedicated Short Range Communication (DSRC) and Long-Term Evolution (LTE) are two technologies that are used in Heterogeneous Vehicular Network (HetVNET). DSRC is used for Vehicle-to-Vehicle (V2V) communication and LTE is assigned for Vehicle-to-Infrastructure (V2I) Communication. These communications are shown in Fig. 1.1.

As shown in Fig. 1.1, Radio Access Network (RAN), Core Network and Service Center are the main components of HetVNET [2]. Authentication, switching, aggregation and many more are the functions in core the network. Various services are provided to the users by service providers and through the service center.

Fig. 1.1 illustrates the architecture of HetVNET considering two types of scenarios: urban scenario and expressway scenario. Each of the scenarios has their own characteristics such as the vehicles speed, the vehicles maneuvers, road capacity in terms of the maximum number of vehicles on the road, barriers, stop points and etc..

As Fig. 1.1 shows, when an accident occurs in the road, a crash warning safety message is broadcast to other vehicles in the path via V2V and V2I connections. This kind of safety message is generated by the safety applications. Safety services and non-safety services and their requirements are explained in the next subsection.

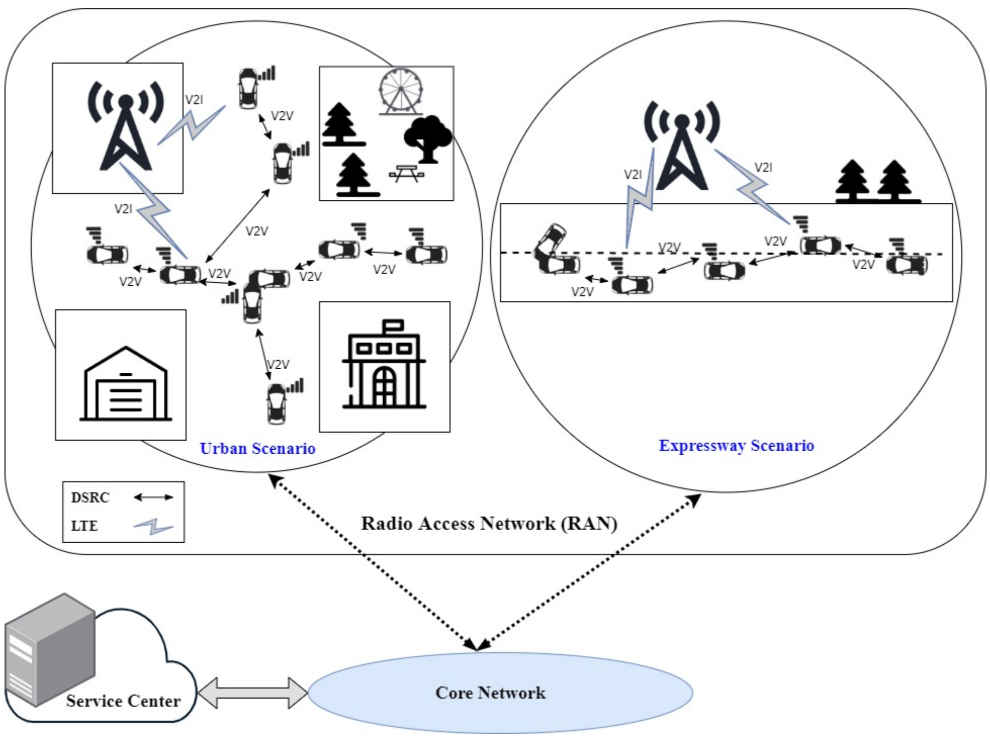

Figure 1.1 HetVNET architecture with showing urban and expressway scenarios.

Applications and Service Requirements in HetVNET

Safety and non-safety services are provided for vehicular users by various applications. The mentioned service types have their own requirements for various user cases [2]. These requirements and user cases are listed in the Table 1.1 and Table 1.2.

Safety services help users to have safer travels with a very low risk of accident, fatality and injury. As Table 1.1 shows, the safety services provide warnings for several specific cases such as the car accident, collision and road hazards. Network reliability is highly demanded for safety services. Since the safety messages should be transmitted in a specific time and before they expire and be useless, timing has significant importance in the safety services. As mentioned in Table 1.1, the safety services are time sensitive and maximum latency time is 100 ms for most of them.

Table 1.1 Safety services and their requirements [2]

Safety Services	User Case	Usage	Communication Mode	Security/ Reliability Requirements	Maximum Latency
Vehicle status warning	Emergency electronic brake lights	Warn a sudden slowdown of the following vehicle	Time limited periodic broadcast on event	High/High	100 ms
	Abnormal condition warning	Warn the abnormal vehicle state	Time limited periodic broadcast on event	High/High	100 ms
Vehicle type warning	Emergency vehicle warnings	Reduce emergency vehicle's intervention time	Periodic triggered by vehicle mode	High/High	100 ms
	Slow vehicle warning	Improve the traffic fluidity	Periodic triggered by vehicle mode	High/High	100 ms
	Motorcycle warning	Collision avoidance	V2X co-operative awareness	High/High	100 ms
	Vulnerable road user warning	Collision avoidance	V2X co-operative awareness	High/High	100 ms
Traffic hazard warning	Wrong way driving warning	Wrong way driving warning	Time limited periodic broadcast on event	High/High	100 ms
	Stationary vehicle warning	Avoid succession of collisions	Time limited periodic broadcast on event	High/High	100 ms
	Traffic condition warning	Reduce the risk of longitudinal collision on traffic jam forming	Time limited periodic broadcast/authoritative message triggered	High/High	100 ms
	Signal violation warning	Reduce the risk of stop/traffic violation	Temporary messages broadcasting on event	High/High	100 ms
	Roadwork warning	Reduce the risk of accident at the level of roadwork	Temporary messages broadcasting on event	High/High	100 ms
	Decentralized floating car data	Improve safety and traffic	Time limited periodic broadcasting on event	High/High	100 ms
Dynamic vehicle warning	Overtaking vehicle warnings	Reduce the risk of accident	V2X co-operative awareness	High/High	100 ms
	Lane change assistance	Active road safety	V2X co-operative awareness	High/High	100 ms
	Pre-crash sensing warning	Accident impact mitigation	Broadcast of pre-crash state	High/High	50 ms
	Co-operative glare reduction	Avoid the frontal collision	V2X co-operative awareness	High/High	100 ms

For example, in Table 1.1, time for pre-crash sensing warning is very tight; it has a maximum latency of 50 ms. Therefore, any difficulty in data flow (such as congestion) could jeopardize the performance of safety application and consequently involves a serious risk for human life.

Non-safety services are provided by infotainment applications. The aim of these applications is to make travel pleasant for the drivers and passengers of vehicles. Media streaming, web browsing and map downloading are examples of services that are provided by non-safety applications. As Table 1.2 shows, most of these services are less time sensitive than safety services. Maximum latency is 500 ms in most of the non-safety services.

Table 1.2 Non-safety services and their requirements [2]

Non-safety Services	User Case	Usage	Communication Mode	Security/ Reliability Requirements	Maximum Latency
Traffic management	Regulatory/ contextual speed limits	Enhance the traffic efficiency/reduce the vehicle's pollution	Time limited periodic broadcast on event	High/High	N/A
	Traffic light optimal speed advisory	Traffic regulation at an intersection	Periodic, permanent message broadcasting	High/High	100 ms
	Intersection management	Road safety and traffic regulation at an intersection	Periodic, permanent message broadcasting	High/High	100 ms
	Co-operative flexible lane change	Enhancement of mobility efficiency	Periodic broadcasting messages	High/High	500 ms
	Electronic toll collect	Traffic fluidity at the toll collection	I2V broadcasting and uni-cast full duplex session	High/High	500 ms
Infotainment	Point of interest notification	Driver and passengers comfort	Periodic broadcasting messages	Medium/Medium	500 ms
	Local electronic commerce	Vehicle driver/passenger comfort	Duplex communication between RSU and vehicles	High/High	500 ms
	Media download	Passenger entertainment	User access to Internet for multimedia download	Medium/Medium	500 ms
	Map download and update	Efficiency and comfort	Access to Internet for map download and update	Medium/Medium	500 ms

1.1.2 Software-Defined Network (SDN)

Notion of detaching control tasks from physical devices in network infrastructure and defining the network control duties to the central control plane, has been emerged by SDN technology [3–5]. SDN has three layers and these layers are interacted by Application Programming Interface (API) [4]. As Fig. 1.2 shows, two types of API are using in SDN architecture:

- Northbound API: to create connection between application layer and control plane [4];
- Southbound API: is used for interconnection between data plane and control plane [4].

SDN provides programmability, agility, and flexibility to the network, also it brings an efficient, simple and advanced control to the network [3].

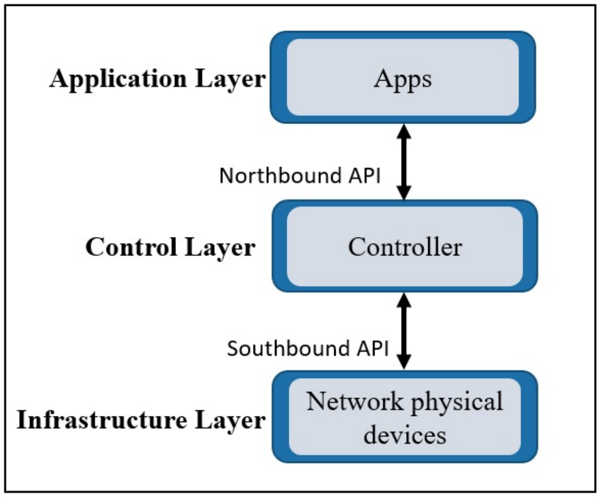

Figure 1.2 SDN architecture

1.1.3 Fog Computing

Equipping devices for using the Internet gives the opportunity of inter connectivity to devices in order to generate, collect, and exchange data with each other. In the Internet of Things (IoT) paradigm, these devices have not sufficient computation or storage resources. Therefore, these devices need to communicate with the cloud to carry out several complex computational tasks. Because of security, and long geographical distance between IoT devices and cloud, it is efficient to have a local object like fog with enough computation resources and close to IoT devices [6]. In fog computing technology, a fog device has the ability of storage, compute, and processing data in order to do some kind of data computations locally and geography close to the user [7, 8]. Indeed, fog is placed between cloud and end user in order to efficiently enhance QoS with providing services with lower latency and disturbance [7]. Using fog computing supports mobility, location awareness, and real-time interaction, it can improve QoS and reduce latency in network [6, 8, 9].

1.1.4 Network Slicing

Network slicing is defined using three layers: Service instant layer, Network slice instant layer and resource layer. Each network slice instant has features that are necessary in order to meet requirements of the corresponding Service instants. A network operator needs the blueprint of a network slice instant to create it. Two or more service instants can use a common network slice instant. A network slice instant can have no or one and more sub-network instants. Fig. 1.3 shows three layers of network slicing concept [10].

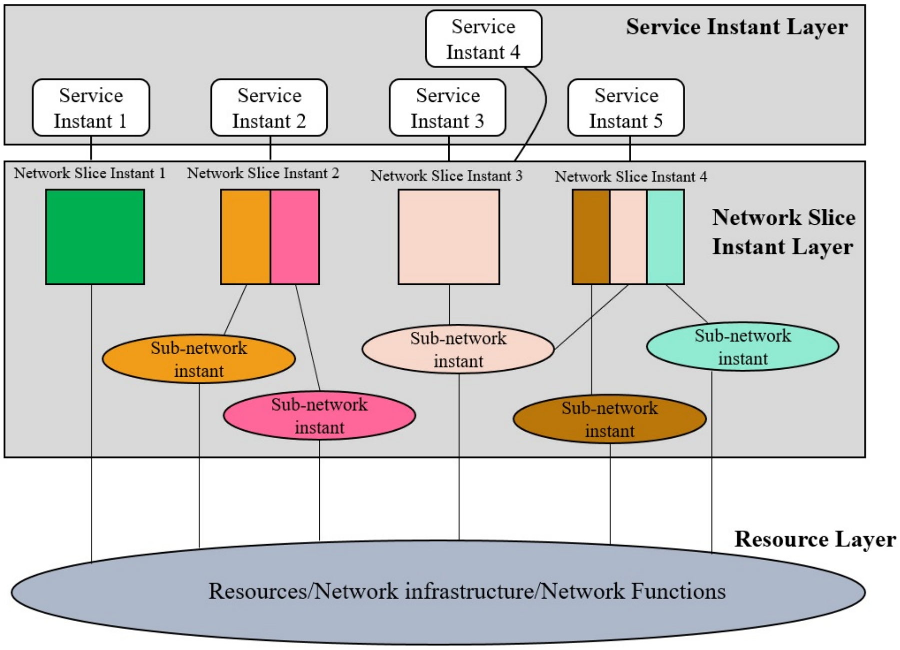

Figure 1.3 The three layers of network slicing concept, proposed by NGMN [10]

1.2 Congestion Problem

The condition that demands for network resources become more than network resource capacity is called network congestion. Network resources can be the link's bandwidth, memory, and devices buffer (e.g. buffer of network router device). Based on the rate of increasing demands for network resources and finite network resources capacities, the network may suffer from congestion situations for a long time or transitory.

When data transmission rate by source nodes is more than network bandwidth capacity then network devices will be overflowed by excessive data load. Therefore, to avoid congestion in the network, the sender node must not send data over bandwidth capacity. Moreover, transmission time in bottleneck is lower than other links in the network.

Data load, which is generated by source nodes and flows towards receiver nodes, is input traffic. Data traffic, which network links can handle (based on link bandwidth capacity) is output traffic. If huge data traffic needs to go through several routers with fixed buffer capacity and via finite bandwidth links to reach the destination node, then as Fig. 1.4 illustrates we will have two states including desired state and congestion state in this scenario.

As long as the ratio of output traffic over input traffic equals one, which indicates that input traffic and output traffic are identical, then we have the desired situation in the network.

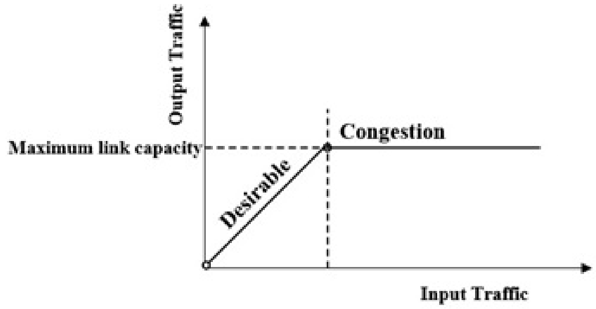

Figure 1.4 Desired state and congestion state of network

Because it shows that data is smoothly flowing in the network. In the desired condition, amount of packet loss and delay are minimum, network throughput is high, QoS is at acceptable level, and consequently user satisfaction from network performance is at excellence level. However, by collapsing the ratio of output traffic over input traffic to less than one, congestion situations gradually will appear in the network.

In the scenario that input data traffic load exceeds output link bandwidth capacity, packets must be waited in the router's buffer to pass through the router and go towards destination nodes. If we use a router with unlimited buffer size in the network, then we will have a long queue of arrived packets in the buffer and destination nodes must wait a long time for receiving packets. Also, in the same scenario a router with a fixed buffer capacity in the network, will discard the incoming packets whenever its buffer gets full. So, as long as the buffer remains saturated, the number of dropped packets is increased. Regarding the network reliability, under this network situation, which source nodes will not receive any acknowledgement about receiving packets from destination nodes, they re-transmit the packets again and it leads to heavy load in the network, which makes network congestion worse. Besides, by dropping packets, all the resources that are used for reaching discarded packets to the congested point in the network are wasted. Thus, congestion in the network leads to the reduction in network throughput, increase in both delay and packet loss, growth in number of packet copies, and waste network resource capacities. Regarding the importance of network congestion problem, it has been considered in many scientific works [11–17] and several congestion control protocols are provided by researchers.

1.3 Open Problems

Lack of centralized approaches

There is a lack in providing efficient reliable congestion control solutions that concede calculations related to the congestion avoidance and control at the infrastructure level instead of vehicles like [18–22] . In [20], results show good efficiency, high packet delivery and low channel busy ratio. However, vehicles must execute too much computations using information of each received beacon from vehicles around. Also, the calculations that are needed to find closest and furthest ahead and behind vehicles, must be done in a limited time. Therefore, having time restriction for running several computations is a challenging problem for the proposed method, because beacon's information will expire after each 100 milliseconds. In [22], all the calculations (specially for predicting value of utility function which use Markov chain method) need computation resources and also are time restricted for vehicles. Since information is changed dynamically and quickly in vehicular networks, therefore the calculations must be done before that new update of information received, and these make a big task to do in a short time for vehicle users. In this regard, using SDN technology gives opportunity to the researchers for providing congestion control mechanisms at controllers of SDN architecture and apart from vehicle user level. However, to the best of our knowledge number of research works which proposed SDN based architecture to solve network congestion problem in vehicular networks are very limited currently.

Underestimating the importance of applying congestion prediction

Congestion avoidance is an open-loop problem. Logically, we keep away from situations or events in which we are sure about occurring or they have a high possibility of occurrence. Therefore, using powerful network prediction mechanisms are vital and valuable for saving energy, time, costs and resources.

Need to take the best actions in golden time

Furthermore, latency is another parameter which is very important in real-time situations. In critical conditions like when an accident happens real-time solutions are vital, and decisions must be made in efficient time, therefore latency must be as less as human life saved. An unfavorable result of network congestion is high latency in time sensitive situations. Minimum human reaction time is 500 ms [23], therefore if an emergency message is received with a delay more than 500 ms, then it means that safety applications are useless because of weak network performance. In [24], results show that average delay in some scheduling algorithms like First-In First-Out (FIFO), Enhanced Distributed Channel Access (EDCA), Distributed-Fair Power Adjustment for Vehicular environment(D-FPAV), and Context Awareness Beacon Scheduling (CABS) is more than 500 ms.

Deficient of adaptable methods applicable for HetVNET as a dynamic environment

Besides, the dynamicity of HetVNET makes anticipating a network situation a challenging problem, because mobility and variation is in the nature of vehicular networks. In other words, each fundamental parameter of the vehicular network is free to change each time. Big necessity of having a dynamic model which shows high flexibility in face of dynamic conversions with minimum error in making suitable solutions, exists. A mechanism that learns from the parameter's changing pattern in order to create a robust model which will be accountable in different dynamic changes, is highly needed.

Handling difficulties result from the crucial roles of the clouds in the vehicular networks

In the IoT environment, all data generated by connected devices are sent to the cloud in order to process, analyze, and find solutions [25, 26]. Then the cloud sends solutions to the devices, but sending many solutions from the cloud to a large number of devices needs huge bandwidth [6, 7]. Furthermore, sending the huge volume of data to the cloud that is not geographically close to the vehicles, makes difficulties in security, time latency and reliability, particularly in time-sensitive conditions.

Shortage of intelligent strategies

Absence of intelligent methods in case of congestion prediction and control in vehicular networks is sensible in [22, 27–30]. The methods which use learning algorithms at infrastructure level for solving congestion problems in the vehicular network. In [29], authors proposed an architecture exploited from SDN and concept of edge as a service to solve congestion problems with no intelligence mechanism for congestion prediction.

1.3.1 Open problems Considered in This Dissertation

Congestion problems can impair network performance and even network service providing becomes very slow. This situation for a high dynamic network type like vehicular network is dangerous for human life. For example, in using the autonomous vehicles, which almost all the decisions are made using the information coming from smart surrounding objects and the data generated by the vehicles sensors, any intolerable delay in receiving safety related data may negatively affect human life and costs.

Regarding the importance of congestion problems and its negative consequences on vehicular network behaviour, many researchers proposed solutions to avoid congestion and many others tried to find mechanisms to control it in the network. In closed-loop congestion control mechanisms, the authors proposed methods to recover the vehicular network situation quickly. For example, in many of the proposed congestion control methods, transmission power and transmission rate are reduced to control the congestion state of the vehicular network. In open-loop congestion control methods, the authors proposed approaches to prevent conditions in which the network will experience congestion. For example, they define a threshold for channel usage level and whenever the channel busy level meets the threshold, a mechanism will be executed to alleviate channel loads. The proposed congestion controlling mechanisms are mostly based on decentralized methods. In these methods, vehicles execute the controlling mechanisms and make decisions. However, in centralized approaches congestion controlling strategies are employed in a central system and vehicles must obey the congestion controlling policies decided in the centralized system. Decentralized approaches need high vehicle cooperation, since vehicles need information received from other vehicles to make appropriate decisions. Moreover, data analyzing and carrying out multiple computations create overhead for vehicles. Due to limited computation and storage resources of vehicles, with increasing the number of vehicle overhead and delay may go up a lot. However, centralized methods are easier in implementing, updating and debugging. Besides, applying powerful resources in the central unit can make a boosted congestion management mechanism.

Congestion detection is a main part of network congestion management. However, it is not significantly considered in the literature. In most of the research works, the authors considered several assumptions for this step. In heterogeneous types of networks like HetVNET, when more than one technology is used by users, the assumptions for one technology may not apply to all the used technologies. Therefore, lack of methods in congestion detection is sensible in the literature. Besides, we believe that to cope with congestion problem in heterogeneous networks we need solutions, which consist of congestion prediction instead of congestion detection. Because, when congestion is detected means it exists and now we just discovered it. However, if we want to have a tolerable heterogeneous network, instead of letting congestion occur and then control it, we need to avoid congestion, and this is feasible by predicting network congestion.

Artificial Intelligence (AI) methods are widely used for various prediction problems such as regression and classification problems. Machine learning methods help devices and machines to learn from existing data and utilize what they learned for new data, which the device may have never seen before. Machine Learning algorithms are categorized to supervised and unsupervised learning. An unsupervised learning method is capable of learning and making solutions but with no error evaluation in the decision models. However, a supervised method is an error correction method, and learning will be matured by training set and experiences. Supervised learning methods are suitable in non-linear real time problems such as prediction [31]. Prediction methods are in the supervised category. Using supervised machine learning methods to predict network congestion in HetVNET is an open problem yet.

Moreover, the learning process and making predictions with accuracy have challenges due to massive information needed and highly computation and storage power. However, with recent technologies such as fog computing and SDN, it is not unlikely that researchers in this area will consider network congestion prediction more than before.

In fog computing technology, a fog device has the ability of storage, compute, and processing data in order to do some kinds of data computations locally and geography close to the user [7, 8]. Using fog computing supports mobility, location awareness, and real-time interaction, it can improve QoS and reduce latency in the network [6, 8]. On the other hand, the notion of detaching control tasks from physical devices in network infrastructure and defining the network control duties to the central control plane has been emerged by SDN technology [3–5]. SDN provides programmability, agility, and flexibility to the network. Also, it brings an efficient, simple and advanced control to the network [3]. Therefore, considering recent technologies, proposing intelligent network congestion prediction and avoidance techniques employed in a centralized architecture are considered in this dissertation.

1.3.2 Research Questions

Regarding the aforesaid open challenges, following main research question may arise:

With regard to the influence role of ML methods to create intelligent communication technology, and advantages of applying SDN and NFV technologies in the centralized network management systems, how can we deal with network congestion problem and its disruptive effects in the HetVNET?

More specifically:

- Regarding the high dynamic nature of vehicular networks, how we can propose an algorithm which is able to precisely avoid network congestion using a congestion prediction model?

- Considering advantages of using fog computing, how can we design an architecture, which intelligently and reliably predict and control network congestion in HetVNET?

- If we suppose that 5G is the main communication system, how does the notion of network slicing help us to alleviate congestion problem in HetVNET?

1.4 Research Objectives

The main objective of this dissertation is to propose AI-based congestion prediction and avoidance methods that can be employed in designated hybrid, centralized and intelligent architectures of HetVNET. The three following sub-objectives are considered towards achieving the main objective of this dissertation:

1. Devising a centralized and dynamic Cloud-Fog based Intelligent Congestion Prediction Architecture of HetVNET;

2. Proposing an Intelligent Congestion Avoidance Mechanism (ICAM) using Artificial Neural Networks (ANN);

3. Augmenting data applicable in creating network slices in HetVNET, using Conditional Generative Adversarial Nets (CGAN) and designing a hybrid CGAN-SDN architecture;

4. Evaluating performance of the proposed methods.

1.5 Global Research Methodology

Network management in high dynamic networks such as HetVNET is a challenge. Resource management, mobility management, debugging and failure recovery are to name a few of network problems. In a high dynamic network the solutions for the regular challenges must be generated and executed precisely, quickly and dynamically too. Autonomous networks is the next evolution in network management in which networks can be self-provisioning, self-diagnosing and self-healing. Indeed, these networks can dynamically adapt themselves with the variety of network situations. They monitor the network and analyze the network situation and prepare themselves to solve any probable problem. Recent technologies such as SDN, NFV, fog computing and network slicing along with AI methods can help create autonomous networks. The notion of emerging autonomous networks are based on proposing and applying centralized AI-based mechanisms using the mentioned recent technologies.

In this regard and toward generating autonomous and adaptive networks, it is required to propose the SDN and fog-based architectures that are programmable and have enough computing and storage resources. Regarding the main objective and with considering network congestion in HetVNET, in this dissertation, novel solutions for this problem are proposed with the aim to generate autonomousity in this type of network.

Congestion prediction is a way toward having an adaptive and tolerable network. Since the difficulty in data flow and congestion could be anticipated by machine learning methods, networks can be prepared for preventing or controlling the congestion state in the network. Therefore, the negative consequences of occurring network congestion such as lots of packet loss, high delay and consequently low QoS can be improved or at least remain at a fine level. In this regard, the concept of congestion and happening it in the network is inferred using these red flags like low network throughput and high packet loss. Indeed, network congestion is a state of the network not an independent network parameter which can be modified and set easily. In the other words, congestion can be discovered based on the network behaviour. For example, if there are plenty of lost packets then it infers to a congestion state in the network. On the other hand, network parameters such as the transmission power, the data rate, and the bandwidth have an impact on occurring congestion in the network. In many congestion related works like [32–38], the authors proposed methods to control network congestion by modifying the amount of the transmission power and data rate. Therefore, considering the key role of both network behaviour and network parameters in making congestion prediction models is a worthy approach, since we can avoid congestion based on network behaviour and modifying the value of effectiveness parameters.

Predicting network congestion can be considered as both the regression and the classification problem. In the former, the prediction model anticipates a quantity, which in this work can be a value which shows performance of the network. In the later, prediction models can predict a discrete class level of state of the network in terms of data flow. After predicting by any of these methods, the HetVNET congestion management system is able to execute any controlling methods to prevent congestion in the network. One of the avoidance approaches can be to modify the value of transmission power and data rate which are important in controlling network congestion in vehicular networks.

Data analyzing is the main part of any AI prediction method. In this dissertation, simulated data is used. Indeed, the HetVNET environment is generated using two simulators: the Simulation of Urban Mobility (SUMO) and Veins LTE. Parameters such as number of vehicles, data rate, transmission power of DSRC, Transmission power of LTE and LTE bandwidth are considered as features of data set. These parameters were changed in the simulation scenarios. The data is gathered in a dataset and the dataset is used as an input of the AI predicting algorithm. Therefore, these five parameters are predictors in the prediction models. In other words, the congestion situation in HetVNET can be predicted using the five parameters.

In the first specific objective of this dissertation, congestion is inferred using a utility function which shows performance of network in terms of smooth data flowing. Therefore, a prediction model must predict the value of the utility function. The network throughput and data generation rate are components of the utility function. To reach the first specific objective, regression congestion prediction model is generated. First of all, several regression algorithms were considered such as Multiple linear regression, support vector machine, decision tree regression and Generalized Regression Neural Network (GRNN). Congestion prediction model of the mentioned methods was generated using Python. The algorithms applied a training dataset to generate the prediction model and tested using a test dataset. Therefore, performance of the algorithms are compared in the same condition. Accuracy, reliability and stability of the algorithms have been evaluated. GRNN is an ANN method which employs the Gaussian activation function in the hidden layer. GRNN is a feed forwarding algorithm which need one step training model (no need to back propagation). This feature makes it a fast learner that can converge to an optimal solution in a short time. Moreover, GRNN can provide accurate prediction model without necessarily needing a huge amount of data. A Python library for ANN named NeuPy is applied to generate GRNN congestion prediction model.

After predicting network congestion we need a controlling strategy that prevents from occurring congestion in HetVNET. Therefore, ICAM has two main components: prediction and avoidance mechanisms. As explained, in the prediction step, GRNN method is applied to generate a congestion prediction model. This model is used to predict the value of utility function. Then, based on the result of the GRNN congestion prediction model, an avoidance mechanism is initiated to work. The avoidance mechanism is based on adapting transmission power using the prediction model.

In the second specific objective, packet loss is considered to determine congestion in HetVNET. Congestion is the reason for packet loss in the network, however, it is not the only reason for lost packets. Indeed, packet loss may happen because of weak signal or bad channel condition. In this situation the strength of the received signal is low. Therefore, we considered data delivery ratio along with received signal strength to infer network congestion in HetVNET.

Moreover, in the second specific objective, the congestion prediction problem is defined as a classification problem. Naive Bayesian congestion classification method is applied to generate a network classification algorithm. In addition to the Naive Bayesian method, Support Vector Machine (SVM), K Nearest Neighbor (KNN) and Random Forest classifier, have been considered in this dissertation. Python is used to generate the network congestion prediction methods. A well-known Python library named Scikit-learn is applied to implement the network congestion classification methods. A centralized and intelligent congestion management approach will be proposed to predict network data flowing classes. To attain this objective, data analyzing and required computations for predicting congestion are accomplished in fog devices. Unlike the decentralized methods, the vehicles do not predict the network congestion and they are informed about the result of the prediction and based on the result, they must run the avoidance strategy. The avoidance mechanism is based on adapting the value of data rate and contention window.

Network slicing approach in 5G networks to provide reliable services using finite network resources is another notion to consider in this dissertation. Using a generative neural network method to tailor the HetVNET slices quickly and based on the previous successful experiences is a novel method which can be applied in SDN based architecture of HetVNET. To accomplish the third specific objective, network slices must be created dynamically based on network situation. In this regard, the past successful experiences in terms of smooth data following in the network must be considered to form the new network slices. Regarding dynamicity of the vehicular network and the quick changes in the topology, service requirements and number of users, the network congestion management system should be compatible with the changes. Therefore, an invariable congestion management strategy could not be the answer for all the variant states of such a network. In the unstable network conditions with rapid changes in the network situation, searching for one optimum solution is difficult and far from realism. However, applying slice configurations that are close to the previous successful experiences could be a potential solution.

In this regard, we can augment information of successful experiences using the CGAN method. The CGAN is based on a min-max game and trains during backpropagation and using feedback. CGAN is a deep learning model, and we need a Python library such as Keras to implement it. The task of CGAN is to generate data similar to real data from the noise. This CGAN method can apply in the controller of SDN to generate new network slices. Then, based on the blueprints created in the controller, various slices are formed with different numbers of vehicles in the infrastructure layer. To boost the controller to accomplish the CGAN's tasks, we can employ fog devices in the control layer of SDN.

The metrics used to evaluate performance of the regression and classification prediction models are different. In the regression prediction problems, Root Mean Square Error (RMSE) shows how much the predicted values vary from actual values. A model with high RMSE is more accurate than that with a low RMSE. Coefficient of determination (R^2) indicates how much variability in the dependent variable is predictable using the predictors. A value close to one shows that the significant amount of variations of dependent variable can be explained by predictors, and it is an advantage for the prediction model. F-statistics shows the link between the predictors and the response (dependent variable). Indeed, we need a large amount of F-statistic to prove that at least one predictor must be related to the response. Therefore, in this dissertation the mentioned parameters are used to evaluate performance of the regression prediction methods.

Regarding the classification problems, metrics such as recall, precision, F1 score and accuracy are typically used to measure the performance of the prediction methods. Recall shows that how many times could the prediction model accurately predict the actual positive data. Precision shows among all the positive predictions how many of them are truly predicted. F1 score combines recall and precision into one parameter. F1 score is typically an important metric in problems where the cost of recall and precision are the same. Accuracy shows how much the prediction results are accurate. In the second specific objective of this dissertation, recall, precision, F1 score and accuracy are used to evaluate performance of the classifiers.

Performance of the CGANs are evaluated by measuring accuracy and loss of the discriminator. In a CGAN, the discriminator is responsible to recognize real data from fake data. The generator trains to create data similar to the real data to deceive the discriminator. Indeed, the discriminator and the generator are players of a min-max game. The best result achieved when the discriminator labeled the data with accuracy around 50%. This means that the discriminator finds the real data from fake data, randomly. Therefore, in the third specific objective of this dissertation, accuracy and loss of the discriminator are measured to evaluate performance of the proposed CGAN model. Since CGAN is a deep learning method, the evaluations are based on implementing the proposed CGAN with different numbers of hidden layers and various batch sizes. Moreover, required training time corresponding to the number of hidden layers and batch sizes are considered. Then the CGAN with the best performance is selected as a final model.

To evaluate performance of the proposed approach in HetVNET various metrics such as packet delivery ratio, packet loss ratio, average throughput and average delay are considered. The comparisons between the proposed methods in this dissertation and the other congestion controlling methods in the literature are made using the mentioned metrics.

1.6 Research Contributions

Regarding the notion of autonomous and adaptive networks, centralized and intelligent methods to predict and avoid network congestion in the HetVNET are proposed in this dissertation. These methods enabled the HetVNET to self-control the network congestion using AI techniques along with employing recent technologies such as SDN, fog computing, network virtualization and network slicing. Additionally, the contributions of this dissertation are as follows:

1. **Proposing a centralized congestion classification approach to predict congestion warning states of the HetVNET:** The network congestion states of HetVNET can be classified to warning and nonwarning states, using the amounts of delivered data and signal strength. By integrating these two parameters we can be sure that the packet loss is due to congestion in the network. The five predictors that are used are the parameters that have significant effects on network congestion. Moreover, a cloudy-fog intelligent congestion prediction architecture of HetVNET is proposed to apply the congestion prediction method in the fog devices. The centralized congestion classification approach could improve performance of the network in terms of packet loss ratio, average delay and average throughput.

2. **Developing an intelligent congestion avoidance technique for HetVNET:** A utility function is proposed to show performance of the HetVNET and the GRNN is applied to predict the value of the utility function. Then, based on the sensitivity of the network and roads, the predicted value can be inferred as a safe or a warning or a congestion state in the network. Afterward, the proposed avoidance mechanism modifies the value of transmission power and data rate to avoid congestion in the HetVNET. Therefore, the proposed congestion avoidance mechanism contains two steps: predicting network congestion state (the value of utility function), and adapting the value of transmission power and data rate accordingly. The proposed GRNN congestion prediction model is an accurate, reliable and stable model. Applying the proposed intelligent congestion avoidance technique could improve packet loss ratio and average delay in HetVNET.

3. **Applying the proposed mechanisms increases stability in the performance of the high dynamic HetVNET:** This is the main contribution of this dissertation. By applying the proposed methods, network prepares for the conditions that are likely to accrue in the future. Therefore, with this self-preparation against network congestion situations in advance, network performance should be more stable than before. Because

occurring congestion is not an unexpected event any more. The results show that the network could perfectly prepare itself in a manner to avoid future network congestion in the dense traffic situation. Indeed, this stability is the result of applying a self-adaptive network.

4. **Augmenting data comes from the past successful network experiences to create HetVNET slices:** CGAN is used in a SDN-based architecture of HetVNET with the aim of avoiding congestion in HetVNET slices. The proposed deep learning method is presented in a fog-controller of SDN-based architecture to create network slices intelligently and dynamically. In this method, network slices are categorized by number of the vehicles. Then, the information to configure the network slices are generated by the CGAN. In other words, the network slices are dynamically generated using the value of parameters generated by CGAN. To the best of our knowledge, it is the first time that the CGAN is applied in a SDN-NFV architecture of HetVNET to generate network slices. In the proposed method, the CGAN could accurately augment the data required to create network slices.

1.7 Outline of Dissertation

This dissertation contains eight chapters. Related works are reviewed in Chapter 2. In this chapter standards and congestion control mechanisms are presented and discussed.

In Chapter 3, a Naive Bayes prediction method is proposed to predict the warning state of data flow in the HetVNET. Indeed, the congestion warning prediction is proposed as a classification problem. Moreover, a centralized cloudy-fog based architecture is proposed in which the proposed Naive Bayes prediction method can be applied.

In Chapters 4 and 5, the states of the network are defined and predicting the state of the network in terms of congestion is proposed as a regression problem. A utility function is proposed to show performance of the network. In Chapter 4, a multiple linear regression method is proposed to make a congestion prediction model. Then, in Chapter 5, a neural network method is applied to predict the value of the utility function. Moreover, a congestion avoiding mechanism based on an adaptive transmission power method is presented in this chapter.

In Chapter 6, the network slicing concept in a highly dynamic environment like HetVNET is considered. A deep learning algorithm is applied to augment data from the most successful past experiences. The generated data can be used to configure the network slices dynamically. Moreover, an intelligent hybrid CGAN-SDN architecture is proposed for HetVNET.

In Chapter 7, a general discussion is presented about the proposed methods in this dissertation. Finally, in Chapter 8 the contributions, the limitations and future works are presented.

CHAPTER 2 LITERATURE REVIEW

2.1 Network Congestion Control Mechanisms

Regarding the importance of network congestion problem, it has been considered in many scientific works [11–17] and several network congestion control protocols are provided by researchers.

Transmission Control Protocol (TCP) contains two phases of slow start and congestion avoidance. In the slow start phase, it uses a variable like congestion window (CW) to show network capacity, and gives an initial value to the CW. Mostly, CW gives a value of 1 MSS (Maximum Segment Size) initially. TCP exponentially increases value of CW by 1 for each received acknowledgment during a Round-Trip Time (RTT), for example in first RTT sender sends 1 segment, for second RTT it sends 2 segments, and for next RTT sender sends 4 segments and so on, till the value exceeds a predefined threshold or sender node finds out a segment is lost. Then in order to avoid congestion in the network, the value of CW will be increased slowly by 1 for each acknowledgment that is received during a RTT, also the sender node reduces the volume of data which it wants to send via TCP connection [39]. In TCP Tahoe when CW exceed a predefined slow start threshold "ssthresh" or when an acknowledgement is not received (it means segment is lost so it is signal of congestion in network), then value of "ssthresh" is changed to the half of CW and value of CW is reset to one and start to re-transmit loss segment [11]. After that, CW is increased for each acknowledgement that is received by the source node. TCP Reno uses two mechanisms for solving congestion problem in the network, Fast Retransmit and Fast Recovery. In fast retransmit, if the sender node receives three duplicate acknowledgments, then it supposes that congestion has occurred in the network, it immediately retransmits the segment again and reduces the number of segments, which are going to be sent. Then, TCP Reno will start fast recovery, hence firstly it reduces value of slow start threshold "ssthresh" to the half of CW and secondly updates CW to the new value of "ssthresh" and finally for each received duplicate acknowledgement it increases CW by 1 [11].

TCP with 'Selective Acknowledgments' (TCP Sack) is based on a notion that acknowledgement can be generated for a group of data segments (instead of an acknowledgement for a segment) that are received by a destination node. Indeed, in the TCP Sack, the receiver node produces an acknowledgement that shows which segment has been received. Whenever, the sender node discovers a segment loss then it retransmits the segment and same to the TCP Reno, fast recovery phase will be started [11]. TCP Vegas is based on congestion avoidance.

Indeed, source node controls RTT to inform about load level in buffers of intermediate routers between itself and destination node. If the source node receives a packet acknowledgement with delay it means that the communication link may have been congested. In TCP Vegas, expected transmission rate for a packet is computed. Expected transmission rate equals the ratio of CW over minimum of all RTTs. Then, the source node calculates actual sending rate and compares this value with expected transmission rate. If expected transmission rate equals to a predefined value like α plus actual sending rate then CW will be linearly increased. If expected transmission rate equals to a predefined value like β plus actual sending rate, then CW will be linearly reduced. Otherwise, the value of CW will not be changed [11].

Random Early Detection (RED) congestion control protocol helps source node to reduce transmission rate before that router buffer becomes full and packets are dropped. RED defined two variables of maximum threshold and minimum threshold for queue length in the buffer. It also computes average queue length. If average queue length is less than minimum threshold then packet will be buffered in router. If average queue length is more than minimum threshold and also less than maximum threshold, then packet may be dropped with a probability like P. Otherwise (average queue length is more than maximum threshold), then packet will be dropped certainly [12]. Therefore, if the probability of dropping packets (P) in the buffer is high, the sender should reduce packet transmission rate.

Another queuing management algorithm is Blue, which calculates the probability of dropping a packet (P) from a queue in the router's buffer. Each time that buffer gets full, Blue increases P and if data traffic in the buffer is alleviated, then probability of dropping a packet will be reduced [13]. Most of the mentioned congestion control algorithms are based on data loss in the network, which is considered that packet/segment is lost because of limited buffer capacity of middle nodes like routers. Indeed, these algorithms use packet loss as a signal of congestion and as soon as a packet loss is recognized they switch to the congestion repair/avoidance phase. However, packet loss may happen because a momentary traffic bursts or it may come after applying a security policy at intermediate nodes between source node and destination node. In this regard, Explicit Congestion Notification (ECN) and Bottleneck Bandwidth and Round-trip propagation time (BBR) are two congestion control protocols, which use another parameter except packet loss for congestion detection and control in the network.

ECN provides a congestion control mechanism as an extension of TCP and Internet Protocol (IP). In ECN two end nodes inform each other about congestion by setting an ECN code point in two bits of IP header in each packet. Code points can have four values, each indicating a state of the network in terms of congestion. Following table contains possible code points and corresponding meaning of each ones [14]:

Table 2.1 ECN code points and descriptions[14]

Cod Points	Descriptions
(00)	Non ECN (a packet with no use of ECN)
(10)	ECN Capable Transport and called ECT(0)
(01)	ECN Capable Transport and called ECT(1)
(11)	Congestion Encounter (CE)

In the case that each of the end nodes do not support ECN they can use (00) as code point in the header of the packet. ECT (0) and ECT (1) have the same meaning and nodes can use them for the condition that transferring data has no difficulty in terms of congestion. Congestion Encounter (CE) is used by routers with overloaded buffer. Actually, the router (which is encountered by congestion problem and its buffer capacity is full) uses CE code point in the header of the packets which are coming after overloading buffer state instead of dropping these packets. In this case, if an end-node receives a packet with CE at the header, it informs that there is a traffic burst at the router point, thus the node reduces its transmission rate [14].

Moreover, BBR that has been proposed by Google recently is not a packet loss based congestion protocol [15]. In BBR, bandwidth and data transmission rate is considered as two main parameters to cope with congestion in the network. After sending data from source node to the destination node, sender node can estimate actual bandwidth between itself and receiver node by computing total delivered data over a time interval. For the next time, the sender node takes risk and increases the amount of data to send in order to know if network bandwidth still can tolerate this volume of data or cannot. Again from calculating delivered data over a time interval, source node decides to prepare more data to send or cease raising it. If the ratio of delivered data over a time interval is reduced in comparison to the previous time interval, then the source node decides to diminish the volume of data to send, otherwise it continues to increase the amount of data for transmission at the next time of sending data. Reduction in data prepared to transmit may not be a stable state for the source node and it is possible that the sender node again decides to increase the volume of data to transmit in order to find out if the congestion was a transient problem in the network or not. Once the ratio of delivered data over a next time interval is improved in comparison with previous ones, then again the source node can consider more data to transmit [15]. In BBR sender nodes perform estimations based on the network throughput and make decisions according to the network performance. Indeed, congestion situations are inferred regarding data delivery level during specific times in the network.

In case of congestion problems, Datagram Congestion Control Protocol (DCCP) provides a way that two end points can cooperatively decide about which congestion control is used [16]. DCCP uses ECN code point of CE, which is used when congestion is experienced by a router. Indeed, several kinds of dropped packets (which are explained at the following) or ECN code point (CE) are considered as signals of congestion in the network. One byte in Congestion Control Identifier (CCID) is used for the congestion control mechanisms negotiation between source node and destination node. CCID can have a positive integer value in the interval of [0, 255], which except CCID 2 and CCID 3 other CCIDs are reserved. The CCID 2 is about applying a congestion control mechanism, which is similar to the TCP congestion protocol. It means that, for each delivered packet receiver produces an acknowledgment for the transmitter node. A CW is defined for the sender. This node sends packets until reaching the CW value. The strategy for coping with congestion is to cut down CW to half of the current value. Besides, the receiver node generates an acknowledgment containing a sequence number of received packets and sends it to the sender node. In CCID 3, which is similar to the TFRC congestion control protocol, the receiver node produces an estimation about packet loss for the transmitter node. Transmit rate at the sender side is updated using the estimation. DCCP uses a code to define the reason for dropping packets and the sender node informs about why a packet is dropped. Table 2.2 contains drop codes and meaning of each ones [16]:

Table 2.2 Drop codes and descriptions[16]

Drop Code	**Descriptions**
0	Protocol restrictions
1	Application is not listening
2	Overloaded traffic in buffer
3	Corrupt data in packet
4-6	Reserved
7	Delivered corrupt data packet at receiver node

When a sender node receives an acknowledgment with drop code 0, it informs that packet is discarded due to protocol limitation and not for congestion problem. Therefore, the sender must cease sending packets until the restriction is resolved. If a packet is dropped with drop code 1, it means that the application at the receiver node stopped listening for data, so it is not a congestion situation and the sender node should not transmit data any more. Drop code 2 must be considered a congestion situation by the sender node and the transmitter should react appropriately to alleviate congestion in the network. Moreover, using drop codes 3, 4-6, and 7 do not indicate congestion problem in the network, except they come with ECN code point (CE) [16]. As explained before, in the DCCP, congestion is recognized with one of the two conditions: first, dropped packet and second, ECN code point. It means, if each one of the two conditions is met then end points must react based on congestion control strategies (CCID 2 or CCID 3). Indeed, all the dropping packet types are not considered as the congestion situation except packet dropping happens along with ECN code point [16]. The main difference between TCP based congestion control protocols (e.g. TCP Tahoe, TCP Reno, TCP Sack, TCP Vegas, and etc.) and DCCP is that DCCP does not accept dropping packets for any reason as an indispensable sign of congestion in the network. The DCCP gives the opportunity of selective reaction to the sender node against dropping packets in the network. However, if a malicious node tries to modify drop code in acknowledgement packets, for example it changes drop code from 2 to 3 or removes ECN code point, then sender node does not inform about congestion and it continues to send packets even with more data transmission rate.

TCP Friendly Rate Control (TFRC) controls the congestion problem in the network using a throughput based formula to find appropriate transmission rate for sender nodes. In TFRC, the sender node sends a group of packets to the destination node under a particular transmit rate. Destination node calculates the rate of loss event and sends this information for the sender. Loss event refers to a situation where one or more packets belonging to a window of data have been lost. Sender node uses the rate of loss event to compute RTT. Then, rate of loss event and RTT are applied as input of throughput formula in order to find a proper transmission rate for sender node [17].

According to the above mentioned congestion control protocols, we can say that the congestion problem is detected mostly by dropping packets, while reduction in throughput can be another sign of congestion which is considered in BBR.

2.2 Analyzing the Conventional Solutions for Congestion Problem in Vehicular Networks

The proposed congestion control mechanisms for vehicular networks are mainly based on two standards of the European Telecommunications Standards Institute (ETSI) and the Wireless Access in Vehicular Environments (WAVE). As Fig. 2.1 shows, IEEE 802.11p is used in the physical and data link layer of both ETSI and WAVE standards. WAVE is adopted in the United States, and all the proposed methods in this dissertation are based on the WAVE standard.

<table>
<tr><td>OSI Layer</td><td>IEEE WAVE</td><td colspan="2">ETSI TC ITS</td></tr>
<tr><td>Application</td><td>BSM</td><td>CAM</td><td>Facilities</td></tr>
<tr><td>Transport</td><td rowspan="2">IEEE 1609.3</td><td>BTP</td><td rowspan="2">Networking & transport</td></tr>
<tr><td>Network</td><td>GeoNet</td></tr>
<tr><td rowspan="3">Data link</td><td colspan="2">LLC</td><td rowspan="4">Access</td></tr>
<tr><td>IEEE 1609.4</td><td>DCC</td></tr>
<tr><td colspan="2">IEEE 802.11p</td></tr>
<tr><td>Physical</td><td colspan="2">IEEE 802.11p</td></tr>
</table>

Figure 2.1 VANET protocol stacks [38].

2.2.1 Decentralized Congestion Control (DCC) Mechanism

ETSI defined Cooperative Awareness Messages (CAM) for safety related applications. ETSI applies Decentralized Congestion Control (DCC) in the MAC layer. DCC is a state machine-based approach which switches between three states of relax, active and restrictive based on channel load. These states are shown in Fig. 2.2.

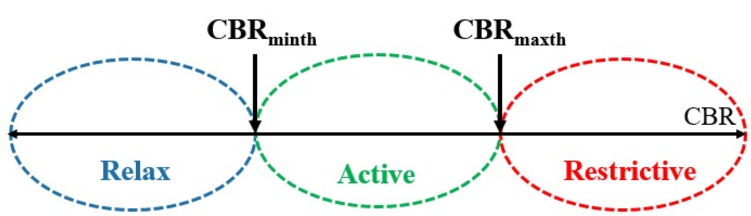

Figure 2.2 Illustration of DCC's states based on the value of CBR.

Switching between the states is based on the load of the channel. Channel Busy Ratio (CBR) shows the load of channel. Indeed, whenever the value of CBR for a specific duration of time is more than a predefined threshold for minimum channel level usage or CBR_{minth}, then the state machine will switch from relax to active state. If the value of CBR for a specific duration of time is more than a predefined threshold for maximum channel level usage or CBR_{maxth}, then the state machine will switch from active to restrictive state. In the restrictive state, the message generation rate will be slower than the active state. Most of the proposed ETSI DCC based methods include transmission power control, message generation rate control and transmission data rate control. In fact, in each state, the DCC applies different transmission power, data rate, message rate and receiver sensitivity. For example, Table 2.3 shows the value of mentioned parameters that are used in [40]:

Table 2.3 Parameters and their values used in [40] for each state of DCC

State	**Packet interval**	**Transmission power**	**Data transmission rate**	**Receiver sensibility**
Restrictive	1000 ms	-10 dBm	3 Mbps	-65 dBm
Active	500 ms	23 dBm	6 Mbps	-85 dBm
Relax	40 ms	23 dBm	12 Mbps	-95 dBm

Considering Table 2.3, when the CBR increased and the state of the machine was switched to the restrictive state, in order to control congestion in the network, the value of transmission power reduced. This parameter has the same value with no changes during active and relax states. Transmission data rate is another parameter that we can see reduction in its value, from relax to the restrictive state. Despite these two parameters, the time interval between packets grew, from relax to the restrictive state. This is to reduce the load in the communication channel.

Finding an optimal value for the thresholds in the DCC mechanisms is a big challenge. Moreover, lack of a value setting procedure for transmission power and data transmission rate that could assign optimal value to them, exists in the literature. For example, in [36], the authors proposed a method in which each vehicle could tune the transmission power based on its speed.

In [41], an utility function optimization method has been proposed to control the amount of transmit power in a cooperative VANET. In [38], the authors proposed channel aware congestion control (CACC) that is based on adjusting the value of transmission power and the value of data transmission rate. In this method, they increased the transmission power and reduced the data transmission rate. Unfair channel allocation, imbalance channel utilization and oscillate between the states are to name but few about current concerns related to the DCC in ETSI especially under critical channel load conditions [42, 43]. Indeed, ETSI DCC suffers from a good configuration, which could respond to the under-utilization and unfair use of the communication channel.

The method named "LIMERIC" presented in [44] is based on controlling message rate in the vehicular networks. In this work, it is assumed that each vehicle can calculate message rate (r_c) using the total channel capacity. The authors assumed that vehicles calculate same amount of message rate at the same time like t. To reach a local fairness among the vehicles,

Table 2.4 Comparison of several proposed congestion control algorithms

Algorithm	**Controlling parameter**	**Required information**
DCC	Transmission power, data rate, channel sensitivity, message rate	CBR
CACC [38]	Transmission power, data rate	Received signal strength, Packet loss
LIMERIC [44]	Message rate	CBR
PULSA [45]	Message rate	CBR, rate from neighbors
ECPR [46]	Transmission power, message rate	CBR, power from neighbors
SBCC [47]	Transmission power	Transmission power from neighbors, number of vehicular
Enhanced Reactive [48] DCC	Data rate	Channel Resource Limit (CRL)
MD-DCC [49]	Message rate, data rate	Minimum channel load

they assumed the desired message rate like r_g which is one share of the total rate. The total rate was equally divided among k vehicles. In calculating r_c, exempting r_g, two other parameters are important: α which is forgetting factor used to decrease effect of the last message rate, and β is weight of the error in the last message rate ($e(t-1) = r_g - r_c(t-1)$). In this work, it was assumed that the value of α, β and r_g are constant. The authors believe that when α= 0.1 and β =1/150 for the maximum 284 number of vehicles the proposed algorithm converges to a message rate in which the channel's capacity is fairly shared among the vehicles. Therefore, they used these values for implementing their proposed method. The big challenge in applying LIMERIC is that we do not know about the optimum value of α, β and r_g. Regarding dynamic change in number of vehicles, for more than 284 vehicles what are the ideal values of α, β and r_g.

In [45], an adaptive transmission range and rate is proposed to control congestion and keep the value of CBR below a threshold. The authors believe that the transmission range is independent from vehicle density, however, the transmission rate depends on the number of vehicles. Therefore, at the first step they used a constant value for transmission range and adjusted the transmission rate. In this method, a target transmission rate was considered. When the value of current transmission rate is less than the value of target rate, the value of transmission rate was multiplied by two. If the value of current transmission rate is more than the value of target transmission rate, the value of current transmission rate must be divided by two. The optimum value for channel threshold is not defined. Besides, it is assumed that each vehicle must exchange channel information which provides extra load for the network.

In the method presented in [46], transmission power and data rate were adjusted in order to increase awareness ratio and control congestion in the communication channel. The range of the vehicles that could receive broadcast messages can be changed by modifying the amount of transmission power. The author applied transmission power adapting technique to affect awareness in the network. They proposed a combined algorithm in which the vehicles can independently modify the value of transmission power and the value of data rate. In this method, vehicles receive the value of transmission power of neighboring vehicles and sort them. Then each vehicle can choose the lowest value of the received transmission power. The vehicles increase the amount of transmission power and data rate with respect to the CBR. In this work, awareness and channel load have the same priority. However, in the real world scenarios, increasing both the transmission power and data rate can escalate the risk of saturating the communication channel. The authors considered a threshold for CBR, however, the optimal value for CBR was not mentioned. Moreover, the proposed approach is a distributed method, thus, the vehicles estimate the channel situation and calculate the value of CBR asymmetrically. In a dynamic environment like vehicular network with an unknown number of vehicles it is difficult to measure CBR correctly by all the vehicles. Hence, the centralized strategies might be a better choice to apply.

Statistical Beaconing Congestion Control (SBCC) has been proposed in [47]. SBCC is a distributed method in which each vehicle calculates how much transmission power must be for beacon messages, regarding the maximum given beacon load in the channel. The amount of transmission power can be computed using number of vehicles and beaconing range and rate. In this method, every vehicle must gather information of transmission power, location and receipt power of the other vehicles for a period of time and store the information in a table. Based on the information of the table, each vehicle estimates the load of the beacon messages in the network. Then each vehicle calculates the amount of transmission power. Since the vehicles calculate transmission power asymmetrically, the beaconing load could not exactly below the predefined maximum allowed beaconing load. Moreover, it is not clear who is responsible for defining the value of maximum allowed beaconing rate in the network. This is an important issue, because a low amount of maximum allowed beaconing load could improve channel busy time in the dense scenarios. In a dense scenario, as the result showed, the authors could not reduce channel busy time very well.

In [48], the problem of under-utilization in the communication channel has been considered. The authors provided an approach to have maximum utilization of the channel when it is busy. In this work, the channel resource limitation was formulated. Then, the time that each node can use the channel with maximum transmission rate could be calculated using the proposed formula. They considered 40 Hz as the maximum transmission rate. This amount can be applied for transmitting packets with size of 400 bytes to 600 bytes and when channel load is less than 64%. However, based on the results, when 70% of the channel is busy and the channel is shared between 100 number of vehicles, the maximum transmission rate for a 400 bytes packet and a 1000 bytes packet is 13.1 Hz and 5.3 Hz, respectively. Note that these low transmission rates are not sufficient for several services in vehicular communications.

In [49], a decentralized combined Message-rate and Data-rate Congestion Control(MD-DCC) has been proposed for V2V communications. In MD-DCC, a threshold for minimum channel load like r_{min} was considered and the beacon frequency must not be less than the value of r_{min}. For this aim, the vehicles must reduce the message rate for beacon messages. However, in dense vehicular scenarios, decreasing the message rate might not be sufficient to control the network congestion. Therefore, in the dense traffic situation the data rate must increase to extend the channel capacity.

2.2.2 Congestion Management in WAVE Standardization

WAVE uses Basic Safety Message (BSM) instead of CAM. In WAVE, the communication channel is divided into seven channels (each 10 MHz), of which channel 172 is dedicated for V2V safety application messages. High Availability Low Latency (HALL) is the other name of this channel. Channel 184 is dedicated to public safety messages and also named High Power Long Range (HPLR). Among the remaining five channels, channel 178 is Control Channel (CCH). The other four channels are Service Channel (SCH) [50]. Fig 2.3 shows the DSRC channel arrangements in the United States.

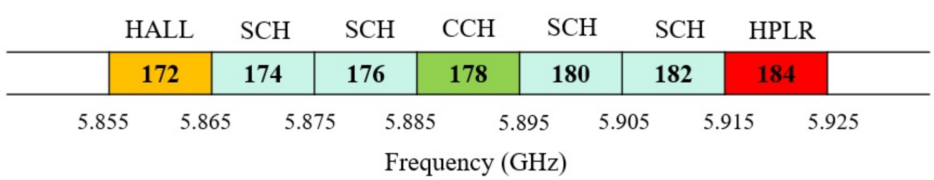

Figure 2.3 DSRC channel arrangement in WAVE.

From the network congestion management point of view, network congestion related solutions (based on WAVE standard) have mainly two phases: network congestion detection, and network congestion control [1]. In the network congestion detection phase, the network management unit discovers that congestion occurred. This phase is the first mandatory step, since if the congestion does not happen, the second phase is not initiated. In the second phase, a congestion controlling mechanism runs to reduce congestion in the network.

Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) is the default method for accessing the communication channel in WAVE [51]. In this mechanism, vehicles have the same priority to access the channel. The vehicle that has data to transfer must sense the medium and if the channel is idle for an Arbitrary Inter- Frame Space (AIFS) period of time, then it could send the data. Otherwise, the vehicle must wait a random time and then run an exponential back-off mechanism and wait for a random amount of time in the range of zero to CW. When the random waiting time is finished, the vehicle sends the data via the communication channel. At the receiver side, when the vehicle received it must send Acknowledgement (ACK) to the sender. If the sender did not receive the ACK, it must double the amount of CW and wait for a new random back-off time and then resend the data. The amount of CW can be increased until it reaches to the maximum value of the CW (CW_{max}). This mechanism is not efficient especially in the dense network since it is very likely that the random back-off periods overlapped and the channel is congested.

Table 2.5 Priorities and parameters of EDCA [52]

Priority	CW_{min}	CW_{max}	AIFS
Voice (AC(3))	$(CW_{min}+1)/4$-1	$CW_{min}+1)/2$	1
Video (AC(2))	$(CW_{min}+1)/2$-1	CW_{min}	1
Best effort (AC(1))	CW_{min}	CW_{max}	1
Background (AC(0))	CW_{min}	CW_{max}	2

In CSMA/CA, the Enhanced Distributed Channel Access (EDCA) mechanism gives priority to messages based on the application. The large CW and AIFS is assigned to the low priority messages [52]. Table 2.5 listed the message priorities and the corresponding CW size and AIFS. The level of priority is shown by Access Category (AC). The highest priority belongs to the AC(3) which has the lowest amount of CW_{min} and CW_{max}. The weakness of the EDCA is that the vehicle with high AC messages has always the highest priority to get the channel. Therefore, a vehicle with lower AC messages should wait a long time to use the channel. This can lead to a greedy behaviour and unfairness in the network.

Congestion Detection and Prediction Methods

In the literature, authors defined strategies for the congestion detection phase. In these methods, they used several assumptions about network congestion happening and then whenever these assumptions got true, they considered that congestion occurred in the vehicular network. The metrics which have influence on congestion in the network are considered to create the assumptions such as vehicle density and channel busy level. For example in [23], the proposed congestion control mechanism is applied when more than 70% of the communication channel is saturated. In [53], authors defined a threshold for channel busy level and whenever the channel is occupied more than the threshold, then the controlling strategy is applied. The authors in [20] considered vehicle's density around vehicles to detect congestion in the VANET. In [27], authors defined several clusters for vehicles and the proposed congestion controlling method is applied for active nodes. By considering several vehicles in each cluster as active nodes, they tuned active vehicle density in clusters. Similar work is presented in [18], the authors distributes vehicle's density in several predefined segments and to avoid network congestion the dedicated bandwidth for each segment must be used just by a member of that segment.

In [54], the authors considered several parameters like Message Delivery Ratio (MDR), average delay, throughput, and network load to propose an aggregate parameter named Q for congestion in VANET. Then, authors normalized Q and used the aggregate parameter for detecting congestion in the network. Also, Zang et al. [55] did the same work in case of congestion detection based on parameters. In the proposed work whenever each of two following conditions are happening they supposed that congestion will occur and then the congestion control mechanism will be executed automatically. First is based on receiving or sending a safety message with high priority, and second is based on channel usage level. Different congestion control mechanisms are assigned for states of when more than 95% of channel and when more than 70% of channel is used. In the proposed mechanism they used a static and fixed threshold for channel usage while from the results congestion may happened also in the interval of [60%,70%] channel usage.

Detection is about something that exists, however it is not discovered or it is hidden.Therefore, when we speak about detecting network congestion it means that it has occurred (or very close that happening) and needs to be discovered, like the works that are mentioned above. However, network congestion prediction is about predicting a congestion state of network before it happens. Logically, we keep away from situations or events that we are sure about happening them or they have a high possibility of occurrence. Therefore, usually we start to analyze evidence and parameters then create an accurate anticipation about the event happening and then based on results we start to use strategy to prevent events occurring in the near future. If we assume that congestion is like that event and we do not want it to happen, hence we try to predict it before happening and then do avoidance strategies. Network congestion prediction is not considered by authors in literature.

Moreover, the first phase of network congestion management which is highly important is not considered very well in the literature. Authors did not explain well about the congestion detection step and they put all of their efforts to control congestion in the network, not to detect or beyond that to predict it.

Congestion Control Methods

Based on the layers of TCP/IP model, the proposed congestion controlling mechanisms are considering congestion problem in one or more layers. In cross-layer techniques the congestion controlling methods consider all the layers [1]. Fig. 2.4 shows the schema of the proposed congestion controlling methods in networking layers. In the application layer, applications can help control network congestion by reducing the number of generated packets per unit of time. In the transport layer, data stream adjustment is a method to control congestion in the network. Proposing new routing methods and creating fair bandwidth allocation techniques are considered as the network layer congestion controlling approaches. Most of the congestion controlling methods have been proposed to apply in the Medium Access Control (MAC) layer such as transmission power and data rate adaptation methods and prioritizing and scheduling techniques. Physical layer has a key role in the most proposed congestion management approach, since channel sensing and measuring the channel load are the main part of detecting congestion in the vehicular networks.

Kumar et al. [56], explained three types of messages: beacon, emergency and query. The query message is about information that the vehicle needs to know. The proposed congestion control algorithm throw-off the same messages by using a neighbor table. For each received message the values of type, identity, current location, message, directions, and speed will be compared with records in the neighbor table. Afterwards, if an incoming message is already in the neighbor table, the new message will be discarded, otherwise the message will be added to the table. Besides, the congestion control algorithm uses a counter, if the counter is zero the message is deleted from the table. This method has several drawbacks. First of all, the delay is significantly high due to comparing the information of the recently received message with the other recorded messages to find similar messages is a time consuming task. Moreover, recording the messages needs enough memory especially in dense traffic situations. Therefore, this method make extra overhead for vehicles. Additionally, in high dense regions, eliminating the similar messages may reduce the traffic load, however, it is not enough that could significantly reduce the risk of occurring congestion in the vehicular network.

In [18], the authors defined several segments and assigned each vehicle to a segment. The segments are different in the number of assigned vehicles. A node in each segment determines

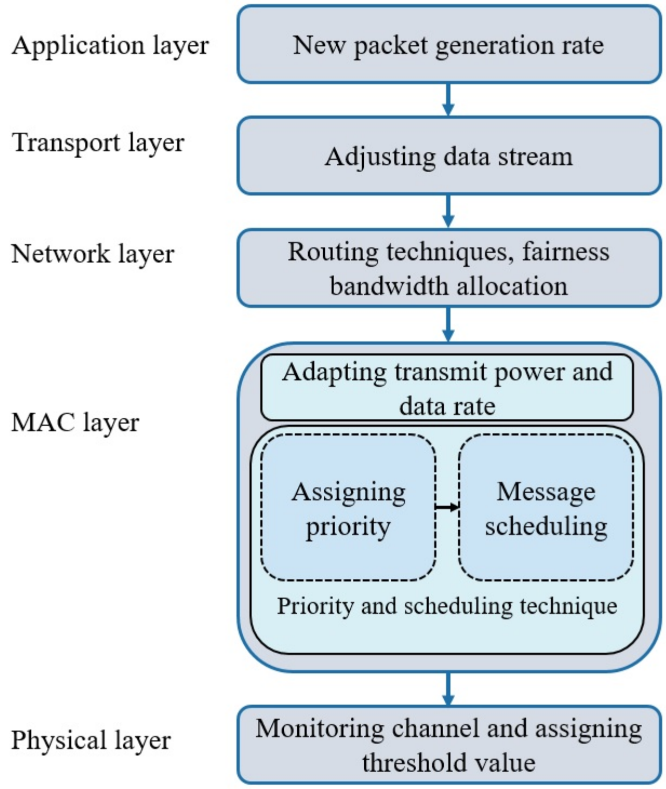

Figure 2.4 Schema of the proposed congestion control approaches in TCP/IP networking model [1]

which node of the segment can use dedicated bandwidth during a specific time interval. Since segment densities are not equal, bandwidth allocation is not fair. Because, a node in a denser segment has to wait more to use dedicated bandwidth. Moreover, time of using bandwidth for a node in a crowded segment is less than a node in a non-crowded segment.

In [23], the authors improved packet loss, average delay and probability of collision by applying K-means clustering technique (unsupervised algorithm). As Fig. 2.5 shows the proposed strategy is divided into the three parts: congestion detection, data control and congestion control. In the congestion detection unit, it is assumed that congestion happens whenever channel usage comes up to 70%. In data control unit messages are collected, filtered and then clustered. In the congestion control unit appropriate communication parameters are assigned to each cluster. In data collecting, it is assumed that all sent and received messages by vehicles are gathered in the Road Side Unit (RSU). Message filtering process is briefly mentioned like "removing same messages in RSU". For each buffered message in RSU like "m", a search algorithm must be executed to find message/s with the same information to "m" in all containing portions (a message involves several portions). Performing a search method for all collected messages in RSU should be done before a message gets outdated. Therefore, it was better to explain more about how many messages are assumed in simulation and also which searching algorithm has been used for finding same messages.

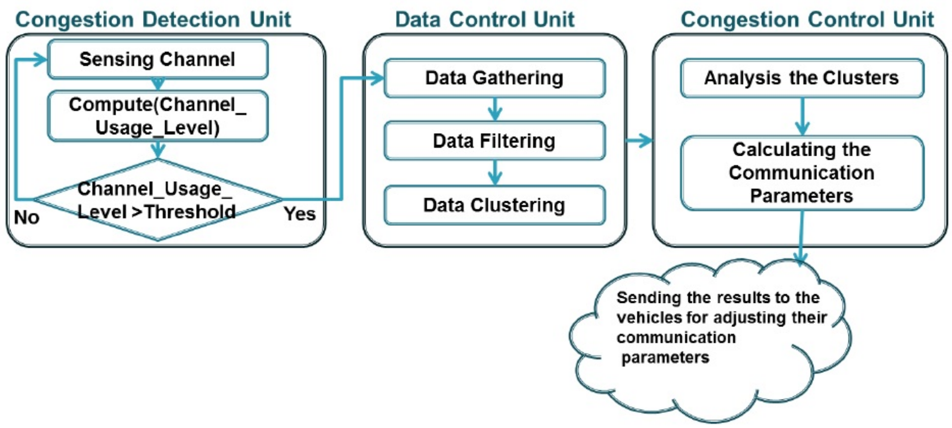

Figure 2.5 Flowchart of the machine learning congestion control strategy proposed in [23]

In [19], the authors worked on solving congestion problem for emergency messages with high, medium and low priority. They added one byte header to the safety messages named Hop_{cpt}. The value in the Hop_{cpt} shows the level of priority of the message. Messages with highest priority have the lowest value in the Hop_{cpt}. Moreover, the lower value of Hop_{cpt} indicates that a safety message came from a vehicle close to the receiver vehicle. Therefore, it could mean that a road danger is very close to the receiver and the safety message must be transmitted very soon. A large value of Hop_{cpt} shows that the road danger is far from the vehicle. Consequently, transmitting the safety message with delay could not make a serious risk for the receiver vehicle. The congestion control mechanism was based on three steps of assigning priority to each emergency message, finding congestion, and tuning transmission power and beaconing rate. To detect congestion in the VANET, the vehicles must calculate the average waiting time to access the channel, collision rate and beacon reception rate, for a period of time. Moreover, each vehicle must record information of: vehicle ID, direction, speed, transmit power and expiration time. Then, the vehicle selects the minimum value between the current transmit power and the transmit power of the received message. Then beacon rate must be calculated and assigned in a manner that the bandwidth divides fairly between the vehicles. Applying this method in a dense vehicular environment could make overload and increase end-to-end delay, since the vehicles must do all the tasks and computations for each safety message.

Broadcast storming enhances both reliability and probability of congestion occurrence for emergency and beacon messages. In [21], the authors introduced “fully distributed congestion control (FD2C)” and “Unicast Multi-hop Data Dissemination (UM2D)” techniques in order to achieve high data delivery plus avoiding congestion. They tuned transmission power of each vehicle based on the crowding level surrounding the vehicle. In the proposed FD2C algorithm, as long as local communication congestion does not exceed communication congestion threshold, transmission power will be increased, and otherwise transmission power will be diminished. In the work, the value of communication congestion threshold is not defined clearly.

In [57], authors proposed a congestion control mechanism which works in profit to safety messages transmission in VANET. They believed that vehicle’s location and vehicle’s velocity are two factors that must be considered in control channel allocation. Therefore, authors defined a utility function for probability of message transmission in a network, in which delivering safety messages to the nearest neighbors has highest priority. Moreover, the maximum value of utility function can provide the minimum delay for VANET.

Taherkhani and Pierre [24] proposed a method for solving network congestion problem in Vehicular Ad hoc Network (VANET) by using an open-loop strategy. From the simulation results, the authors were successful in reducing average delay and number of packet loss. Indeed, they could control the congestion before it occurred by using a mechanism based on giving priority to each message and then scheduling them in two different CCH and SCH. With improving in average delay and number of lost packets, they could increase average throughput in the network. Main feature of VANET is changing dynamically in the number of vehicles, the speeds, distance, the direction of vehicles, and network topology, hence decisions must be made by using dynamic factors. In [24], authors used priority and scheduling techniques by considering dynamic parameters like velocity of sender’s vehicle, usefulness metric, message validity, distance between sender and receiver, and direction of sender and receiver in order to assign priority to each message. Then, they used two different dynamic scheduling strategies: first based on priority (DySch) and second based on minimizing delay (TaSch) and jitter. Logical concept of the work is congestion acceptance like an asleep problem that must not be waked, by using proposed mechanisms. Moreover, end-to-end delay in the proposed method is high because the vehicle must reschedule the service channel and control channel which make overheads for them.

Zemouri et al. [20] proposed a model to predict density around a vehicle in the next time window by using beacon’s information. They supposed a beacon message contains: ID, current position, speed, destination, number of vehicles ahead, and number of vehicles behind.

By using information of received beacon messages, a vehicle like “V” can find out: furthest back vehicle like “B”, furthest forth vehicle like “C”, closest back vehicle like “D” and closest forth vehicle like “E”. Next, from speed information in beacon messages, if “C” is slower than “V”, then the number of vehicles ahead of “E” will be used as predicted forth density for “V” in the next time window. Also, if “B” is faster than “V”, then the number of vehicles behind “D” will be used as predicted rear density for “V” in the next time window. Then based on density prediction, the vehicle can adjust parameters in order to avoid congestion for the next time window.

Hasanabadi and Valaee [27] proposed Synchronized Persistent Coded Repetition (SPCR) algorithm. By SPCR, each active vehicle node broadcasts composition linear coding of messages which are selected randomly from its queue. If the number of vehicles in a cluster is N, then the congestion control mechanism randomly selects n node as active node (which is defined as $n \leq N$) and abandons all messages from $(N - n)$ inactive nodes. Therefore, the value of n can be maximally equal to N and minimally equal to zero. If $n = N$, it means that all N vehicles in the cluster are active and all can broadcast messages. It is like no congestion control mechanism is applied, because as the authors mentioned in their work, the objective of proposing the congestion control mechanism is to control the amount of messages by abandoning . On the other hand, if $n = 0$, it means all nodes are passive and whole safety messages will be dropped and it is dangerous especially in critical situations like facing road hazards.

Two goals of 1) avoiding congestion in channel by keeping beaconing rate for each transmission power less than threshold like C and 2) enhancing number of delivered beacons messages by assigning minimum beaconing rate like r_{min}, are considered together in [58]. The authors proposed “fair adaptive beaconing rate with multiple power levels for inter vehicular communications (FABRIC-P)”. In FABRIC-P each vehicle must calculate best rate for a beacon message with use transmission power p which $r_{min} \leq$ best beacon rate $\leq R_{max}$ (the topmost beacon rate for a vehicle is R_{max}). Moreover, every vehicle informs a neighbor’s vehicles about its beaconing rate, transmitting power, and traffic level of the wireless channel by broadcasting each beacon.

Controlling congestion by proposing adaptation methods for the amount of CW is considered in several research works. In [59], a deep Q-learning adaptive method is proposed to find optimum value of CW. Three actions of: keeping, increasing and decreasing the amount of CW were considered. The method is proposed in two different models: discrete and continuous changes in the CW’s value. "Binary exponential back-off algorithm" was applied for the discrete model. Results show that the proposed method could improve performance of VANET in comparison with the simple Q-learning method and DSRC with CW=31. However, as the results indicate, a dense vehicular environment still suffers from low packet delivery ratio and high average end to end delay.

In [52], transmission power and CW joint adaptation technique has been proposed to control congestion in VANET. Inspired by EDCA technique, in the proposed method messages have priorities, however the size of CW is not consent and can be doubled or halved. The size of CW is doubled if the estimated collision rate is more than a predefined threshold. This value is halved if the estimated collision rate is less than a predefined threshold. Moreover, the vehicles must calculate vehicle density. If the vehicle density is less than a predefined threshold, then the vehicle can apply the maximum transmission range. Otherwise the vehicle must calculate the transmission range. Then, the vehicle must use the lookup Table 2.6 to find the corresponding value of transmission power using the transmission range.

Table 2.6 Transmission ranges and corresponding transmission powers [52]

Transmission range (m)	Transmission Power (dBm)
0—9	−20
10—49	−12
50—100	−5
100—125	−3
126—149	1
150—209	4
210—299	6
300—349	10
350—379	12
380—449	14
450—549	17
550—649	20
650—749	24
750—849	27
850—929	29
930—970	31
971—1000	32
>1000	N/A in DSRC

In [60], a lookup table is considered to help vehicles find appropriate transmission power of each transmission rate. Then, the value of CW for emergency messages and basic safety messages must be defined. A low value of CW for emergency messages was assigned in order to minimize waiting time for this type of messages. The basic safety messages could be transmitted using a higher value of CW. Therefore, the basic safety messages might be awaited more than emergency messages in this work. The obtained results show that the emergency packet delivery ratio was improved, however, this metric remains at a low level for basic safety messages. Moreover, based on the results, average delay for basic safety messages still requires significant improvements.

In [61], the vehicles can access to the communication channel during predefined time slots. Vehicles can make reservations for the time slots in advance. When a vehicle wants to transmit a beacon message, it must see if there is a reservation for the coming time slot or not. If the vehicle was reserved for the next time slot, it can use it and start the data transmission. Otherwise, the coming time slot is assigned to the other vehicle that reserved it before. If there is not any available time slot to reserve, the vehicle must wait for a random back-off time and after passing that time, the vehicle should check if there is any available time slot to reserve. The vehicle can only transfer the beacon messages during their reserved time slot or when the next time slot is available and was not reserved by other vehicles. Moreover, the vehicle should transfer the messages based on the EDCA priorities (Tabla 2.5).

In [62,63], Tabu Search method has been considered to adjust the value of transmission range and data rate. The authors believe that finding optimal value for these parameters in such a dynamic environment like VANET is a NP-hard problem that should be solved by Meta-heuristic techniques. In [62], authors proposed the Uni-Objective Tabu Search (UOTabu) method to control congestion problem in VANET. Congestion is detected by estimating the channel load. A short-term memory was applied to find the transmission range and rate in which the network has minimum delay. Therefore, delay is the objective function in the UOTabu. In [63], Multi-Objective Tabu Search (MOTabu) method is proposed to control network congestion. MOTabu has two components: congestion detection and congestion control. The congestion detection unit is the same as the UOTabu and detects congestion by measuring the channel usage level. In the congestion control unit; same as UOTabu; the objective is to find a close to optimal value for transmission range and rate. However, in MOTabu the Tabue search method was applied to minimize delay and jitter. In MOTabu, a short term memory is applied to make sure that the new solution had not been considered in the past. A list of solutions is kept in the short memory and the new solution must be checked with the solutions in the table. If it is a repetitive solution and there is already the same solution in the table, then the new solution must be ignored. Otherwise, the new solution is added to the end of the table. The capacity of the table is limited and when it is reached, the solutions from the top of the list are removed. Mid-term memory of MOTabu selects the recent best solution from the end of the table to find the new solutions in their neighborhood. However, the MOTabu might be caught in a local minimum trap. Therefore, long-term memory was applied to restart searching and generating new solutions. Fig. 2.6 shows the components of the MOTabu mechanism [63].

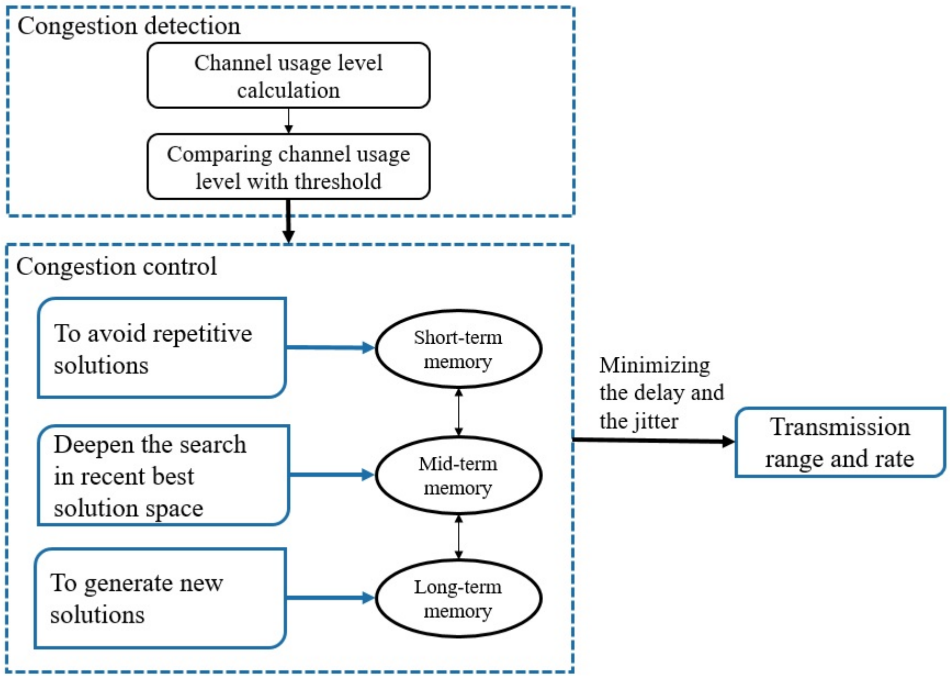

Figure 2.6 MOTabu mechanism.

In the heterogeneous vehicular network, vehicles may tend to switch between different technologies based on their preferences like achieving better data transmission. Using WiFi instead of cellular network when congestion happens in vehicular environment is proposed in [22]. In this scenario, when vehicle user which is connected to cellular network encounters delay in performing an application (e.g. because of lack of required resources in cellular network), then by computing utility function, vehicle user estimates whether it is cost effective to wait for cellular network or stop using cellular network and connect to WiFi. From the utility function's result, each vehicle decides whether to use WiFi or not. If the utility function shows that the vehicle cannot use WiFi, then it may wait for time t till utility function's result is changed to WiFi intended value, or the vehicle may use cellular network. When a vehicle like v decides to use WiFi, it announces intelligent offloading Engine (IoE) in Base Station (BS), but IoE randomly replies to the vehicles, which have the same request as v has. So, it is possible that a vehicle waits a long time for response from IoE and consequently with changing effective parameters in value of utility function like location, number of vehicles, which use the requested WiFi and etc. the result of utility function for the vehicle v will be changed. In this scenario, having desires resulting from utility functions alone cannot solve congestion problems and having a good chance of being selected by IoE to respond is mandatory. In [22], as mentioned for any change in value of: cost of using cellular network, cost of using WiFi, and number of users that are served by access point, vehicle user should calculate utility function. In a dynamic environment like the vehicular network, the topology of the network changed quickly because of the fact that vehicular nodes move and change their location very frequently, it seems that vehicle users should calculate utility function very frequently as well. These many calculations consume energy, resources and time for vehicle users. In [22], for the case that value of utility function is zero, the authors believe that if vehicle user can find out that whether value of utility function will be changed in near future or not, then it can help vehicle user to make better decision among the choices such as wait or connect to the cellular network. If the vehicle user estimates that the value of the utility function will be increased to a positive value very soon (for example in the case of reduction in the number of vehicles in the same access point's coverage area), then maybe it prefers to wait instead of using cellular networks. But, a good prediction needs accurate information such as the exact place of other vehicles, direction of vehicle movement, etc. to determine if they are going to leave the coverage area of the access point or not. As results show in [22], the authors could mitigate traffic load in cellular networks. However, in this work all the computations and decisions must be done by vehicle users that have not widely network view. Therefore, it was better if these calculations and decisions had been made at infrastructure level. As a suggestion, applying a SDN-based architecture can be a good choice, since global view and programmability of controllers are two advantages of SDN that can help performance of proposed mechanisms especially in terms of prediction of next value of utility function which discussed above.

Another form of data offloading is considered in [64]. Huge volumes of delay tolerant data (for example with tolerance of 5 hours in delay) are transferred from a vehicle node which stores the big data to the destination node [64]. During big data transmission vehicles which have idle resources are used to store and forward data, instead of using other network resources.

Network selection based on a utility function is another solution that has been worked in [28]. In [28], based on the vehicle's Global Position System (GPS) location, cloud generates and sends a list of RSUs with their information to the vehicle. By using received information, the vehicle computes utility function. Then, an alliance of vehicles with fair and equal profit from the selected network will be created. Afterwards, handoff will be executed by each vehicle in order to start data transmission between vehicles and RSUs. In [28], regarding the extensive area that every cloud can support, for any location update of every single vehicle, cloud must create a new list of RSUs and other related information. It means many computations, huge data generation and dissemination, and even delay in response. In such a scenario, dividing a large covered area of cloud to the small regions and then using a local computing object for storing and retrieving information of local RSUs in each region, can be more efficient than applying a cloud base system. Moreover, as authors mentioned in the work, because of vehicles' movement, location of vehicles in near future can be changed. So, a strong analysis and prediction of the future location of vehicles can prevent the problem of useless data generation. Because, it is possible that in the time that vehicle receives information from the cloud, vehicle location has been changed and consequently the delivered information gets invalid.

In several existing works, authors decided to cope with congestion in the network by adjusting data transmission rate. For instance, tuning transmission rate along with transmission power have been considered in some works related to the congestion problem in vehicular network [36, 42, 46].

In [65], authors considered 5G network slicing for vehicular networks and more specifically for autonomous car users. They proposed Radio Access Technology (RAT) slicing, core network slicing and user device slicing for vehicle to everything (V2X) communications, with the aim of improving data transmission and QoS.

In [66], two different architectures are proposed for network resources sharing. In the proposed gateway core network model, authors applied common mobility management entities. However, in the multi operator core network model, they used separate interfaces to make links between Shared Radio Access Network (RAN) and network operators entities. Although the former is more cost effective, the later is more flexible.

In [67], authors proposed a service model by which cellular operators could sell their network slices as a service. They believe that creating several data pipes with different QoS and offering various guarantied services are the advantages of the proposed approach.

In [68], four mathematical models are proposed for resource allocation in network slicing of 5G. The first proposed model named "General Model" is based on linear and nonlinear optimization methods with the aim of improving packet delay, throughput and reliability of networks. In the second proposed model, the three main actors of: network operator, the tenant, and the network users are the players of a game. They try to maximize the profits for the network resource owners while satisfying the users by offering affordable prices for users service requirements. For example, offering various types of network slices with varied services and resource capacities at different prices. The third proposed model is prediction models, which are based on analyzing past experiences and current situation of the network. The last proposed model is about boosting networks, specifically fast recovery in unpredictable network states like network congestion. Our proposed network slicing method for heterogeneous vehicular networks belongs to the last two categorized models.

In [69], the idle time of a service channel of DSRC technology is used to transmit non safety messages. Control Channel is used to transmit safety message transmission and reserving a service channel for non safety messages.

In [70], a network slicing perspective is proposed for vehicular networks. In the proposed network slicing model, network slices can be created based on the application needs. For example, low latency is required for safety based applications and can be provided by a network slice.

In [71], the four RAN slicing approaches and their challenges for vehicular networks are studied. The resource allocation policy for safety data traffic is considered. When a huge volume of safety driving data is generated and there are not enough available free resources in a slice to dedicate, the available network resources of other network slices are used. In this approach, safety data has higher priority than other types of data in vehicular networks.

2.2.3 Taxonomy of the Proposed Congestion Management Mechanisms

Fig. 2.7 shows the taxonomy of the congestion management strategies proposed in the literature. In this figure, the proposed methods are divided into two standards of ETSI and WAVE. Then the metric used in the congestion detection part of the methods are shown in the gray rectangles. In the lower levels of the taxonomy, the orange rectangles show the congestion controlling strategies. The proposed methods applied a single or combination of strategies (orange rectangles) using novel methodology.

In the literature, the proposed congestion control mechanisms are following ETSI or WAVE standardization. ETSI-based controlling methods, mostly consider CBR to detect the congestion in the communication channel and consider one or combination of transmission power tuning, data rate adjustment, message rate controlling and channel sensitivity, for controlling congestion in the vehicular network. The big challenge of these methods is to find the optimum value of CBR to use as a threshold. In most of these methods, the controlling strategy is designed in a way that the channel usage does not exceed the threshold. Therefore, the value of the threshold is important because a low threshold has a risk of under-utilizing the channel. This means that vehicles do not use all the available and accessible capacity of the channel due to the channel congestion phobia. On the other hand, considering a high value for the threshold of CBR can put the network at risk of congestion.

ETSI methods are distributed methods, therefore, vehicles must sense the channel, calculate the channel busy level and run the other necessary computations to control the congestion. In the real world scenarios, it is difficult to say that sensing the channel and calculating the

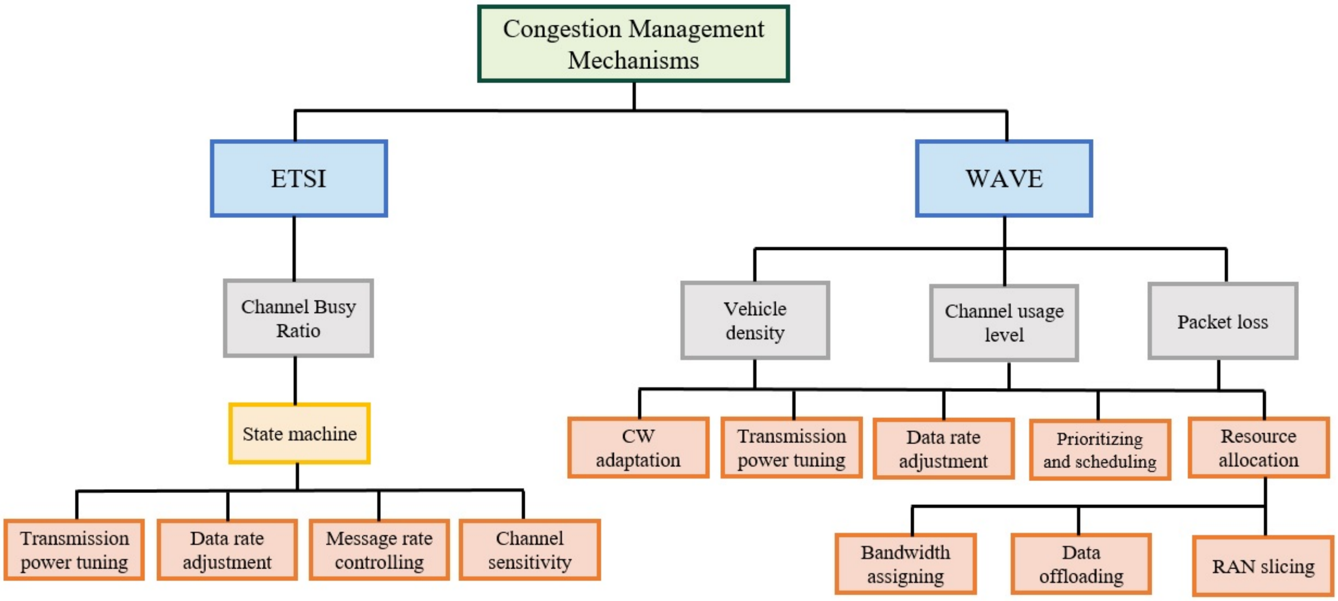

Figure 2.7 Taxonomy of the proposed congestion management methods.

CBR are done at the same time by all the vehicles. It may cause asymmetry in channel busy levels. Consequently, it is likely that all the vehicles do not calculate exactly the same (and real) value of the CBR, especially in such a high dynamic environment like vehicular network. This issue brings unfairness to the network.

Considering ETSI-based approaches, the congestion controlling mechanisms are mainly based on the four strategies of: tuning the transmission power, data rate adjustment, message rate controlling and channel sensitivity.

Congestion management mechanisms using the WAVE method have been considered in the literature. In the proposed methods, vehicle density, channel usage level and packet loss are the metrics used to detect congestion in the vehicular networks. The proposed methods are mostly distributed in which vehicles must sense the environment using the received and stored information from other vehicles, do computations and make congestion controlling decisions. However, in some of the works, centralized methods have been considered in which an infrastructure unit such as RSU is responsible to monitor the vehicular environment and make congestion controlling decisions if it is necessary.

Considering the (high number of) proposed distributed methods to control congestion in the vehicular networks, vehicles need enough and appropriate computation and storage resources to monitor the network and analyze the huge received information and make the controlling decisions. However, the vehicles have small and limited resources. The high dense vehicular environment that changes dynamically, these resources are not enough to analyze the network situation and make decisions. Therefore, high overhead and end-to-end delay are the weaknesses of the proposed methods. Moreover, these methods need significant vehicle cooperation, otherwise, the delays and overhead can be even higher.

In the proposed methods, estimating the number of vehicles around, measuring the channel usage level and calculating the amount of lost packets were the predominant strategies to detect congestion in the vehicular networks. In the proposed mechanisms, there are two different approaches: first, the authors considered that congestion occurred and then detection strategies should recognize the happening of congestion in the vehicular network. This approach is applied in the closed-loop solution to control congestion in the network. Based on the second approach, the thresholds are predefined for the amount of metrics such as channel usage level or packet loss, then the controlling mechanism executes when the value of threshold is met or exceeded. This approach is employed in the open-loop congestion controlling strategies. Indeed, the network congestion is controlled before it occurs in the network.

Adaptive techniques for the value of transmission power, data rate and CW or combination of them are common congestion controlling strategies among all methods.

Prioritizing and scheduling is another technique that has been applied to control congestion in the vehicular network. This method could improve delay for the high priority messages. However, the applications with lower priority messages may wait longer to access the communication channel. Therefore, they experience significant delay. Besides, scheduling a large number of messages in a high dense and dynamic environment like vehicular network makes high overhead in the network.

Resource allocation is another approach for Congestion controlling in the vehicular networks. In this regard, assigning extra bandwidth to the high priority messages is a method to reduce suffering safety related applications from network congestion. However, this approach may cause unfairness in the network since the lower priority messages must wait longer and be received with delay because the higher portion of available bandwidth is dedicated to the high priority messages. Offloading data to control congestion in VANET using other technologies such as WiFi and cellular network requires further authentication and extra cost for the vehicle's users. RAN slicing is a novel method introduced in 5G technology to share the network resources with the aim of improving QoS ,providing service requirements and preventing congestion in the network. Network slices are created and modified as needed and based on the users requirements. Consequently, flexibility and compatibility of vehicular networks are increased using this method.

Published and submitted papers are the main parts of this dissertation. The next four chapters contain the published/submitted papers.

The main objective of this dissertation is to propose intelligent congestion prediction and avoidance methods using AI methods for HetVNET. As mentioned in Chapter 1, regarding the main objective, we defined three sub-objectives in this dissertation. In each sub-objective, we employ an AI method to solve a regression or a classification problem along with applying new network technologies and with the aim of predicting or avoiding congestion in the HetVNET.

The article presented in Chapter 3, considers the network prediction problem as a classification problem. This article entitled "A Centralized and Dynamic Network Congestion Classification Approach for Heterogeneous Vehicular Networks" has been published as a journal paper in "IEEE Access" journal. A Naive Bayesian classification method is proposed to predict the network congestion warning states of the HetVNET. Moreover, SDN and fog computing technologies are applied to propose a centralized and dynamic cloudy-fog intelligent congestion prediction architecture for HetVNET. In this approach, roads are segmented by a centralized management unit and based on the number of vehicles in each segment in the future time t, the Naive Bayesian classification method predicts a congestion warning/non-warning state in the network.

The article presented in Chapter 4, considers the network prediction problem as a regression problem. A utility function is proposed to show the network performance in terms of smooth data flow in the HetVNET. A Multiple Linear Regression method is applied for regression prediction. This part of the dissertation was published as a conference paper with the title of "A Multiple Linear Regression Model for Predicting Congestion in Heterogeneous Vehicular Networks" by IEEE publisher. In Chapter 5, a GRNN method is proposed to predict the value of the utility function. Then, based on the prediction result an avoidance mechanism executes to prevent congestion in the network. A transmission power adaptive method is proposed to avoid network congestion using the result of the GRNN prediction model. This chapter of dissertation is submitted in "IEEE Transactions on Intelligent Vehicles" with the title of "An Intelligent Congestion Avoidance Mechanism Based on Generalized Regression Neural Network for Heterogeneous Vehicular Networks".

The article presented in Chapter 6, considers a deep learning method and network slicing technique to help control the congestion problem in a dynamic environment like HetVNET. This article was published in "Telecom" journal with the title of "A Conditional Generative Adversarial Network Based Approach for Network Slicing in Heterogeneous Vehicular Networks". A CGAN is proposed to provide configurations used to create new network slices. In this approach a CGAN-SDN architecture is proposed to apply the CGAN method in the controller of the SDN. Regarding the high dynamicity of the HetvNET, the aim of this article is to propose a method that can be used to generate network slices fast and based on the previous successful experiences in terms of occurring network congestion in the HetVNET.

Thus, the main objective and the three sub-objective of this dissertation are met and will be extensively explained in more detail in the following four chapters. Predicting network behaviour in terms of how smooth data flowing is in the network is significantly important and helpful in order to make appropriate decisions and executing controlling mechanisms and consequently preventing congestion occurring in the network. Therefore, the contributions of this dissertation help improve the network performance, QoS and user experience of HetVNET. Moreover, the presented methods help us towards forming an autonomous and adaptive HetVNET.

CHAPTER 3 ARTICLE 1: A CENTRALIZED AND DYNAMIC NETWORK CONGESTION CLASSIFICATION APPROACH FOR HETEROGENEOUS VEHICULAR NETWORKS

Farnoush Falahatraftar, Samuel Pierre, Steven Chamberland

Mobile Computing and Network Research Laboratory (LARIM)
Department of Computer and Software Engineering, Polytechnique de Montréal
Montreal, QC H3T 1J4, Canada
Email : {farnoush.falahatraftar, samuel.pierre, steven.chamberland}@polymtl.ca.

Status: published in *IEEE Access*, August 2021.

Abstract Network congestion-related studies consist mainly of two parts: congestion detection and congestion control. Several researchers have proposed different mechanisms to control congestion and used channel loads or other factors to detect congestion. However, the number of studies concerning congestion detection and going beyond into congestion prediction is low. On this basis, we decide to propose a method for congestion prediction using supervised machine learning. In this paper, we propose a Naive Bayesian network congestion warning classification method for Heterogeneous Vehicular Networks (HetVNETs) using simulated data that can be locally applied in a fog device in a HetVNET. In addition, we propose a centralized and dynamic cloud-fog-based architecture for HetVNET. The Naive Bayesian network congestion warning classification method can be applied in this architecture. Support Vector Machine (SVM), K Nearest Neighbor (KNN) and Random Forest classifiers, which are popular methods in classification problems, are considered to generate congestion warning prediction models. Numerical results show that the proposed Naive Bayesian classifier is more reliable and stable and can accurately predict the data flow warning state in HetVNET. Moreover, based on the obtained simulation results, applying the proposed congestion classification approach can improve the network's performance in terms of the packet loss ratio, average delay and average throughput, especially in the dense vehicular environments of HetVNET.

Keywords: Vehicular networks, congestion control, classification methods, network congestion prediction, WAVE.

3.1 Introduction

A Heterogeneous Vehicular Network (HetVNET) enables a connected vehicle to inform other smart vehicles on the road by sending and receiving safety driving information (e.g., the location, speed, direction, road hazards, road traffic congestion, and road accidents) using Dedicated Short Range (DSRC) and Long Term Evolution (LTE) technologies [2]. Minimum human reaction time is 500 ms [23]. Due to network congestion, if an emergency message is received with a delay of more than 500 ms, then the safety applications are useless due to weak network performance.

In the literature, there are usually two phases of network congestion: the first is the detection of congestion, and the second is the relief of congestion by the use of a control method or a prevention mechanism. However, the approach to solve the problem of network congestion has focused mainly on controlling congestion, which is in the second phase. For the first phase, the authors used assumptions to determine the congestion, such as defining a threshold for the channel busy level [23, 32, 42], and vehicle density [18, 20, 27]. Although congestion detection is not widely considered in current studies, it is a key part of addressing network congestion problems. If network congestion is not sensed and detected, applying the controlling mechanism (phase two) is meaningless. Indeed, to initiate the second phase, it is necessary to meet the first phase, which is congestion detection [23]. The obtained results in the related literature [23,72,73] demonstrate that congestion in a dynamic environment, such as a vehicular network with a high number of vehicles, has not been completely controlled. Consequently, this limitation could threaten the stability of the network performance, and such instability is obvious in the published results. This challenge needs to place greater emphasis on studying and proposing novel intelligent methods, which are based on analyzing the network performance and establishing avoidance mechanisms before congestion occurs in the network. With regard to the importance of reaching a stable performance in a highly dynamic network environment such as HetVNET, we need to predict congestion in networks and then execute an avoidance mechanism. In this paper, we propose an intelligent mechanism for predicting congestion in such networks to solve the problem of instability in network performance in dense vehicular scenarios.

Applying Machine Learning (ML) methods in building congestion management approaches (congestion prediction and control) in a highly dynamic network such as a vehicular network was considered in [1, 74, 75] to be an open challenge and new future path toward centralized and dynamic congestion network management. Recent technologies, such as Software-Defined Networks (SDNs), Network Function Virtualization (NFV) and fog computing, provide programmable features along with high storage and computing power to the networks. Relying on the advantages of applying these technologies, predicting the behavior of vehicular networks in terms of data flow, especially in the case of congestion in the network, is a novel and worthwhile research path. Referring to this open challenge, as a contribution, we propose an intelligent and dynamic network architecture using a Naive Bayesian classifier to predict the warning state in the data flow situation in HetVNET.

Generally, ML methods must analyze data and perform computations to achieve accurate and reliable results. With regard to this concern and many other advantages that will be mentioned, fog computing is used in this work. Fog computing technology changes the traditional architecture, in which only clouds play key roles, by using powerful objects close to devices in the network [6, 7]. Fog computing supports mobility, location awareness, and real-time interaction. Well designed and configured it can improve metrics [6, 8]. Although the application of fog computing technology has significant advantages, there is still a lack of intelligent methods that use fog computing technology to solve the HetVNET congestion problem in the literature. With regard to computing, storage, data management and analysis, in addition to network abilities of fog computing units [76], a novel approach is to implement a robust, supervised network congestion classifier method in fog computing units with the aim of improving the performance of HetVNET by providing a smooth data flow. The implanted prediction model can be created and evaluated at the cloud level. Thus, a fog congestion predictor unit can predict congestion locally using the current information on the parameters, which make up the prediction model. In fact, data are sent to the fog devices that are close to the vehicles, and any required computational process can be performed at the fog devices. Making decisions using fog devices that are close to vehicles in a time-sensitive situation is advantageous, because the latency is reduced and the reliability is improved [6]. Moreover, the data in a local and limited area, such as traffic zones, are less than the big data generated from unlimited vehicles located in different zones. Processing data that are more local and smaller in volume at fog devices is less time-consuming and more efficient than processing and analyzing enormous amounts of data remotely [74].

In addition, the proposed approach is compatible with both the European Telecommunications Standards Institute (ETSI) and the Wireless Access in Vehicular Environments (WAVE) for V2V communication. Therefore, both standards can use the result of the proposed congestion prediction, and then in the case of the congestion warning state, ETSI or WAVE specific controlling mechanisms can be applied.

In this paper, we went beyond detection, and we proposed a classification congestion prediction method. Congestion prediction using ML methods is a novel future path toward creating intelligent congestion management in vehicular networks [1]. Predicting congestion before it occurs in the network and applying the controlling mechanisms in advance can increase elasticity, sustainability and tolerance of such a dynamic network as HetVNET. Based on the prediction approach, network parameters can be modified with the aim of preventing congestion in the future. Compared to the literature, the proposed approach of this paper makes significant contributions as follows:

- First, considering the importance of the congestion detection phase in heterogeneous types of networks, predicting the warning state of network congestion (before congestion occurs) in HetVNET using a supervised machine learning classification method;
- Second, a centralized and dynamic cloud-fog-based intelligent congestion prediction architecture of HetVNET is proposed;
- Furthermore, the proposed congestion prediction and avoidance methods provide stability in the network performance.

We will show that the main achievements, including these contributions, are the precision and novelty of the proposed HetVNET congestion classification approach in an intelligent cloud-fog-based architecture, which is applicable in various vehicular 5G and beyond-based scenarios.

The remainder of this paper is organized as follows. Section 3.2 presents related work. Section 3.3 describes the methodology and classification model. Data collection and performance evaluation are presented and discussed in Section 3.4, and Section 3.5 concludes the study and introduces future work.

3.2 Related Work

The lack of use of intelligent methods in the case of congestion avoidance and control in vehicular networks and, more specifically, in HetVNET is evident in the current literature [24, 29, 33, 77]. The authors in [29] proposed a congestion game to avoid congestion based on scheduling the required services for safety-related applications in HetVNETs. In [23], the authors used a clustering technique as an unsupervised machine learning method for controlling congestion in a Vehicular Ad hoc Network (VANET). In this method, named "Machine Learning Congestion Control (ML-CC)", the k-means technique was applied to cluster messages based on the size, type and validity of the messages. Then, ML-CC assigned appropriate values for the Content Window (CW), data transmission rate, Arbitration Interframe Spacing (AIFS) and transmission range to each cluster of messages. In this method, a Road-Side Unit (RSU) should cluster all of the generated messages and set the values at the same time. It could be difficult to perform these tasks before time-out of the messages, considering the high dynamicity of the network, the increasing number of vehicles, and the large number of generated messages. This issue negatively affects the performance of the network in high density scenarios, and could make an unstable network. In [78], the proposed "Dynamic Congestion Control Scheme (DCCS)" is based on the channel usage level and the amount of CW. The authors considered three levels of 30%, 70% and 90% for channel occupation. Then, based on these three thresholds, the value of CW decreases (for the channel busy level of 30%) or increases (when 70% or 90% of the channel is occupied). In [24], congestion avoidance strategies were executed without prediction. Even if no congestion occurs, message priority and message scheduling will run by default during no-pick time, as well as when the traffic density is low. In [29], the authors proposed an architecture built on SDN and the concept of edges as a service to solve a congestion problem with no intelligence mechanism for congestion prediction. The proposed prediction in [29] is mainly based on the pattern of user demands during different times of the day. The Internet Service Provider (ISP) provides clients with the required resources based on the pattern. Therefore, in some intervals during a day, the demand for resources can be higher than other times of the day; thus, the ISP will then adjust the resource allocation to maintain the network performance at an excellent level. The results show that the authors could improve the quality of the service and propose an efficient mechanism in resource allocation. Nevertheless, an intelligent method could significantly improve the performance of the proposed mechanism. In [33], the authors worked on a prediction method for controlling congestion in VANETs. They proposed a new adaptation method for the transmission power and data rate based on vehicle density prediction. However, the authors did not apply intelligent methods in the proposed prediction method and relied solely on the information they received from the vehicles in front of the targeted vehicles. This prediction is not accurate for scenarios in which there is a malicious vehicle node in front of the targeted vehicles. In [79], the authors proposed a dynamic vehicle clustering mechanism based on the estimation of the network density and the speed of the vehicles to avoid congestion in VANETs. They could use a deep learning regression method to predict the density and speed of the vehicles. Vehicle density estimation was used in [34] to propose an approach for controlling congestion using dynamic transmission power control. In [80] a predictive control model was used by a control agent to define the optimum transmission rate for vehicle nodes in vehicular networks. Prediction is a major task of machine learning methods, and it is not applied in the proposed predictive control model in [80].

Moreover, the number of proposed methods for controlling congestion problems using a fog computing-based architecture is very low in the current literature on vehicular networks.

In [20], the results show good efficiency, high packet delivery, and a low channel busy ratio. Vehicles in decentralized congestion controlling mechanisms must monitor and analyze a large number of messages to detect and control congestion with a low delay (much less than 500 ms, which is the human reaction time). Therefore, in such a decentralized approach, too many computations must be performed by the vehicles using the information of each beacon received from the surrounding vehicles. Most safety services even need less than 100 ms of latency; for example, the maximum latency in precrash warning services is 50 ms [81]. Therefore, an emergency safety message must receive with a delay lower than 50 ms; otherwise, vehicular networks and applying safety applications could not do anything to save a human life, especially in the presence of road hazards. In decentralized congestion controlling mechanisms, vehicles must monitor and analyze a large number of messages to detect and control congestion with a delay of less than 50 ms. Moreover, distributed methods require high vehicle cooperation. Exchanging a substantial number of messages between the vehicles causes overhead and significant delay. In the case of low cooperation among vehicles, the delay increases even more. In addition, the calculations needed to find the closest and furthest ahead and behind vehicles must be done within a limited period of time. Having a time restriction for running multiple computations is therefore a challenging task for the proposed method in [20]. These challenges exist in all of the proposed decentralized (or distributed) congestion control mechanisms, such as [20] and [22]. In [22], based on the proposed distributed approach, all of the calculations (especially for predicting the value of the utility function, which makes use of the Markov chain method) require computational resources and are time-restricted for a vehicle. Since the information changes dynamically and quickly in vehicular networks, calculations must be made before a new information update is received, which is a major task for vehicles in a short period of time. In [82], data offloading from vehicles to infrastructure was proposed to control congestion in an SDN-based vehicular network environment. The authors used a controller to make decisions on offloading the data load from vehicles to each of the RSUs or Base Stations (BSs) of the cellular network. They could use a fog device as a controller to locally manage the data offloading process.

Network performance and Quality of Service (QoS) metrics are critical in HetVNET. These metrics are highly related to the network congestion levels. If we consider two levels of safe (no congestion) and congestion for data flow in HetVNET [83], then the network performance and QoS will drop when the network data flow shifts from safe to congestion level. The approach that will be explained in the next parts of this paper is a novel solution to avoid a drop in the network performance and QoS to a low level. In this solution, we define a warning level (before the congestion level) and predict this warning state of the network data flow. An accurate prediction method that uses the computing and storage power of fog devices can locally predict congestion before it occurs in a dynamic HetVNET. Therefore, targeted HetVNETs have time to execute congestion control/avoidance mechanisms (phase two) to prevent congestion. Accordingly, network performance and QoS will remain at an acceptable level.

Considering the discussed issues of instability in the performance of the network by increasing the number of vehicles and applying Artificial Intelligence (AI) methods in HetVNET congestion-related works and the absence of a congestion avoidance mechanism using fog computing technology in HetVNET-related literature, we propose a novel approach to predict congestion warnings using a supervised machine learning classification method in a centralized and dynamic cloudy-fog-based architecture.

3.3 Methodology and Classification Model

Congestion in the network leads to a reduction in the data delivery ratio. This metric is considered in vehicular network congestion-related work to detect congestion [1,38]. However, packet loss could also accrue due to weak signals. It is necessary to be certain that congestion is the only reason for packet loss. Therefore, in this paper, we consider the Data Delivery Ratio (DDR) and Received Signal Strength (RSS) to interpret the congestion situation in HetVNETs.

Moreover, due to the strong capacity of neural networks and deep learning methods to generate complex models, these methods have recently been widely used in a variety of research fields and problems. Deep learning methods are worthwhile in applications to problems that have a high-dimensional dataset that contains enormous amounts of data, while our problem in this paper is not in this category. We therefore decided to use a supervised machine learning classification algorithm.

3.3.1 Classifying the Data Flow

This paper aims to predict the warning state in terms of network congestion in HetVNETs. If we have knowledge about a warning state for a data flow situation, then we can save time by executing an avoidance mechanism to prevent the network situation from attaining a critical state. DDR is the ratio of the amount of data successfully received at destination points to the amount of data sent by source nodes in the network. Therefore, DDR can have a value between zero and one.

RSS is the power of the received signal at the receiver side. The RSS can be measured by adding the transmit power and antenna gain minus the path loss [84]. The value of RSS in network congestion is higher than the value of RSS in situations in which path loss is the cause of packet loss in the network. Therefore, defining a threshold for the value of RSS can be useful for assuring that congestion is the reason for the packet loss. On this basis, if the value of RSS in the received packet is more than a predefined threshold (RSS_{th}), then the packet loss is due to network congestion.

Based on the definition of DDR and RSS, the data flowing warning situation is defined based on three thresholds for the minimum value of DDR (DDR_{minth}), for the maximum value of DDR (DDR_{maxth}), and for RSS (RSS_{th})which are just for the warning state. Accordingly, we define two classes of warnings and nonwarnings in this work:

$$\text{Data flowing classes} = \begin{cases} \text{Warning, if:} \\ \quad DDR_{minth} \leq DDR \leq DDR_{maxth} \\ \; and, \\ \quad RSS_{th} \leq RSS \\ \text{Nonwarning, } \; otherwise \end{cases} \tag{3.1}$$

The amount of DDR_{minth}, DDR_{maxth} and RSS_{th} can be defined by the network management unit (in which DDR_{minth} and $DDR_{maxth} \in (0, 1)$ are not equal). In this way, the network management can change the value of DDR_{minth} and DDR_{maxth} any time and based on the network situation. Thus, this method provides tolerable congestion management approach that can define different congestion warning intervals over time and is based on the network's situation. For example, if the network management unit assigned -96.26 dBm for RSS_{th}, 0.4 as a value for DDR_{minth} and 0.6 as a value for DDR_{maxth} then it is a warning state while the data is flowing, when $DDR \in [0.4, 0.6]$ and the value of RSS is more than -96.26 dBm [38].

3.3.2 Proposing Naive Bayesian Network Congestion Classifier

Naive Bayesian classifier is a powerful ML method for solving current real-world classification problems, such as spam filtering and text classification. This classifier is very fast compared to other classification algorithms. Moreover, it does not necessarily require a large amount of training data for good prediction. Thus, it is widely used in many scientific studies and in research today. Considering Bayes theorem, the Naive Bayesian classifier provides minimum error using independent features [85]. The Naive Bayesian classifier calculates the probability that the hypothesis is true when the given data are used and is called the posterior probability.

In this paper, we consider five parameters, the number of vehicles (v), data transmission rate (dr), DSRC transmission power (tp_{DSRC}), LTE transmission power (tp_{LTE}), and LTE bandwidth (b), to propose a Naive Bayesian classifier. Therefore, $\mathbf{x}= [x_1, x_2, \ldots, x_n]$ is a set that contains n features, which corresponds to $\mathbf{x}= [x_1, x_2, x_3, x_4, x_5] = [v, dr, tp_{DSRC}, tp_{LTE}, b]$ ($n = 5$). Additionally, let us consider $\mathbf{c}= [c_1, c_2, \ldots, c_m]$ to show a set of classes that contains m ($m = 2$) different classes.

We consider two types of classes: w_0, which is a class for no congestion warning in HetVNET, and w_1, which is a class for having congestion warning in HetVNET; hence, here $\mathbf{c}= [c_1, c_2] = [w_0, w_1]$. Therefore, the posterior probability, where class c_i is true using $\mathbf{x}= [v, dr, tp_{DSRC}, tp_{LTE}, b]$, is calculated as follows:

$$P(c_i|x) = \frac{P(x|c_i)P(c_i)}{P(x)} \tag{3.2}$$

The Naive Bayesian algorithm calculates as follows:

$$P(w_0|v, dr, tp_{DSRC}, tp_{LTE}, b) = \frac{P(v|w_0)P(dr|w_0)P(tp_{DSRC}|w_0)P(tp_{LTE}|w_0)P(b|w_0)P(w_0)}{P(v, dr, tp_{DSRC}, tp_{LTE}, b)} \tag{3.3}$$

and

$$P(w_1|v, dr, tp_{DSRC}, tp_{LTE}, b) = \frac{P(v|w_1)P(dr|w_1)P(tp_{DSRC}|w_1)P(tp_{LTE}|w_1)P(b|w_1)P(w_1)}{P(v, dr, tp_{DSRC}, tp_{LTE}, b)} \tag{3.4}$$

where $P(w_0 \mid v, dr, tp_{DSRC}, tp_{LTE}, b)$ is the probability of a no-congestion warning using input data of $\mathbf{x}= [v, dr, tp_{DSRC}, tp_{LTE}, b]$, and $P(w_1 \mid v, dr, tp_{DSRC}, tp_{LTE}, b)$ is the probability that a congestion warning situation is true using input data of $\mathbf{x}= [v, dr, tp_{DSRC}, tp_{LTE}, b]$. Since the value of the prior probability is the same for all given data of the dataset, it can be removed, and (3.2) can be written as:

$$P(c_i|x) \propto P(c_i) \prod_{j=1}^{n} P(x_j|c_i), \tag{3.5}$$

$i = 1, ..., m$.

The Naive Bayesian classifier selects the maximum posterior as output, which is a class with a higher probability of being true. There, if we assume $\hat{y} = c_i$ as output of the Naive Bayesian classifier, then we have following:

$$\hat{y} = argmax P(c_i) \prod_{j=1}^{n} P(x_j|c_i), \tag{3.6}$$

where n and m equal to five and two, respectively.

3.3.3 Centralized and Dynamic Cloudy-fog Intelligent Congestion Prediction Architecture

In a centralized and dynamic cloudy-fog intelligent congestion prediction architecture of HetVNET, as shown in Fig. 3.1, a Fog Congestion Predictor Unit (FCPU) is placed between the cloud and end users like a skillful intermediary, to locally and efficiently predict the warning state in the data flow using a prediction model. A Centralized Management Unit (CMU) is connected to the FCPUs to orchestrate them and make decisions, such as setting the warning interval using (3.1) by defining the values of DDR_{minth}, DDR_{maxth} and RSS_{th}. Therefore, Fig. 3.1 shows a centralized and intelligent architecture in which FCPUs locally and dynamically analyze data that came from vehicles and BSs. In this cloudy-fog architecture, there are five types of connections, as follows:

- Cloud2fog: communication between a cloud and a fog device (FCPU);
- CMU2fog: communication between CMU and a fog device (FCPU);
- Fog2I: communication between a fog device (FCPU) and an infrastructure such as the BS of the cellular network;
- V2I: communication between a vehicle and a BS, using LTE;
- V2V: communication between two vehicles, using DSRC.

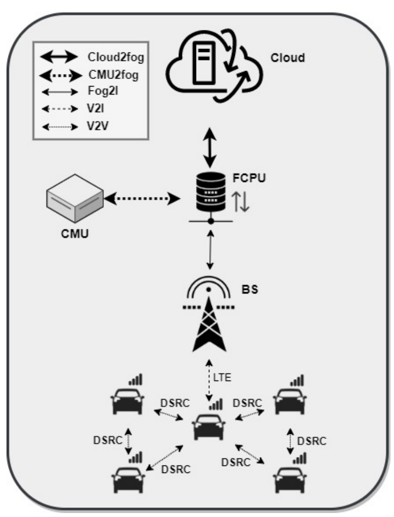

Figure 3.1 A centralized, dynamic and intelligent cloudy-fog congestion prediction architecture of HetVNET.

As Fig. 3.2 shows, each of the FCPUs is connected to the CMU, other FCPUs and cloud. The CMU is responsible for the following tasks:

- Defining the size of the segments, the amount of Δt and j, and the value of RSS_{th}, DDR_{minth} and DDR_{maxth} to be used by the FCPU;
- Assigning segments and BSs to the FCPUs.

According to Fig. 3.2, we divided the street area into several segments with equal lengths of r meters, and an FCPU was assigned to a maximum number of segments s, where $s \geq 1$. In addition, we assumed that we had z FCPUs with $z \geq 1$. For each vehicle such as v in a segment, the corresponding FCPU estimates the distance of vehicle v to a location in the next time unit such as t using the following formula:

$$D_{(v)} = \frac{1}{2}a_{(v)}t^2 + q_{(v)}t, \tag{3.7}$$

where $q_{(v)}$ is the velocity of vehicle v, $a_{(v)}$ is the acceleration of v, and $D_{(v)}$ is the distance of v to the next location at time t, where $t = j\Delta t$ and $j \geq 1$. Δt has a preliminary amount, and each time, the amount of j will be increased. The preliminary value of Δt and the value of j are defined by CMU. For example, if $\Delta t = t_1$ and $j = \{1, 2, 3, ..., N\}$, then for $j = 1$ and for each vehicle, FCPU calculates the value of $D_{(v)}$ with $t = t_1$, and the next time, FCPU calculates the value of $D_{(v)}$ with $t = 2t_1$, and so on. Fig. 3.3 shows how predicting a warning state of the network can save time for executing congestion control mechanisms and preventing congestion in the network. The proposed vision in a highly dynamic network type such as HetVNET (and any other type of vehicular network) can help the network management system to have a dynamic and tolerable solution for any future challenge in the network.

Since the estimation of $D_{(v)}$ is the distance to a location where v will reach at a future time (next t) and we do not have information of $a_{(v)}$ and $q_{(v)}$ during the next time t, the FCPU considers the average of both $a_{(v)}$ and $q_{(v)}$ from the previous time t. The FCPU uses $D_{(v)}$ and the length of each segment (r meters) to estimate the corresponding segment that v will reach in the next time t. FCPU can estimate the number of vehicles in each segment located in its coverage area. Therefore, the vehicle densities of the segments at the future time t are estimated by the corresponding FCPU. As an example, Fig. 3.2 depicts the case when FPCU 1 estimates that the red vehicle in segment 2 (blue dashed vertical line) will leave the coverage area of FCPU 1 and arrive at segment 1 in FCPU 2 (green dashed vertical line); then, FCPU 1 sends the location information to FCPU 2. Therefore, the number of vehicles in a segment at future time t is estimated using the number of vehicles estimated by the corresponding FCPU plus the number of vehicles estimated by a neighbor FCPU (if applicable). Moreover, whenever a vehicle leaves an FCPU coverage area, the velocity and acceleration during the last time t must be sent by the FCPU to the new FCPU.

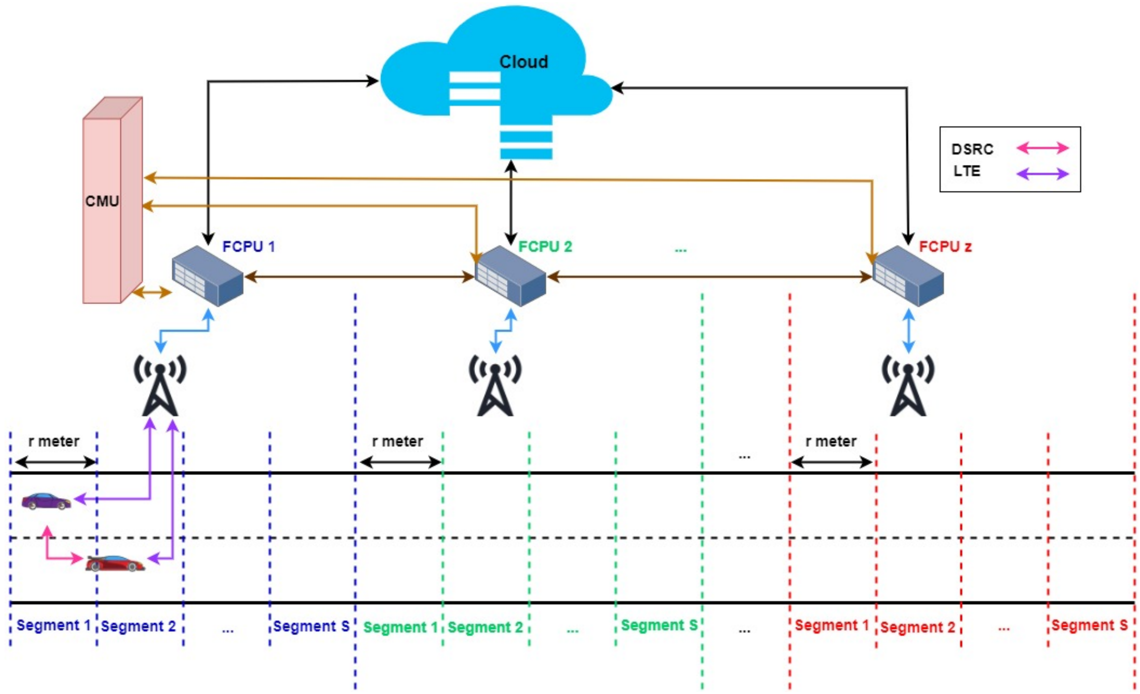

Figure 3.2 Illustration of intelligent congestion prediction architecture of HetVNET.

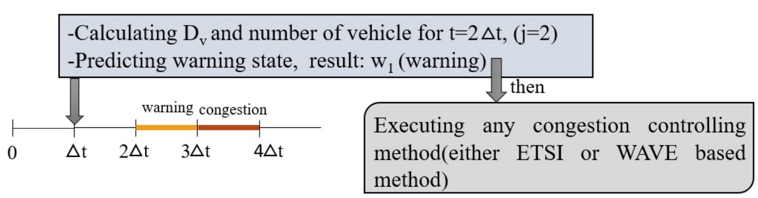

Figure 3.3 Congestion warning state in a HetVNET.

The traffic situation in such a highly dynamic environment as HetVNET is changing fast and can even vary substantially between segments of FCPUs. This variation can be due to the specific conditions in the segments, such as holding special events during rush time and locating high demand places such as hospitals or airports in the segments. For any change in the value of v, which is the number of vehicles in a segment, the corresponding FCPU must perform Naive Bayesian congestion prediction.

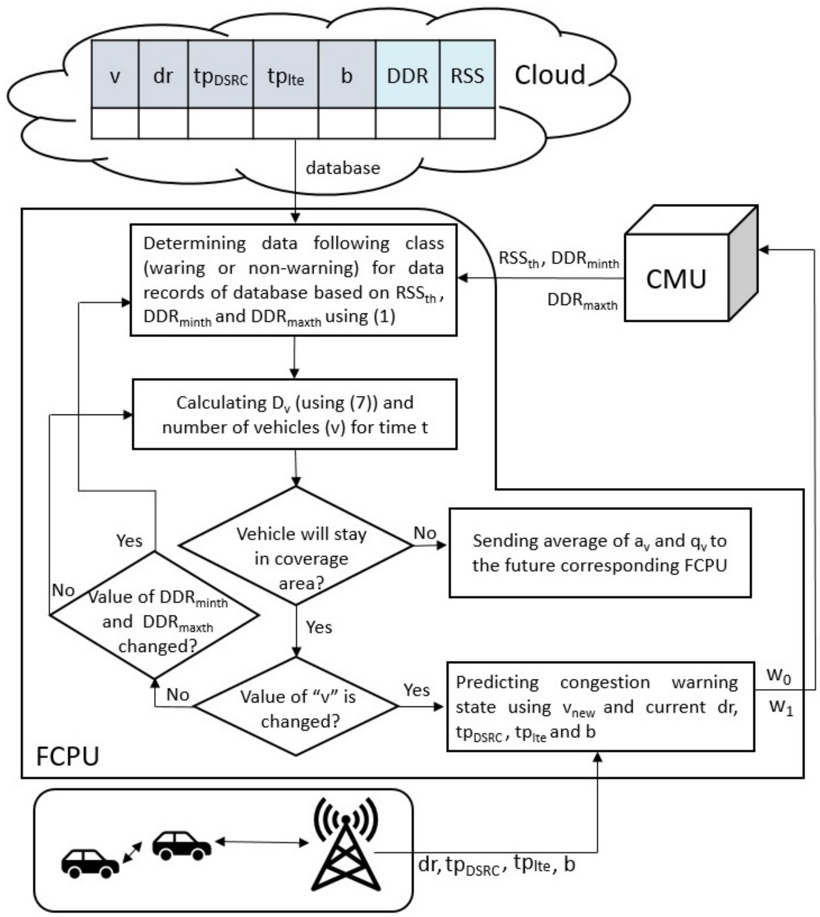

Figure 3.4 A Flowchart of main steps in FCPU.

To apply the proposed Naive Bayesian classifier in the centralized and dynamic cloudy-fog-based architecture of HetVNET, as shown in Fig. 3.4, a reference database is needed, which is prepared at the cloud level. In this approach, information about five considered features, such as the number of vehicles v, dr, tp_{DSRC}, tp_{LTE}, and b, must be gathered, each as a record in a database. For each data record, DDR and RSS have been calculated. This database can be updated and matures with time. First, the FCPU receives such a reference database from the cloud and stores it. Then, based on the value of DDR and RSS and using DDR_{minth}, DDR_{maxth}, RSS_{th} and (1), each data record obtains a class of w_0 or w_1. Afterward, the FCPU calculates $D_{(v)}$ using (3.7) and locally estimates v for each segment at future time t. In the case of an update in the value of v, the FCPU calculates $P(w_0|v_{new}, dr, tp_{DSRC}, tp_{LTE}, b)$ and $P(w_1|v_{new}, dr, tp_{DSRC}, tp_{LTE}, b)$ using the updated value of v, the current value of dr, tp_{DSRC}, tp_{LTE}, b and the database. Then, based on the computation results, it predicts warning or nonwarning state for data flowing in the target HetVNET at a future time t. In Algorithm 1, the pseudocode of the proposed Naive Bayesian classifier algorithm in an FPCU is presented. The values of P_{w_0} and P_{w_1} can be calculated using (3.3) and (3.4). Note that the task of the FCPU is to predict the warning or nonwarning state of HetVNET based on the data of five parameters: v_{new}, dr, tp_{DSRC}, tp_{LTE}, b. By this approach, we can implicitly infer that independent variables such as v_{new}, dr, tp_{DSRC}, tp_{LTE}, and b have an effect on the value of DDR and consequently mitigate or intensify the network congestion state of HetVNET.

Algorithm 1 Naive Bayesian network congestion classifier in a FCPU

```
Input 1: a reference database generated at cloud level and contains values of v, dr,
tp_DSRC, tp_LTE and b as features, values of DDR and RSS, and "warning state" as output.
For each data record, the output column can have a value of w_0 as a nonwarning or w_1
as a warning state (based on DRR and RSS, using (3.1)).
Information collection locally as x= [v, dr, tp_DSRC, tp_LTE, b] form HetVNET.
Calculate D_v and v_new.
if v_new ≠ v then
    Input 2: x_updated= [v_new, dr, tp_DSRC, tp_LTE, b]
    Based on Input 1 and using Input 2, calculate:
    P_w0 = P(w_0|v_new, dr, tp_DSRC, tp_LTE, b)
    P_w1 = P(w_1|v_new, dr, tp_DSRC, tp_LTE, b)
    if P_w0 < P_w1 then
        it is a warning state.
    else
        it is a nonwarning state.
    end if
end if
```

In the centralized cloudy-fog architecture, we consider LTE for V2I communications. Therefore, the required information is exchanged between vehicles and FCPU using LTE BSs. Large coverage and high downlink and uplink capacity are the advantages of the LTE [2], which help to provide requirements for necessary data transmission in the proposed approach. However, if the proposed congestion classification method is applied in a decentralized system, then the vehicles should employ the Naive Bayesian network congestion prediction method. In this case, and similar to most decentralized methods, network can encounter high overhead and delay. Applying the edge computing concept by clustering the vehicles and selecting cluster heads as edge nodes, can be a potential solution. The edge nodes are responsible for gathering and analyzing data, running prediction functions, and distributing the result. To select the cluster head vehicle, several metrics, such as available computing and storage resource capacity, communication reliability and accessibility (in terms of distance to another vehicle in the cluster), can be considered.

3.3.4 Advantages and Challenges of the Proposed Centralized Architecture

The proposed cloudy-fog architecture is compatible with current standards. ETSI applies Decentralized Congestion Control (DCC) in the Media Access Control (MAC) layer. DCC is a state machine-based approach that switches between three states of relax, active and restrictive based on the channel load. Most DCC-based algorithms, such as the Linear Message Rate Integrated Control (LIMERIC) [44], Dual-α DCC [86], and Dynamic Beaconing (DnyB) protocol [87], depend on the value of the Channel Busy Ratio (CBR). Based on the literature, these algorithms have the challenge of finding and setting optimum values of the parameters [86]. An optimal value for the CBR threshold can prevent the underutilization of the channel [32]. Thus, applying the congestion prediction method instead of calculating the current channel busy level can help the network management system develop policies for using the channel to prevent congestion from occurring in the future. In other words, the predicted warning state in terms of the congestion problem can be a complementary feature for dynamic and tolerant network congestion management. For example, based on the result of the proposed Naive Bayes congestion prediction, if we have a congestion warning in the area covered by an FCPU at time t, the congestion control algorithm can switch between the states and change the value of the data rate in a such manner that there will be no congestion problem in the future.

Based on the literature, network congestion in VANET has been considered more during the past decade, including cross-layer approaches, event driven and priority-based approaches, topology-based approaches, and dynamic and adaptive approaches [1]. In the proposed WAVE based congestion control algorithms, the solution part can be applied when a warning state is predicted by the proposed Naive Bayesian congestion prediction method.

Therefore, the network management system will have one eye on the present and one eye on the future by using the proposed congestion prediction result and creating policies and applying them at the current time with the aim of avoiding congestion in the future.

Moreover, using multihop strategy instead of the proposed architecture to send traffic states has other challenges. First, in multihop methods, the distance between the nodes has a direct effect on the delay. Applying n-hop communication to transfer the data flow state to a far node increases the delay in the network. In addition, there is a risk of unsuccessful data delivery in multihop strategy due to fragile communication links between those nodes that are in a long way from each other. Furthermore, multihop communication increases the overhead for the middle vehicles. The nodes must apply an algorithm to choose the best next hop. In addition, implementing, upgrading, and debugging centralized methods are easier than decentralized methods.

On the other hand, the centralized system should have fault tolerance ability. Otherwise, with any fault in the system, it will crash. In case of failure in the system, the supporting (back up) strategy must handle the situation and prevent crashing the system. In the proposed centralized method, the centralized system contains cloud, FCPUs, and CMU. Therefore, we can have three possible failures:

- Failure in communication with the cloud: In this case, FCPUs can use the last reference database until the problem is solved.
- CMU Failure: In this case, the last update for the values of Δt, j, DDR_{minth}, DDR_{maxth} and RSS_{th} from CMU can be used until the problem is solved. Additionally, the last assignment of segments and BSs to FCPUs can be applied until the CMU can join the system again.
- FCPU failure: In this case, the CMU can assign the coverage area of the failed FCPU to other neighboring FCPUs until recovering the FCPU failure.

In addition to these suggested backup strategies, improving the fault tolerance ability of the centralized methods should be investigated more in the future.

Moreover, the part of assigning segments and BSs to FCPUs can be studied in the future to find the optimum solution, especially in complex urban environments. For example, in the most crowded parts of a city, it can be better to consider a low number of segments for FCPUs to cut down the load of the FCPUs and share it among a greater number of FCPUs. In this scenario, communication between a BS and more than one FCPU should be considered since it is possible to assign a BS to several FCPUs.

3.4 Data Collection and Performance Evaluation

3.4.1 Data Collection and Simulation

The lack of datasets containing HetVNET information was the reason why we generated a dataset using HetVNET simulation scenarios. Since we generated the dataset and we did not have a large amount of data (a limitation in our work), we could not consider the deep learning methods. Moreover, complex prediction methods are not necessarily the best choice to use, and depending on the conditions of the problem, we might obtain better results with simpler and faster methods such as ML prediction methods. Therefore, we study supervised ML classification methods. Nevertheless, the proposed centralized and dynamic cloudy-fog based architecture is compatible with more complex prediction methods such as deep learning algorithms. Indeed, the computation and storage power of FCPUs are suitable for executing more complex prediction methods.

The dataset contains data records of five mentioned parameters, which are effective in network congestion problems. We generate our data using the Simulation of Urban Mobility (SUMO) 0.26.0 [88] simulator and the Veins LTE version 1.3 [89], both in Linux (Ubuntu 16.04). The boroughs of Montreal city in Canada are considered for simulating vehicular traffic and heterogeneous network environment. “OpenStreetMap” [90] is used to extract the map data related to a part of Montreal as an “.osm” file. SUMO is used to generate urban vehicular traffic, and Veins LTE is simultaneously used as a network simulator. Vehicles are equipped with both LTE and IEEE 802.11p interfaces. DSRC is used to exchange intragroup vehicle information. LTE is used to exchange information on inter-groups of vehicles. Moreover, an accident is defined to occur at a specific time (t= 70 s) when running the simulation scenario to generate extra load of data. The duration of each run is 1000 s. The minimum path loss coefficient is 2 [91] in the simulation scenarios. DCC (used in ETSI) is based on changing the value of the data transmission rate and transmission power [32,46,86]. Additionally, most of the proposed congestion controls in WAVE standard are based on adapting the transmission power and data transmission rate [35,35,38,41]. Therefore, in each run, we changed the value of v, dr, tp_{DSRC}, tp_{LTE} and b according to Table 3.1, and we calculated the values of DDR and RSS. The values of DDR_{minth} and DDR_{maxth} are 0.4 and 0.6, respectively [92]. The amounts of generated and transmitted data (during 1000 s of running a simulation scenario) are used to calculate the DDR.

In [38], the authors proposed an RSS cutoff value for V2V communication, in which if the RSS is higher than the cutoff value, then the packet loss is due to network congestion. They provide simulation results and technical discussions to support this issue. In [38], the threshold value for RSS is -96.26 dBm for data rates of 3, 6 and 12 Mbps. On this basis, the value of RSS_{th} is -96.26 dBm in this paper. Based on the amount of DDR and RSS in each simulation scenario and using (3.1), each data record belongs to a warning or nonwarning class. In other words, among the simulated data gathered in the dataset, the data records that their DDR value is in a warning range and the RSS value is greater than a threshold value such as -96.26 dBm were labeled by w_1 as a warning state.

For the implementation, we used Python version 3.6 with well-known libraries, such as Scikit-learn, NumPy, Matplotlib, Pandas, and others, to generate the proposed Naive Bayesian network congestion classifier and evaluate and compare its performance with the performance of the Support Vector Machine (SVM), K Nearest Neighbor (KNN), and Random Forest using the same data. Moreover, normalization is performed on the data since the data extracted from the simulation scenario vary in unit and range. Normalizing data helps with accurate prediction models. In addition, the training dataset is balanced.

Table 3.1 Parameters and corresponding values used in the simulation scenario.

Parameter	**Value**
Bandwidth (IEEE802.11p)	10 MHz
Bandwidth (LTE)	5 MHz, 10 MHz, 20 MHz
Transmission power (IEEE802.11p)	30 dBm (Maximally)
Transmission power (LTE)	43 dBm, 46 dBm
Transmission rate (IEEE802.11p)	3 Mbps, 6 Mbps, 12 Mbps
Resource Blocks size	25, 50, 100
Minimum path loss coefficient	2
Message size	400 Bytes
Number of base station (eNB)	1
Simulation area	1000 m $\times$ 1000 m
Number of lanes	4 (two in each direction)
Simulation time	1000 s
Number of vehicles	30, 50, 100, 150, 200
Vehicles speed	0-40 km/h
Propagation model	Nakagami (m=3)
Simulation runs	260

3.4.2 Performance Evaluation of the Congestion Classification Method

Table 3.2 is prepared to clarify the relationship between the actual and predicted classes [93].

If our target HetVNET is in a warning congestion situation but the predicted result incorrectly shows a nonwarning state that is introduced as False Negative (FN) in Table 3.2, then it will have undesirable and unexpected consequences for vehicular users. Therefore, the cost of FN prediction in our proposed problem is higher than the cost of False Positive (FP). In the latter case, the actual state of congestion in the network is nonwarning, but it is predicted as a warning case. Although this case is a fault in the performance of the proposed prediction model, vehicular users do not experience the result of the side-effects of this error as much as the bad consequences from FNs. Regarding this issue, the recall factor helps us to evaluate the proposed prediction classification model more efficiently. High recall values show that most of the warning cases are correctly predicted and that the number of warning states that are incorrectly predicted as a nonwarning state is low. Precision considers only positive predictions, both those that are truly predicted warning state (TP) and those that are falsely predicted warning state (FP). Therefore, for the proposed problem in this work, the recall factor is more important than the precision because the costs of FP and FN vary for vehicular users. The False Positive Rate (FPR), which indicates the ratio of states, accounts for the warning states but does so incorrectly. If it is not a warning case and it is predicted truly, then we have a True Negative (TN) in our results. The True Negative Rate (TNR) is called the specificity. The FPR is $1 - specificity$ [94].

Regarding the above-mentioned discussion, we evaluated the performance of the proposed Naive Bayesian classifier.

Table 3.2 Relationship between the actual and predicted classes.

		Predicted class	
		$Classw_1$	$Classw_0$
Actual Classes	$Classw_1$	True Positive (TP)	False Negative (FN)
	$Classw_0$	False Positive (FP)	True Negative (TN)

We used Receiver Operating Characteristics (ROC), which is a common graphical tool, to measure the performance of binary classifiers and the Area Under the Curve (AUC).

As shown in Figs. 3.5 and 3.6, the ROC curve plots the True Positive Rate (TPR), which is the recall against the FPR for binary classification models. In the ROC curve, the x axis shows the FPR, and the y axis illustrates the TPR. Each of the TPR and FPR can be equal to a value in [0,1]. In a Roc curve, when both the TPR and FPR are zero (i.e., (0,0)), it indicated that the classification model predicts negative output in every prediction. Therefore, this outcome indicates in our problem that the prediction model will predict nonwarning state for every input data of x. Indeed, such a prediction model is useless since its performance means that there is no warning state at any time. Thus, a nonwarning state can be considered regardless of the value of the predicted variables every time. In other words, there is no warning at all about the congestion that makes us worry. We know, however, that this circumstance is not true in the real world. On the other hand, when the TPR and FPR are equal to one (i.e., (1,1)), this case designates that the prediction model predicts positive for every input data of x, regardless of whether it is truly predicted or not. In other words, the probability of true positives and the probability of false positives are the same. If the model predicts the warning state for every input data point, with a probability of 0.5, it is correct, and with a probability of 0.5, it is false. The diagonal line that connects the two points (0,0) and (1,1) shows a random classifier at which the probability of truly predicting a warning state is equal to the probability of falsely predicting it. The AUC in a random classifier model is 0.5 [94].

Regarding the time sensitivity of the problem, we choose machine learning methods that are stable and accurate but not complex. High levels of complexity in the methods mean more time for training and predicting. Therefore, we apply SVM, KNN, Random Forest and Naive Bayesian algorithms to the data using k-fold cross validation technique [95], with k=10. The entire data is divided into 10 subsets or folds. We considered one of the folds as the test data and the other 9 folds as the training data. Then, the classification algorithm uses training data to generate the model. Afterward, the performance of the generated model is evaluated using the test fold. At this step, the ROC curve was plotted, and the AUC was computed. We iterate this procedure 10 times, and in each round, one of the 10-fold is selected as the test fold and the other remaining 9 subsets as the training folds. Therefore, every fold was considered a test subset one time. As mentioned above, the ROC curve is a graph used to illustrate TPR and FPR, and then, after 10 repetitions of the procedure, the mean ROC curve shows the average performance of the model during the K=10 iterations in terms of the TPR and FPR. Figs 3.5 and 3.6 show the mean ROC curves of the 10 folds along with AUCs for SVM, KNN, Random Forest and Naive Bayesian classifiers. In these figures, the dotted lines show the ROC curves of the 10 fold. The black diagonal dashed line shows the random classifier. The colored area around the mean ROC illustrates the variance around the mean ROC. The variance area indicates confidence intervals of the models. The variance area, mean ROC curve and its AUC help us to comprehend the stability of the classification models. A perfect classifier has a ROC curve far from the diagonal line toward its upper left side with an AUC value equal to one [94].

In Fig. 3.5, SVM and KNN congestion warning classifiers are compared with each other. From this figure, in terms of having a higher AUC of the mean ROC, the KNN congestion warning classifier shows better performance than the SVM. In addition, the variance area (red light area) around the ROC mean curve for the KNN congestion warning classifier is smaller than the variance area for SVM (green area), which indicates that the predicting behavior of the KNN congestion warning classifier is more stable than that of the SVM congestion warning classifier. Although KNN performs better than SVM in predicting congestion warning states of HetVNET, its performance is slightly weak compared to the Random Forest classifier, as shown by the AUC value in Fig. 3.6. However, from the variance area of KNN in Fig. 3.5 and Random Forest (pink area) in Fig. 3.6, it is likely that the KNN congestion warning classifier is more stable than the Random Forest congestion warning classifier, with larger variance area. Due to the Random Forest progression mechanism, which is based on extending the tree randomly, the algorithm is less stable than the KNN and Naive Bayes classifiers.

The farthest point from the random classifier is at the top-left corner of the ROC curve plot, where the recall is one and the FPR is zero. When the FPR is zero, $1 - specificity$

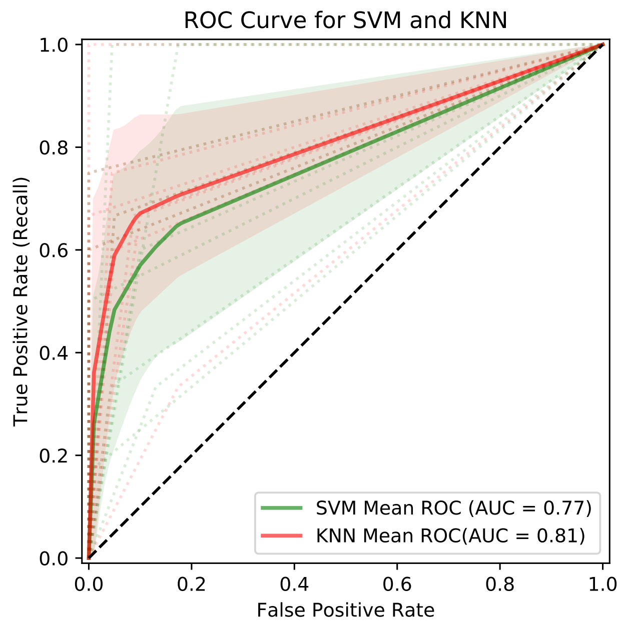

Figure 3.5 ROC curve for SVM and KNN congestion warning classifiers of HetVNET.

equals zero; consequently, the specificity is one. Therefore, at this point of the curve (top-left corner), both the recall and specificity have their best values that can be obtained, and in this case it is one. The blue line in Fig. 3.6 shows the mean ROC curve of the Naive Bayesian congestion warning classifier. As illustrated in the figure, among the four classifiers, the mean ROC curve of Naive Bayesian classifier is farther from the random classifier and closer to the top-left corner compared to the mean ROC curves of Random Forest, KNN and SVM (using Fig. 3.5). As a result, the Naive Bayesian congestion warning classifier has the highest AUC value of 0.94 compared to SVM, KNN, and Random Forest with AUC values of 0.77, 0.81, and 0.82, respectively. In addition, in Fig. 3.6, the small light blue area around the mean ROC of the Naive Bayesian classifier demonstrates its stability, which is more than the SVM, KNN, and Random Forest classifiers.

Table 3.3 Confusion parameters.

	Mean Accuracy (%)	**Mean Precision**	**Mean F1**
SVM	88.01	0.827	0.767
KNN	89.95	0.848	0.821
Random Forest	90.00	**0.856**	0.826
Naive Bayes	**91.87**	0.848	**0.875**

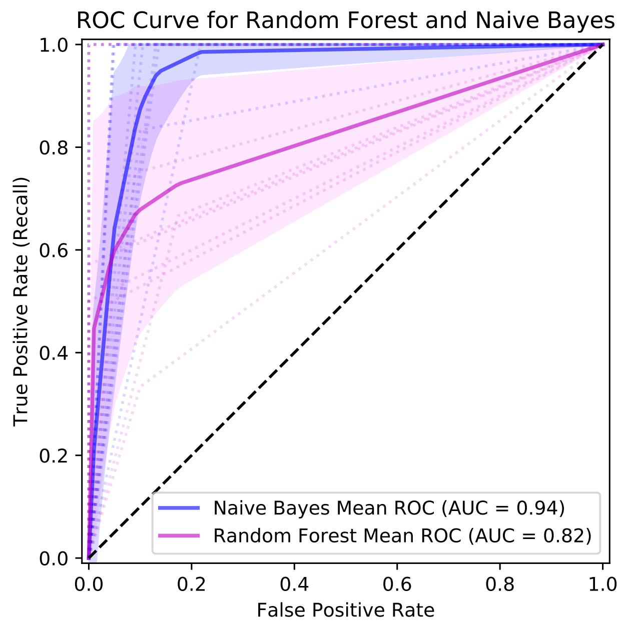

Figure 3.6 ROC curve for Random Forest and Naive Bayesian congestion warning classifiers of HetVNET.

In classification-related problems, other metrics, such as the accuracy, precision and F1 score are used together with the recall and ROC curve. We also evaluated the performance of the proposed Naive Bayesian congestion warning classifier in terms of the mentioned metrics. For each of the 10 folds in the SVM, KNN, Random Forest, and Naive Bayesian congestion warning classifiers, the accuracy, precision, and F1 score are calculated, and then, the average values of the 10 folds belonging to each metric are listed in Table 3.3. The proposed Naive Bayesian congestion warning classifier with a mean accuracy value of 91.87% is more accurate than the SVM, KNN and Random Forest classifiers. In terms of the mean precision, the KNN, Random Forest, and Naive Bayesian classifiers have close values, and SVM has the least mean precision value. The merging of the mean recall and the mean precision gives the mean F1 score. Indeed, the F1 score is the weighted harmonic mean of the recall and precision. The best value for the F1 score is one, which signifies high precision and recall. With a mean F1 score value of 0.875, the Naive Bayesian congestion warning classifier shows better performance than the SVM, KNN, and Random Forest classifiers. The obtained results in Table 3.3 affirm that the performance of the proposed Naive Bayesian congestion warning classifier in almost all of the mentioned parameters is better than that of the other three classifiers.

CPU time is another metric that must be considered in machine learning related works. If a method has good performance in terms of accuracy but requires high CPU time for processing, this trend could be a significant weak point for that method, especially in time sensitive problems. Therefore, we evaluate the performance of SVM, KNN, Naive Bayes and Random Forest classifiers in terms of CPU time in microseconds. As shown in Fig. 3.7, the CPU requires more time to execute KNN, SVM, and Random Forest, respectively than the Naive Bayesian classifier. KNN needs time to calculate the distance between new data and each existing data record. Accordingly, the CPU time for the KNN classifier is higher than that of the other classifiers (using Fig. 3.7). In contrast, Naive Bayes does not require a large dataset for estimations. Moreover, it assumes that the predictors are independent. Therefore, as correctly shown in Fig. 3.7, Naive Bayes is a faster learner classifier than SVM, KNN, and Random Forest.

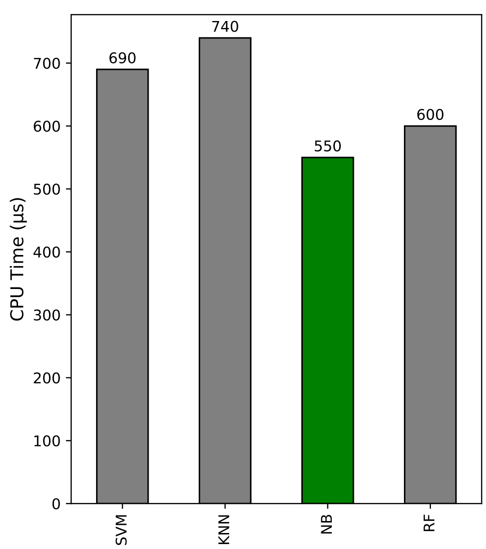

Figure 3.7 CPU time in microseconds for four classifiers.

Finally, considering the discussion about the results related to mean ROC curves, which are demonstrated by Figs. 3.5 and 3.6, the obtained performance results in Table 3.3 and the required CPU time indicate that the proposed Naive Bayesian congestion warning classifier could accurately predict the network congestion warning state in a target HetVNET.

Unfortunately, similar work could not be found in the HetVNET-related literature to make a comparison between the proposed Naive Bayesian congestion classifier and the legitimate benchmark or state-of-the-art. This issue confirms the novelty of this work. Therefore, we compared the Naive Bayesian congestion classifier with three other well-known and powerful supervised classification algorithms, SVM, KNN, and Random Forest.

3.4.3 Performance Analysis of the Proposed Approach

To show how the proposed congestion classification approach positively affects the data transmission in the network, we perform a controlling mechanism named Centralized Network Congestion Classification (CNCC), when a warning result is made by the prediction model. In this mechanism, in nonwarning situation, the value of CW is 15, which is a minimum allowed amount in DSRC, as mentioned in [29, 95], and the data transmission rate is 3 Mbps. This low data rate is selected to prevent noise and interference [38]. Moreover, based on a study presented in [50], in a moderate channel load, data transmission rate of 3 Mbps has a higher average reception rate than other transmission values. In the CNCC mechanism, when the result of the Naive Bayes prediction is a warning state, the value of CW is set to 1023, which is a maximum allowed value [29, 96], and the value of the transmission rate increases. To determine how much the value of the data transmission rate must be increased, we investigate the performance of the CNCC using a range of allowed data transmission rates in DSRC. Figs. 3.8 to 3.10 show the variation in the packet loss ratio, the average throughput and average delay of the CNCC using 3 Mbps, 6 Mbps, and 12 Mbps as the data rate. According to the presented results in Figs. 3.8, 3.9 and 3.10, CNCC outperforms when we applied a 6 Mbps data transmission rate. The aim of increasing the value of the data rate is that the data that might have waited for a while (because of the large value of CW) could be transferred quickly. Based on these figures, with an increase in the number of vehicles that results in a high channel load, a data transmission rate of 6 Mbps is the best selection. According to Figs. 3.8, 3.9 and 3.10, in the dense vehicular environment, applying a higher data transmission rate such as 12 Mbps, could increase noise and interference and have a negative impact on the network performance. Moreover, this circumstance can create a critical network congestion situation because increasing the value of the data rate requires an increase in the transmission power, which can escalate channel collisions in a dense environment.

In CNCC, the result of the Naive Bayes prediction model is announced by the FCPUs to the corresponding vehicles in their range. In a predicted warning case, the vehicles must apply the new values of CW and data rate (CW=1023 and 6 Mbps for data rate) until they receive the new nonwarning result of the prediction from FCPU. Then, the vehicles can apply CW= 15 and 3 Mbps data rate.

In this architecture, the BSs are the gateway nodes that provide the required information for

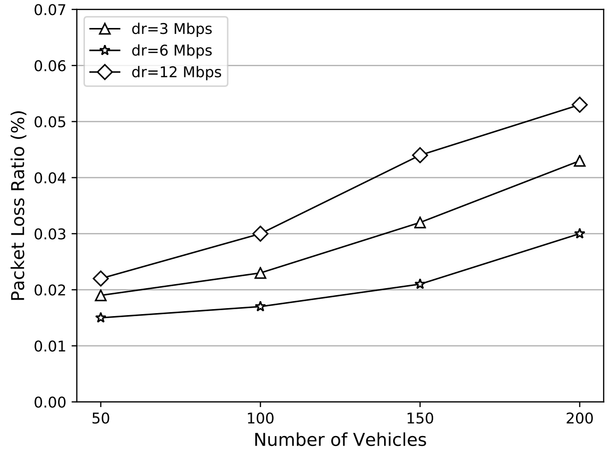

Figure 3.8 Variation in the packet loss ratio using the CNCC mechanism with the numbers of vehicles for various values of the data transmission rate.

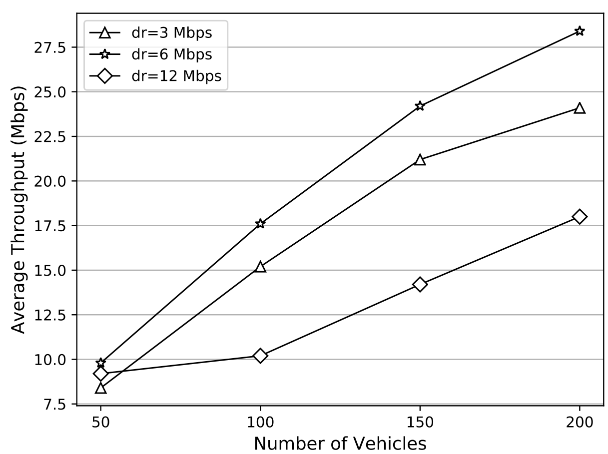

Figure 3.9 Variation of the average throughput using the CNCC mechanism with the numbers of vehicles for various values of the data transmission rate.

the FCPUs and the prediction results for the vehicles. The FCPUs have information on the current values of the predictors, vehicle ID, location, average speed, and average acceleration of every vehicle via gateways. FCPUs are well equipped with enough memory, storage, and processing cores to analyze large amount of data and predict the network congestion state. For example, to implement the proposed ML classification method and make predictions, advanced hardware such as Graphics Processing Unit (GPU) can be employed in FCPUs [97].

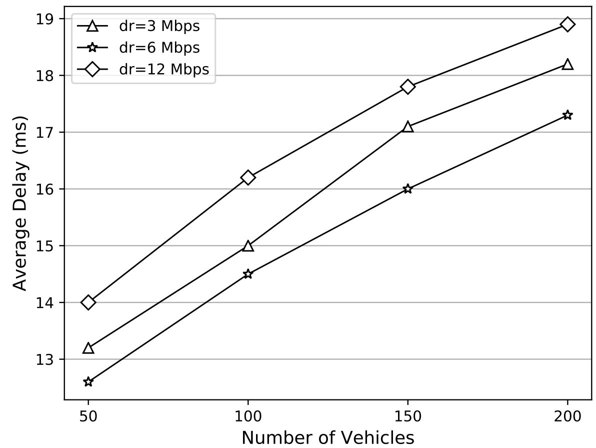

Figure 3.10 Variation in the average delay using the CNCC mechanism with the numbers of vehicles for various values of the data transmission rate.

The FCPU computes $D_{(v)}$ and v_{new} and predicts the congestion state using (3.6). Then, the prediction result must be sent via a gateway node to the vehicles located in the corresponding segment. Based on the prediction result, if the vehicles receive w_1, they apply CW=1023 and dr=6 Mbps to avoid congestion in the network, and if the vehicles receive w_0, there is no need for the vehicles to change the values of the parameters.

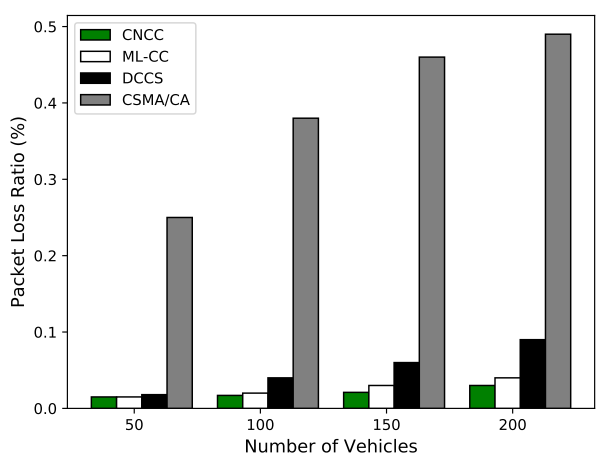

Figure 3.11 Packet loss ratio of the four considered mechanisms with various numbers of vehicles.

In this paper, we compare the performance of the CNCC to contention window-based methods

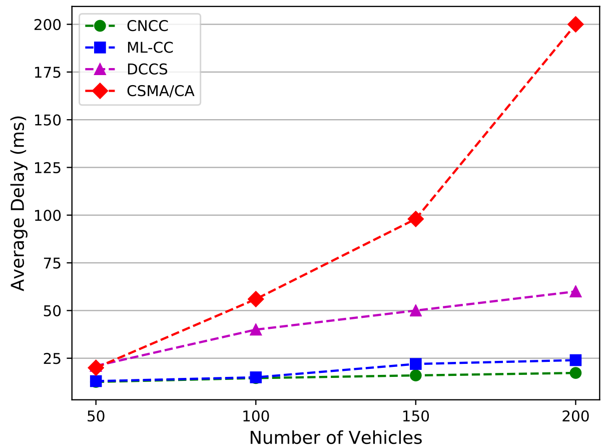

Figure 3.12 Average delay of the four considered mechanisms with various numbers of vehicles.

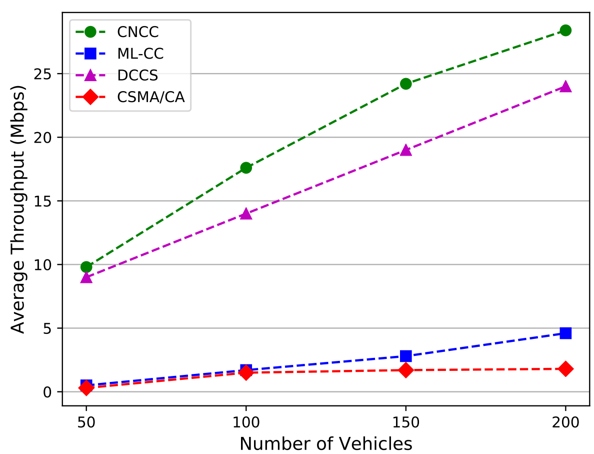

Figure 3.13 Average throughput of the four considered mechanisms with various numbers of vehicles.

such as CSMA/CA, ML-CC and DCCS. The results of the packet loss ratio are presented in Fig. 3.11. Applying CNCC could significantly improve the packet loss ratio compared to CSMA/CA. Moreover, the variation in the value of the packet loss ratio in the CNCC is lower than that in the ML-CC and DCCS with an increasing number of vehicles. Therefore, based on Fig. 3.11, CNCC could improve the packet loss ratio compared to CSMA/CA, ML-CC and DCCS.

Fig. 3.12 shows the results of the average delay (in ms) for the four considered congestion controlling mechanisms. Congestion in the network can increase end-to-end delays in data transmissions. Based on Fig. 3.12, the CNCC could improve the average delay, especially in dense vehicular environments. According to the results shown in Fig. 3.12, the performance of the CNCC in terms of the average delay is much better than that of CSMA/CA and DCCS. Moreover, in comparison to ML-CC, the CNCC could reduce the average delay in scenarios with over 100 vehicles. With an increase in the number of vehicles, the CNCC shows stability in the results that is due to applying the prediction method before congestion occurs in the network.

In Fig. 3.13, the average throughput results of the four considered congestion controlling mechanisms are presented. In a congested network, the amount of average data delivery in seconds is low. Therefore, the results on the average throughput can show how much the mechanisms control congestion in the network. Based on the previous figures, the ML-CC was successful in decreasing the packet loss ratio and the average delay; however, it could not increase the average throughput. As Fig. 3.13 shows, DCCS and CNCC have better performance than ML-CC and CSMA/Ca. In other words, the average amount of successfully received data in a second in the CNCC mechanism is higher than that in the other three methods.

3.5 Conclusions and Future Work

In this paper, we have proposed a centralized and dynamic cloudy-fog-based architecture of HetVNET. Moreover, we have proposed a classification method using a Naive Bayesian algorithm to predict the congestion warning state in the data transmission of HetVNET. The proposed Naive Bayesian classification approach can be applied in the centralized and dynamic cloudy-fog-based architecture of HetVNET, to accurately predict warning situations in data flow. We used the data delivery ratio and the received signal strength as metrics to categorize the congestion warning and nonwarning states in HetVNET. We used five features: the number of vehicles, data rate, DSRC transmission power, LTE transmission power, and LTE bandwidth to predict the congestion warning state of HetVNET. In addition, SVM, KNN, and Random Forest algorithms, which are widely used in current classification problems, have been applied to generate prediction models. Numerical results emphasize that the Naive Bayesian classification approach is not only more suited to the proposed problem but is also more accurate than the other three approaches.

The aim of this approach is to improve the stability in the performance of the network. Employing a congestion prediction model helps us to prepare a network before congestion occurs.

As the results indicate, by applying this approach, we can make a network that is flexible with various vehicle densities and shows stable performance. Based on the obtained simulation results, applying the congestion classification approach could improve the performance of HetVNET in terms of the packet loss ratio, average delay and average throughput.

We will consider the following open challenges as future works:

- Applying the proposed method using real data and evaluating the performance of the method in the real environment of HetVNET;
- Considering other factors, such as the mobility model, modulation technique, complexity of scenarios (urban, rural, straight highway and so on), number of eNBs, and number of resource blocks as predictors to generate a more complex congestion prediction model for HetVNET;
- A Recurrent Neural Network (RNN) method is implemented in real time to analyze the sequential and time series network data of the dataset traces.

CHAPTER 4 ARTICLE 2: A MULTIPLE LINEAR REGRESSION MODEL FOR PREDICTING CONGESTION IN HETEROGENEOUS VEHICULAR NETWORKS

Farnoush Falahatraftar, Samuel Pierre, Steven Chamberland

Mobile Computing and Network Research Laboratory (LARIM)
Department of Computer and Software Engineering, Polytechnique de Montréal
Montreal, QC H3T 1J4, Canada
Email : {farnoush.falahatraftar, samuel.pierre, steven.chamberland}@polymtl.ca.

Status: published in Proceeding of the 16th International Conference on Wireless Mobile Computing, Networking and Communications(WiMob), October 2020.

Abstract

Finite capacity of network resources and enormous data generated by vehicles using safety and comfort applications, have made network congestion a challenge to manage in Heterogeneous Vehicular Network (HetVNET). In this paper, we propose a reliable network congestion model based on a Multiple Linear Regression (MLR), which is a supervised machine learning algorithm to predict network congestion in HetVNET. We have evaluated the performance of our proposed network congestion prediction model using a Cross-Validation test approach. Numerical results show that the proposed linear congestion prediction model is reliable, which can explain and support variability of the response as well. Moreover, we have weighted effectiveness of each considered HetVNET parameters, in association with congestion situation in HetVNET.

Keywords: Heterogeneous Vehicular Networks(HetVNET), network congestion prediction, supervised machine learning method, Intelligent Transportation System (ITS)

4.1 Introduction

In Heterogeneous Vehicular Network (HetVNET) connected vehicle's users can be profited from various services, which are provided by Dedicated Short Range Communication (DSRC) and Long-Term Evolution (LTE) [98]. In vehicular networks, data generated by vehicles can be transmitted via two types of communications: Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I). HetVNET applies DSRC and LTE in V2V and V2I communications respectively [81].

The growing number of smart vehicles and the significant tendency of people to enjoy diverse services from Internet ubiquity generate high resource demands for data transmission through network, which is a notable challenge. In the scenario that a huge volume of data needs to be transmitted, but we do not have enough resources in HetVNET to dedicate, network will experience congestion situations. Unfortunately, congestion can impair network performance and user satisfaction by collapsing the Quality of Services (QoS). When a vehicular user finds out that data transmission is low and has to wait for network respond, then his satisfaction typically will drop. Therefore, congestion in a vehicular network is negatively related to QoS. In a real time situation, when an accident has happened, two actions of warning other vehicles on the way and sending an alarm to immediate health help must be done within an appropriate time and with least latency. In such a scenario, if the HetVNET encounters congestion problem and no congestion mechanisms are used, the direct effect of congestion on QoS could have irreparable harms for human life, time consumption and money expenditures.

Intelligent learning methods help devices and machines to learn from existing data, and then use what they learned for new data, which the device may have never seen those data before. Machine learning algorithms are categorized into supervised and unsupervised learning. An unsupervised learning method is capable to learn and making solutions with no error evaluation. However, a supervised method is an error correction method, and learning will be matured by training [31].

Regarding to this preface, we decided to study current works related to network congestion in vehicular networks, which the authors used machine learning algorithms in them. However, the number of works in this area is limited. That being said, controlling congestion by applying an unsupervised algorithm for clustering information generated by vehicles in Vehicular Ad hoc Network (VANET) has been proposed by Taherkhani and Pierre [23]. Although their work is not about network congestion prediction, they succeeded in controlling congestion by clustering data using learning K-means algorithm. In many similar works, authors assumed congestion situation from channel busy level and then tried to propose a mechanism to control network congestion. However, for the first time, in this work, we do not use any assumption for network congestion, we propose a method to predict it, and results will show to which extend the proposed prediction method is accurate and reliable.

Moreover, the concept of fracturing control unit and data plane has been emerged by Software-Defined Networking (SDN) architecture [3,5]. In SDN, the control layer plays administrative roles in the whole network. Therefore, the control layer is able to update, configure and optimize network resources very fast and dynamically thanks to its programmability attribute [3]. SDN is adaptable, manageable, cost-effective, and ideal for dynamic environment like Het-

VNET. In this regards, a network congestion prediction model at the control layer, can help in forming and boosting an intelligent network management in SDN based architectures of HetVNET. In this paper, we propose a multiple linear network congestion prediction model for HetVNET.

This paper is organized as follows. In Section 4.2, we present a literature review, and in Section 4.3 a methodology and the proposed prediction model. Simulation scenario and numerical results are presented and discussed in Section 4.4. Concluding remarks are presented in Section 4.5.

4.2 Related Work

In the literature, several authors decided to cope with network congestion by adjusting the transmission power [21, 36, 46, 58]. For instance, Ali Shah et al. [36] defined a mechanism to reduce the transmission power in order to control traffic load of control channel in VANET. If a vehicle finds out that control channel is congested, it will inform other vehicles that they may be affected by the congestion problem. Then, vehicles are sorted based on their current transmission power. Vehicles formed several groups, and vehicles belonging to each group start to reduce transmission power fairly. In this method, the congestion is alleviated by all vehicles that are impacted by the effects of network congestion. Chakroun and Cherkaoui [21] tuned transmitting power of each vehicle based on crowding level surrounding of the vehicle. In the proposed algorithm, as far as local communication congestion does not exceed a communication congestion threshold, the transmitting power is increased, and otherwise, the transmitting power is decreased. In this paper, the value of communication congestion threshold is not defined. Rostami et al. [42] compared the performance of two approaches of reactive state based and linear adaptive approaches. In reactive state based approaches three different states are defined for channel: idle, active, and high traffic load. Active level is divided to three sub sets. Each channel occupancy level has a predefined related message transmission policy in terms of time for transmission message and message transmission rate. In linear adaptive approaches message transmission policy is defined to worthy channel utilization. Simulation results illustrate that the message throughput with the linear adaptive approach is higher than the message throughput with the stable reactive approach. Zang et al. [55], used a static and fixed threshold for channel usage. Different congestion control mechanisms are used according to the channel usage, for more than 95% and for more than 70%. However, as found by the authors, congestion may occurred with lower channel usage figures.

Taherkhani and Pierre [23] improved packet loss, average delay and probability of collision metrics by applying the K-means clustering technique (unsupervised algorithm). Its proposed strategy is divided in to three parts: congestion detection, data control and congestion control. In the congestion detection unit, it is assumed that congestion is happened whenever channel usage comes up to 70%. In the control unit, messages are collected, filtered and then clustered. In the congestion control unit, appropriate communication parameters are assigned to each cluster. Lu et al. [53] proposed a method, which reduce the bandwidth assigned to delay tolerant data and adding it to the bandwidth used by sensitive delay data. The solution approach was applied where the channel queue length had been grown more than a threshold and congestion happened. Zemouri et al. [20] proposed a model to predict density around a vehicle in the next time window by using beacons' information. Then based on density prediction, the vehicle can adjust its parameters to avoid congestion for the next time window. Hasanabadi et al. [27] proposed the Synchronized Persistent Coded Repetition (SPCR) algorithm. With SPCR, each active vehicle node broadcasts composition linear coding of messages, which are selected randomly from its queue. If the number of vehicles in a cluster is N, then the congestion control mechanism randomly selects n nodes as active (which defined as $n \leq N$) and abandons all messages from $(N-n)$ inactive nodes. Therefore, the value of n is from 0 to N. If $n = N$, it means that all N vehicles in the cluster are active and can all broadcast messages. On the other hand, if $n = 0$, it means all nodes are passive and all safety messages are dropped which is dangerous especially in critical situation like road hazards. Kolte and Madnkar [18] defined several segments and assigned each vehicle to a segment. In each segment, one node decides that which of the other nodes of the segment can use dedicated bandwidth during specific time interval. Since, segments densities are not equal, bandwidth allocation is not fair, as a node in a denser segment has to wait more to use dedicated bandwidth. Besides, time of using bandwidth for a node in crowded segment is less than a node in a non-crowded segment.

4.3 Methodology and Prediction Model

4.3.1 Designing Structure of Data set

Inspiring of current works, we consider a group of parameters, which each one has effect on creating congestion in vehicular network. Indeed, analyzing a group of different parameters, which have effect on congestion in vehicular network using learning algorithm, helps us to produce prediction model with high accuracy in congestion prediction result. Besides, assigning a weight to each parameter in prediction model can guide us to find out the importance level of parameters in terms of their effects on congestion in the HetVNET. This approach can guide us towards creating a congestion control mechanism based on most effective parameters on congestion occurrence. We consider following five vehicular network parameters in this work: Number of Vehicles (v), Data Rate (dr), DSRC Transmission Power (tp_{DSRC}), LTE Transmission Power (tp_{LTE}), and LTE Bandwidth (b).

4.3.2 Proposing the Utility Function

The network throughput is the amount of data successfully received at the destination point per unit of time. The data generation rate is the amount of data generated by the network nodes per unit of time. As a result, in part of creating data set, we propose utility function like $U(v_t)$ as a metric to detect congestion in the HetVNET. Indeed, congestion recognition is based on two factors of network throughput ((τ)) and data generation rate (α) by v vehicles. Value of $U(v_t) \in [0, 1]$ for time of t and with v vehicles. We can assume that if the value of $U(v_t)$ grows towards one, the network condition in terms of congestion improves, and if value of $U(v_t)$ collapsed towards zero then congestion is happened in the network:

$$U(v_t) = \frac{(\tau_{DSRC})(v_t) + (\tau_{LTE})(v_t)}{\alpha(v_t)}, \tag{4.1}$$

where, $\alpha(v_t)$ is the data generation rate by v vehicles for time unit t. Moreover, we consider total throughput in the heterogeneous vehicular network as sum of throughput of DSRC ($tp_{DSRC}(v_t)$) and throughput of LTE ($tp_{LTE}(v_t)$), both based on Bytes per second (Bps). This vision helps us to investigate congestion problem based on sensitivity of urban roads. For various scenarios and based on network sensibility, we define a threshold for value of utility function like T (which $T \in (0; 1)$) and based on that, we can define three network congestion states in HetVNET:

$$\text{congestion state} = \begin{cases} safe, \text{ if } (T+\gamma) < U(v_t) \leq 1 \\ warning, \text{ if } T < U(v_t) \leq (T+\gamma) \\ congestion, \text{ if } 0 \leq U(v_t) \leq T \end{cases}, \tag{4.2}$$

where $\gamma \in (0, 1)$ is used to define the warning interval and $(T+\gamma) < 1$. For instance, assume that the HetVNET is implemented in a non-safe road with high risk of car accident and the weather is rainy. In such a scenario, as the required emergency and safety services should be provided smoothly, we may set $U(v_t)$ to 0.4 or over, so T= 0.4. If we assume that $\gamma = 0.2$, then based on (5.2), we have:

$$\text{congestion state} = \begin{cases} safe, \text{ if } (0.6) < U(v_t) \leq 1 \\ warning, \text{ if } 0.4 < U(v_t) \leq (0.6) \\ congestion, \text{ if } 0 \leq U(v_t) \leq 0.4 \end{cases}$$

In this example, the congestion prediction model must make a warning when it predicts that the value of $U(v_t)$ will get less to below 0.6. Then, at this moment, the congestion control/avoidance mechanism will be executed before the value of $U(v_t)$ is collapsed to 0.4, and pushes it up to upper level like beyond 0.6. Therefore, we can be assured that in a critical network situation in terms of data traffic, our target HetVNET can provide at least an acceptable level of network services for vehicular users.

4.3.3 Multiple Linear Regression Prediction Model

In order to predict quantitative values such as for $U(v_t)$, linear regression is a popular method [95]. According to the Multiple Linear Regression (MLR) method, we use the least squares method in order to generate a best possible fitted prediction model by minimizing predicting error. It means that by using least square approach we attempted to find model coefficients (β) for our prediction model in the manner of minimizing Residual Sum of Squares (RSS). RSS is the difference between observed values of $U(v_t)$ in training data set and response values that are predicted by the prediction model [95, 99]. Based on MLR, if $\mathbf{x}= (x_0, x_1, x_2, \ldots, x_m) = (1, v, dr, tp_{DSRC}, tp_{LTE}, b)$ contains our predictor variables, and β is a set including of our model coefficients (β) which $\beta = (\beta_0,\beta_1,\beta_2, \ldots,\beta_m)$, then the quantitative value of $U(v_t)$ (which we call $\hat{y}$) can be predicted as follows:

$$\hat{y} = \beta_0 + \beta_1 v + \beta_2 dr + \beta_3 tp_{DSRC} + \beta_4 tp_{LTE} + \beta_5 b \tag{4.3}$$

If we suppose that y is the observed value of $U(v_t)$ in the data set, then $e_i = y_i - \hat{y}_i$ is residual error for i^{th} data record [95, 99]. A prediction model can be trustful, where amount of e_i is at minimum value of itself. To achieve this goal, least squares method can help us find a best fitted prediction model in linear regression problems. Therefore, according to the least squares method, we propose different values for our predictor variables and output Y. Therefore, if we consider a $(n \times 6)$ matrix of $\mathbf{X}$ and a $(n \times 1)$ matrix of Y as follows (which

n is number of observed data records in data set):

$$\mathbf{X} = \begin{bmatrix} 1 & v_1 & dr_1 & tp_{DSRC1} & tp_{LTE1} & b_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & v_n & dr_n & tp_{DSRCn} & tp_{LTEn} & b_n \end{bmatrix},$$

$$\mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} U_1(v_t) \\ U_2(v_t) \\ \vdots \\ U_n(v_t) \end{bmatrix},$$

then, we can calculate ($\mathbf{X}^T \times \mathbf{X}$) as a (6×6) matrix with subsequent members:

$$\mathbf{X}^T \times \mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} & x_{15} & x_{16} \\ x_{21} & x_{22} & x_{23} & x_{24} & x_{25} & x_{26} \\ x_{31} & x_{32} & x_{33} & x_{34} & x_{35} & x_{36} \\ x_{41} & x_{42} & x_{43} & x_{44} & x_{45} & x_{46} \\ x_{51} & x_{52} & x_{53} & x_{54} & x_{55} & x_{56} \\ x_{61} & x_{62} & x_{63} & x_{64} & x_{65} & x_{66} \end{bmatrix}, \tag{4.4}$$

where:

$x_{11} = n$	$x_{14} = \sum_{i=1}^{n}(tp_{DSRC})_i$
$x_{21} = \sum_{i=1}^{n} v_i$	$x_{24} = \sum_{i=1}^{n}(v_i)(tp_{DSRC})_i$
$x_{31} = \sum_{i=1}^{n} dr_i$	$x_{34} = \sum_{i=1}^{n}(dr_i)(tp_{DSRC})_i$
$x_{41} = \sum_{i=1}^{n}(tp_{DSRC})_i$	$x_{44} = \sum_{i=1}^{n}(tp_{DSRC})_i^2$
$x_{51} = \sum_{i=1}^{n}(tp_{LTE})_i$	$x_{54} = \sum_{i=1}^{n}(tp_{LTE})_i(tp_{DSRC})_i$
$x_{61} = \sum_{i=1}^{n} b_i$	$x_{64} = \sum_{i=1}^{n}(b_i)(tp_{DSRC})_i$
$x_{12} = \sum_{i=1}^{n} v_i$	$x_{15} = \sum_{i=1}^{n}(tp_{LTE})_i$
$x_{22} = \sum_{i=1}^{n}(v_i)^2$	$x_{25} = \sum_{i=1}^{n}(v_i)(tp_{LTE})_i$
$x_{32} = \sum_{i=1}^{n}(dr_i)(v_i)$	$x_{35} = \sum_{i=1}^{n}(dr_i)(tp_{LTE})_i$
$x_{42} = \sum_{i=1}^{n}(tp_{DSRC})_i(v_i)$	$x_{45} = \sum_{i=1}^{n}(tp_{DSRC})_i(tp_{LTE})_i$
$x_{52} = \sum_{i=1}^{n}(tp_{LTE})_i(v_i)$	$x_{55} = \sum_{i=1}^{n}(tp_{LTE})_i^2$
$x_{62} = \sum_{i=1}^{n}(b_i)(v_i)$	$x_{65} = \sum_{i=1}^{n}(b_i)(tp_{LTE})_i$
$x_{13} = \sum_{i=1}^{n}(dr_i)$	$x_{16} = \sum_{i=1}^{n}(b_i)$
$x_{23} = \sum_{i=1}^{n}(v_i)(dr_i)$	$x_{26} = \sum_{i=1}^{n}(v_i)(b_i)$
$x_{33} = \sum_{i=1}^{n}(dr_i)^2$	$x_{36} = \sum_{i=1}^{n}(dr_i)(b_i)$
$x_{43} = \sum_{i=1}^{n}(tp_{DSRC})_i(dr_i)$	$x_{46} = \sum_{i=1}^{n}(tp_{DSRC})_i(b_i)$
$x_{53} = \sum_{i=1}^{n}(tp_{LTE})_i(dr_i)$	$x_{56} = \sum_{i=1}^{n}(tp_{LTE})_i(b_i)$
$x_{63} = \sum_{i=1}^{n}(b_i)(dr_i)$	$x_{66} = \sum_{i=1}^{n}(b_i)^2$

therefore, we can calculate $\mathbf{X}^T \times \mathbf{Y}$ as follows:

$$\mathbf{X}^T \times \mathbf{Y} = \begin{bmatrix} \sum_{i=1}^{n} U_i(v_t) \\ \sum_{i=1}^{n} v_i \times U_i(v_t) \\ \sum_{i=1}^{n} dr_i \times U_i(v_t) \\ \sum_{i=1}^{n} (tp_{DSRC})_i \times U_i(v_t) \\ \sum_{i=1}^{n} (tp_{LTE})_i \times U_i(v_t) \\ \sum_{i=1}^{n} b_i \times U_i(v_t) \end{bmatrix}, \tag{4.5}$$

Finally, β contains the proposed model coefficients is computable using (4.6), and the multiple linear regression congestion prediction model can be completed:

$$\beta = (\mathbf{X}X^T \times \mathbf{X})^{-1} \times (\mathbf{X}^T \times \mathbf{Y}) = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \end{bmatrix}. \tag{4.6}$$

4.4 Simulation Scenario and Numerical Results

4.4.1 Simulation Scenario

In this paper, as part of data generation and towards generating urban simulation scenario, we use OpenStreetMap (OSM) to create map of boroughs of the city of Montreal in Canada. Then, we used the ".osm" file in Simulation of Urban Mobility (SUMO) 0.26.0 to generate road traffic. Finally, we worked with Veins LTE version 1.3 [89], which is standing on the OMNeT++ (4.6) Network simulator to simulate heterogeneous vehicular network based on IEEE 802.11p and LTE.

Table 4.1 contains attributes and parameters values, which are applied in each of the 260 running simulation scenarios. For each scenario, we set the values of v , dr, tp_{DSRC}, tp_{LTE}, and b and then calculated the value of $U(v_t)$.

After generating data extracted from executing simulation scenarios and putting it in shape of a data set, we use R programming language (using RStudio version 1.1.463) in order to create multiple linear regression congestion prediction models and statistically analyzing their performance to finally find a congestion prediction model most fitted to the observed data.

Table 4.1 Configuration Used to Generate Simulated Environment

Parameter	**Value**
Size of simulated area	1000 m $\times$1000 m
Number of lanes	4 (two in each direction)
Number of vehicles	30, 50, 100, 150, 200
Number of base station (eNB)	1
Bandwidth (IEEE802.11p)	10MHz
Bandwidth (LTE)	5 MHz, 10MHz, 20MHz
Transmission power (IEEE802.11p)	1 mW, 50 mW, 100 mW
Transmission rate (IEEE802.11p)	6-27 Mbps
Transmission power (LTE)	43 dBm, 46 dBm
Resource Blocks size	25, 50, 100
Message size	400 Bytes
Vehicles speed	0-40 km/h
Propagation model	Nakagami
Simulation time	1000 s

4.4.2 Multiple Linear Regression Analysis: Assessing Congestion Prediction Model

In the current HetVNET related works, we did not find any work that could be used as a benchmark (until today), and make comparison with our proposed method. Therefore, in order to evaluate congestion prediction model generated by multiple linear regression method, we will respond to the following questions, which are mainly considered in regression problems:

Q1) How much of variability in amount of $U(v_t)$ can be expressed by predictor variables (v , dr, tp_{DSRC}, tp_{LTE} and b) in congestion prediction model? Or using a subset of predictor variables is more effective for predicting $U(v_t)$ than having a congestion prediction model contains all five predictor variables? R square parameter can show us that how much changing in value of dependent variable like $U(v_t)$ is determined by independent variables like v , dr, tp_{DSRC}, tp_{LTE} and b in our problem [95]. Fig. 4.1 illustrates amount of R^2 for each possible 30 prediction models, which are generated using information of data set and MLR method.

Fig. 4.1 shows that in model 24, which we apply all five predictor variables to make a congestion prediction model, as we expected, it has highest coefficient of determination (R^2) among all possible 30 congestion prediction models. It confirms that, information from the variables like number of vehicles, data rate, DSRC transmission power, LTE transmission power, and LTE bandwidth are helpful to better predict the network performance in terms of problem in data transmission in HetVNET. The regression congestion prediction model 24 composed of all five considered variables, with coefficient of determination of 0.82733 (which is most close to one among other models), is most capable to express variability in amount of $U(v_t)$. Therefore, in continue we just consider information of all five predictor variables in order to generate congestion prediction models using MLR.

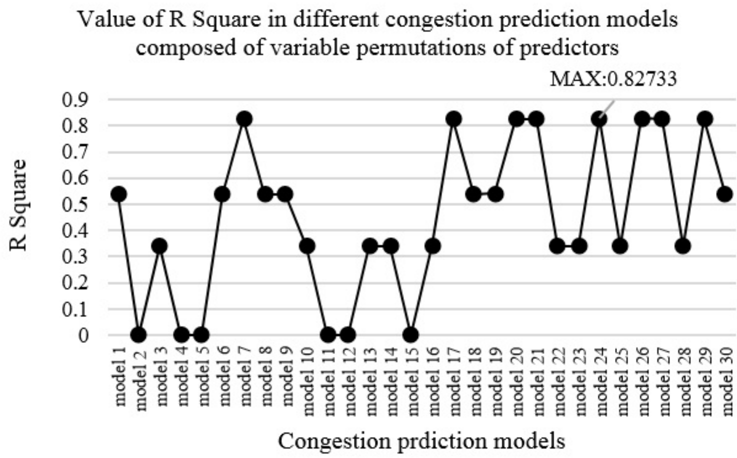

Figure 4.1 Value of R^2 for each possible 30 prediction models.

Q2) Is proposed congestion prediction model a reliable model? Based on Cross-Validation test method [95], we considered 80% of our observed data in a training data set and remaining 20% are used in a test data set. We performed this approach 20 times and for each time data belonging to training data set and test data set are selected randomly among 260 data records of our observed data. In each split of training data set, the MLR algorithm generates a model based on data of training data set.

Table 4.2 Statistical Parameters About Congestion Prediction Model 19.

	Estimate	**Std. Error**	**t value**	Pr(>|t|)
(Intercept)	−1.036e−02	1.943e−02	−0.533	0.594
v	1.555e−04	6.799e−06	22.876	>2e−16
dr	1.921e−06	5.555e−05	0.035	0.972
tp_{DSRC}	1.754e−04	1.036e−05	16.930	>2e−16
tp_{LTE}	2.371e−04	4.482e−04	0.529	0.597
b	3.281e−05	1.033e−04	0.318	0.751

We evaluate accuracy of generated prediction models using test data sets and based on Root Mean Square Error (RMSE) parameter, which is one of the best parameter to show

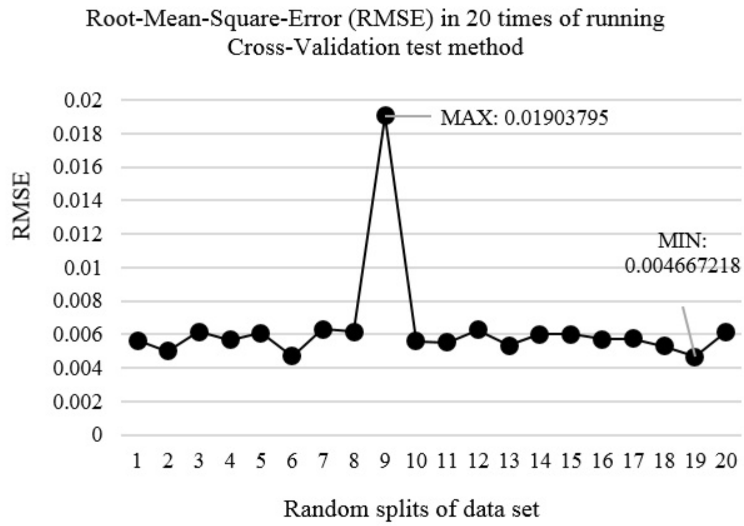

Figure 4.2 RMSE in different splits of data set using cross-validation test approach.

how much the prediction model can be trustful in terms of producing results close to actual data [100]. Fig. 4.2 shows value of RMSE for different splits of data set using cross-validation test approach. In prediction problems, a model with less RMSE is more accurate than other model with higher RMSE value. As Fig. 4.2 indicates, the prediction model 19 has minimum amount of RMSE in compare to other congestion prediction models. Table 4.2 contains coefficients of congestion prediction model 19 based on least squares in multiple linear regression method. We can see values of matrix β for model 19 (related to formula (4.6)) in estimate column of Table 4.2. Each of the five predictor variables has its own level of effectiveness on $U(v_t)$. Estimate column in Table 4.2, shows effect of each five predictor variable on smooth data transmission. Based on the estimate column, transmission power of LTE has highest effect on boosting data transmission by increasing value of respond $U(v_t)$. DSRC transmission power has been placed at second level of importance in terms of having contribute on enhancing $U(v_t)$.

Table 4.3 Statistical Metrics of Congestion Prediction Model 19.

Parameter	**Value**
Residual Standard Error	0.00607
R^2	0.801
Mean Square Error	0.000021
F-statistic	162.7

Table 4.3 provides other information about the congestion prediction model 19, as well. F-

statistic close to one indicates no relationship between $U(v_t)$ and five model's predictors. However, as Table 4.3 illustrates, F-statistic factor is far from one, which emphasizes that at least one of predictor variables of number of vehicles, data rate, DSRC transmission power, LTE transmission power, and LTE bandwidth has strong relationship with $U(v_t)$.

Even if we propose the most fitted prediction model, we could not say that our model can predict exactly observed value of $U(v_t)$ with hundred percent of accuracy in prediction results. If we apply the congestion prediction model, Residual Standard Error (RSE) helps us to estimate variance of the error (σ^2). Therefore, from Table 4.3, we can say that the proposed congestion prediction model has a variance of error of about $\sigma^2 = 0.00607$. Based on this value, we infer that the predicted value of $U(v_t))$ is as much as 0.0.00607 different from exact observed value of $U(v_t)$.

All the assessments in this work is based on prediction models that are made from analyzing and learning of data, which are generated by simulator tools. Having more and more data generated from real HetVNET could help us toward proposing congestion prediction models closer to real situations of HetVNE.

4.5 Conclusion

In the current literature related to congestion problem in vehicular networks, only a few authors applied intelligent methods using machine learning algorithms. The reason for that could be the absence of data needed for analyzing, learning and making congestion prediction models applicable to HetVNET. In this paper, we proposed a utility function to explain how the heterogeneous vehicular network can satisfy its vehicular users in terms of smooth transmitting of data. Besides, we explained about how the proposed utility function can help in having a tolerate HetVNET, which can provide required services even in critical network traffic situation. Afterwards, we move toward generating a congestion prediction model, which can predict the utility function. We generate a data set containing information records extracted from simulation scenarios of HetVNET using Veins LTE 1.3 and SUMO 0.26.0. Moreover, we propose congestion prediction model using multiple linear regression, which is a supervised machine learning method. We evaluate reliability of the proposed congestion prediction model in terms of accuracy in predicted result by using various statistical metrics such as RMSE, coefficient of determination (R^2), and Fstatistic. The approach for predicting congestion in such a proposed tolerable manner with respect to the target network's conditions and based on predicted value of defined utility function, can be applied for 5G based SDN architectures as well.

CHAPTER 5 ARTICLE 3: AN INTELLIGENT CONGESTION AVOIDANCE MECHANISM BASED ON GENERALIZED REGRESSION NEURAL NETWORK FOR HETEROGENEOUS VEHICULAR NETWORKS

Farnoush Falahatraftar, Samuel Pierre, Steven Chamberland

Mobile Computing and Network Research Laboratory (LARIM)
Department of Computer and Software Engineering, Polytechnique de Montréal
Montreal, QC H3T 1J4, Canada
Email : {farnoush.falahatraftar, samuel.pierre, steven.chamberland}@polymtl.ca.

Status: submitted in *IEEE Transactions on Intelligent Vehicles*, May 2021.

Abstract

Information generated by safety and traffic efficiency applications need strict communication requirements to smoothly exchange in Intelligent Transportation System (ITS). Data congestion in the vehicular network is a challenge that negatively affects data transmission and network performance. In this paper, we present a dynamic DSRC transmission power adaptation technique using a supervised machine learning method. This paper proposes an Intelligent Congestion Avoidance Mechanism (ICAM) based on Generalized Regression Neural Network (GRNN) to prevent congestion in the Heterogeneous Vehicular Network (HetVNET). We compare performance of the proposed GRNN congestion prediction model to other well-known methods in regression prediction problems such as Multiple Linear Regression (MLR), Support Vector Machine (SVM) for regression and Decision Tree Regression (DTR). Numerical results show that the GRNN congestion prediction model outperforms in terms of accuracy, reliability and stability. Simulation results show a substantial improvement in network performance compared to other congestion control methods in terms of packet delivery ratio, average delay and packet loss ratio.

Keywords: Heterogeneous Vehicular Networks, network congestion prediction, supervised machine learning method, Intelligent Transportation System (ITS)

5.1 Introduction

In Heterogeneous Vehicular Network (HetVNET), connected vehicle's users can be profited from various services, which are provided by Dedicated Short Range Communication (DSRC) and Long-Term Evolution (LTE) technologies [98]. In vehicular networks, data generated by vehicles can be transmitted via two types of communications: Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I). HetVNET applies DSRC and LTE in V2V and V2I communications respectively [2].

With the ever growing number of smart vehicles and significant use of diverse safety and comfort applications by users, high resource demands for data transmission could create congestion situations in HetVNET. Unfortunately, congestion typically impairs network performance and user satisfaction by collapsing Quality of Services (QoS). When a vehicular user feels that data transmission is low and has to wait for network to respond, then the user satisfaction will drop. Therefore, congestion in vehicular networks has an adverse impact on QoS. For example, when an accident has occurred, several event-driven safety messages must be sent to alert the other vehicles and the emergency services. In a such scenario, if HetVNET encounters congestion problems and a congestion control strategy is not in place to solve it, low network performance could make irreparable harm for human life.

Controlling congestion in the network is widely considered in the literature. However, in order to meet dynamic and strict communication requirements of the vehicular network such as high QoS, ultra reliability and low latency, we still need efficient centralized computational intelligence methods to controlling congestion. Considering European Telecommunications Standards Institute (ETSI), Decentralized Congestion Control (DCC) based methods suffer from instability due to oscillating between states, and unfairness [86, 101]. Moreover, defining an optimal threshold for the value of Channel Busy Ratio (CBR) is a challenge and consequently, imbalance channel utilization is a problem in ETSI based methods [32]. These challenges make DCC unreliable, especially in dense vehicular networks.

Prosperity of Artificial Intelligence (AI) based methods to solve network related problems such as misbehaviour detection, driver assistance and traffic management shows that these methods become promising approaches to provide efficient solutions for network concerns such as data congestion. Indeed, we can use machine learning methods to create reliable data traffic prediction models. Using the accurate data traffic prediction result can effectively increase performance of the network by applying the congestion controlling policies in advance.

In the literature, network congestion related papers have notably two steps: finding congestion and then controlling it. Number of research works that applied AI methods in these two steps are limited. Controlling congestion by applying an unsupervised machine learning algorithm to cluster the information generated by vehicles in Vehicular Ad-hoc network (VANET) has been presented in [23] by Taherkhani and Pierre. Although their work is not about network congestion prediction in HetVNET, they succeeded in controlling congestion by clustering data using learning K-means algorithm.

Moreover, distributed congestion controlling techniques are widely considered in research works of VANET [36, 38, 102]. In these approaches, vehicles make decisions and execute controlling mechanisms independently and relying on the information received from other vehicles. Frequently processing the huge information and making appropriate congestion controlling decisions within a short time, make overloads for vehicles. Therefore, they need powerful computation and storage resources. Considering these challenges, applying centralized methods could improve quality of congestion controlling decisions using powerful resource. The focused congestion management vision can help quick implementation of congestion controlling decisions. Regarding implementing centralized methods, Software-Defined Networking (SDN) architecture is a helpful technology. In SDN, the control layer plays administration role in whole of the network. Therefore, the control layer as a manager has wide information of network [3], [5]. It is able to update, configure and optimize network resources very fast and dynamically thanks to its programmability attribute [3]. Using global view of SDN to create centralized congestion management system is a privilege. In this regards, generating and then applying network congestion prediction and avoidance mechanism at the control layer, can help in forming and boosting an intelligent network management in SDN based architectures of HetVNET.

In this paper, we propose an intelligent network congestion avoidance mechanism based on an Artificial Neural Network (ANN) prediction method for HetVNET. This congestion avoidance technique is applicable in the control layer of SDN in HetVNET.

Our contributions are listed below:

1. We propose a Generalized Regression Neural Network (GRNN) congestion prediction model;
2. We devise an intelligent congestion avoidance mechanism;
3. We evaluate performance of: the GRNN congestion prediction method, and the proposed Intelligent Congestion Avoidance Mechanism (ICAM), using numerical results.

We will show that the proposed intelligent congestion avoidance mechanism is a reliable method, which can successfully keep the performance of HetVNET at a fine level for users, even in a potentially congestion state of the network. A combination of defined utility function and GRNN prediction method, assures stability and accuracy of the proposed intelligent congestion avoidance mechanism.

This paper is organized as follows. In section 5.2, we provide a review of the existing congestion controlling methods. Section 5.3 presents methodology and the proposed ICAM. Simulation scenario and numerical results are presented and discussed in section 5.4, and Section 5.5 concludes this paper.

5.2 Related Work

Regarding importance of the network congestion problem, it has been widely considered in scientific works and several congestion control protocols are provided by researchers [11, 13, 14, 103].

Many of the congestion control algorithms, such as the numerous variants of the Transmission Control Protocol (TCP) and Blue [11, 13] are based on data loss in the network. The authors considered that packets have been lost due to limited buffer capacity before they are transmitted by routers. Indeed, these algorithms use packet loss as a signal of occurring congestion in the network and as soon as a packet is lost they switch to congestion repair/avoidance phase. However, packet loss may happen because of a momentary traffic bursts or it may come after applying a security policy at intermediate nodes between the source node and the destination node. In Bottleneck Bandwidth and Round-trip propagation time (BBR) that has been proposed by Google, sender node performs estimations based on network throughput and making decisions are according to the evaluation of network performance [103]. Throughput is widely used to show performance of the congestion controlling mechanisms in the network. Therefore, we considered throughput in this paper as a main component of the utility function.

We narrow down our study and consider existing research works specifically related to vehicular networks.

In the literature, several adapting transmission power and data rate approaches have been proposed to control congestion in the vehicular network [36, 46, 58]. Aygun et al. [46] proposed an algorithm named "Environment and Context-aware Combined Power and Rate Distributed Congestion Control for Vehicular Communication (ECPR)". This algorithm made a trade off between two issues. First, enhancing awareness of vehicular users (by adjusting the transmission power) and providing maximum message transmission rate. Second, controlling channel load in VANET with respect to efficient utilizing of channel. Ali Shah et al. [36] worked on controlling congestion in control channel in VANET. They defined a mechanism to reduce transmission power, in order to control traffic load in control channel. If a vehicle finds out that control channel is congested, it will inform other vehicles that may be affected by the congestion problem. In this method, congestion in control channel is alleviated by all the vehicles that are suffered from congestion in control channel. After that congestion is reduced, all vehicles have the right to rise transmission power step by step to reach a standard level.

In [104], the authors proposed a method to adjust beaconing transmission power in which a predefined value for Maximum Beaconing Load (MBL) should not be exceeded. According to the Distributed Fair Power Adjustments for Vehicular environments (D-FPAV) mechanism, each vehicle like j which applies transmission power of Pj receives the value of transmission power Pi used by vehicle i in the range. If the value of Pi is lower than the value of Pj, the vehicle j must select Pi as its new transmission power. Regarding fast changing in the topology of vehicular networks, the new value of transmission power might not be usable, since the vehicles have changed their locations and they are not in the same range. This could reduce the possibility of successfully receiving beacons in the network.

Egea-Lopez and Pavon-Mariño [58] considered two steps: 1) avoiding congestion in channel by keeping beacon rate for each transmission power less than a threshold like C, and 2) enhancing the number of delivered beacons messages by assigning minimum beacon rate (r_{min}), simultaneously. The authors proposed "fair adaptive beacon rate with multiple power levels for inter-vehicular communications (FABRIC-P)". In FABRIC-P each vehicle must calculate best rate for a beacon message use transmission power p, where $r_{min} \leq$best beacon rate $\leq R_{max}$ (the topmost beacon rate for a vehicle is R_{max}).

In [105], the authors considered DCC gatekeeper in ETSI TS 102687 [106] that has three main components: prioritization, queuing, flow control and rate adaptation. The prioritization mechanism are based on three categories: high priority Decentralized Environmental Notification Messages (DENM), regular DENM, and Cooperative Awareness Messages (CAM). The high priority DENM must be transmitted immediately and without waiting in the queue. In the flow control scheme, to prevent from saturating the channel, the messages that have been waited for a time more than the maximum queuing time must be discarded. In the rate adaptation, regarding the priority of the messages, the packet rate adaptation for each node has been considered. The results showed that the DCC algorithms almost withheld CAMs transmission. Therefore, the DCC gatekeeper queue required efficient strategies to increase transmitting of CAMs in the medium to dense vehicular environments.

In [102], vehicles should adjust the beaconing transmission power using proposed non-cooperative power control game theory. In this theory, a pay-off function is introduced that composed of two parts: utility function and price function. The utility function encourages vehicles with lower transmission power to enhance the beaconing transmission power. On the other hand, in dense vehicular environment, the value of price function is high for vehicles which use high transmission power. Therefore, the utility function and the price function are the players of the game. Although the proposed approach is a fairness method, in a very dense vehicular environment a joint beaconing transmission power and transmission rate controlling method is required to solve unfairness of ETSI DCC.

Moreover, estimating the vehicles density is widely used for generating network congestion controlling mechanisms. In the proposed mechanisms mostly vehicles independently calculate the vehicle density around. Zemouri et al. [20] proposed, a model to predict density around a vehicle in the next time window by using beacon's information. They assumed a beacon message contains: ID, current position, speed, destination, number of vehicles ahead, and number of vehicles behind. By using information of received beacon messages, a vehicle can find out: furthest back, furthest forth, closest back and closest forth vehicles. Then based on density estimation, the vehicle can adjust the network parameters in order to avoid congestion in the next time window.

Hasanabadi and Valaee [27] proposed the Synchronized Persistent Coded Repetition (SPCR) algorithm. With SPCR, each active vehicle node broadcasts composition linear coding of messages selected randomly from its queue. If the number of vehicles in a cluster is N, then congestion control mechanism randomly selects n node as active node (which defined as $n \leq N$) and abandons all messages from $(N - n)$ inactive nodes. However, if $n = N$, it means that all N vehicles in cluster are active and all can broadcast messages. It is like no congestion control mechanism is applied, while, as mentioned in [27], the objective of proposing the congestion control mechanism is to control amount of messages by omitting a part of them. On the other side, if $n = 0$, it means all nodes are passive and all safety messages will be dropped and it is dangerous especially in critical situation like facing road hazards.

Kolte et al. [18] defined several segments and assigned each vehicle to a segment. A node in each segment determines which node of the segment can use dedicated bandwidth during specific time interval. Since, segments densities are not equal, bandwidth allocation is not fair. As a node in a denser segment has to wait more to use dedicated bandwidth. Moreover, time of using bandwidth for a node in crowded segment is less than a node in a non-crowded segment.

In [107], a Density Histogram (DH) is created to estimate the vehicle density in a predefined road segment. The authors believe that using information of vehicle density can improve performance of Linear Message Rate Integrated Control (LIMERIC) algorithm [44]. The results show that using estimated density of vehicles in the LIMERIC algorithm could improve awareness probability, especially in the scenario of using three packets in time window of one second and in a dense vehicular network. However, awareness probability using five packets in time window of one second still needs efficient methods.

As explained, different parameters have been considered in existing research works as effective metrics on congestion in network, such as transmission power, data transmission rate and vehicles density. Inspired by related works, we consider a group of parameters, which each one has effect on creating and reducing congestion in vehicular network. Using learning algorithm and analyzing different parameters that affect on congestion in a vehicular network, help us produce a reliable prediction model with high accuracy and more stability in the congestion prediction result. In this paper, a transmission power adaptation technique based on a supervised machine learning method is proposed to avoid congestion in HetVNET. Moreover, Wireless Access in Vehicular Environments (WAVE) standardization is considered in this paper.

5.3 Methodology and Prediction Model

5.3.1 Proposing Utility Function

Network throughput is the amount of data successfully received at destination per unit of time (e.g., s). Data generation rate is the amount of data generated by network nodes per unit of time (e.g., s). Based on the network throughput and the data generation rate definition, if a node generates data with rate of k bytes per unit of time and q bytes are successfully received at destination per unit time ($k \neq 0$ and $q \leq k$), therefore we can calculate network performance in terms of smooth data transmission by dividing q over k. Regarding this explanation, in part of creating dataset, we propose utility function like $U(v_t)$ (which $U(v_t) \in [0, 1]$) as a metric to detect congestion in the heterogeneous vehicular network. Indeed, congestion recognition is based on two factors of network throughput (τ) (which in our example is equal to q) and data generation rate (g) (which in our example is equal to k) by v number of vehicles for time of t. We can suppose that, if the value of $U(v_t)$ increases towards one, the network condition in terms of congestion goes to desired situation, and if the value of $U(v_t)$ decreases towards zero then congestion is occurred in the network:

$$U(v_t) = \frac{(\tau_{DSRC})(v_t) + (\tau_{LTE})(v_t)}{g(v_t)}. \tag{5.1}$$

In (5.1), $g(v_t)$ is the data generation rate which is equal to number of generated data, by v vehicles over time t (based on unit of time). Besides, we consider total throughput in the heterogeneous vehicular network as sum of throughput of DSRC ($\tau_{DSRC}(v_t)$) and throughput of LTE ($\tau_{LTE}(v_t)$) both based on Bytes per second (Bps).

In the roads and the crosstown expressways, which vehicular users traveling cross-country, the emergency information about road hazards or accidents is valuable and must be transmitted within an appropriate time. Therefore, HetVNET performance should be as acceptable as could save human's life and their money expenses. For various scenarios and based on network sensibility, we can define a threshold for value of utility function like T (which $T \in (0,1)$) and based on that we can specify three states for network congestion situation in HetVNET as follows:

$$\text{congestion state} = \begin{cases} safe, \text{ if } (T+\gamma) < U(v_t) \leq 1 \\ warning, \text{ if } T < U(v_t) \leq (T+\gamma) \\ congestion, \text{ if } 0 \leq U(v_t) \leq T \end{cases}, \tag{5.2}$$

where $\gamma \in (0,1)$ is used to define the warning interval, and $(T+\gamma) < 1$. This will help us to investigate congestion problem based on sensitivity of our urban roads. For clarifying more about (5.2), assume that we are talking about a HetVNET implemented in one of non-safe roads with high risk of car accident and the weather is rainy. In this kind of scenario, required emergency and safety services should be provided smoothly by target HetVNET. Therefore, the network manager may define at least value of 0.4 for $U(v_t)$ in target HetVNET, then T= 0.4. If we assume that γ= 0.2, then based on (2), we have:

$$\text{congestion state} = \begin{cases} safe, \text{ if } (0.6) < U(v_t) \leq 1 \\ warning, \text{ if } 0.4 < U(v_t) \leq (0.6) \\ congestion, \text{ if } 0 \leq U(v_t) \leq 0.4 \end{cases}$$

In this example (all these values are used as examples, just to clear to understand the application of the utility function), congestion prediction model must make a warning when it predicts that the value of utility function is less than 0.6. Therefore, at this moment the congestion control/avoidance mechanism should be executed before the value of $U(v_t)$ decreases to 0.4. Then, performing congestion avoidance mechanism pushes the value of $U(v_t)$ up to an upper level like beyond 0.6. Therefore, we can be assured that in critical network situation in terms of data traffic, our target HetVNET can provide at least an acceptable level of network services for vehicular users. Thus, the vehicular users will not experience disrupting in the required services, consequently the user satisfaction will stay at a fine level. Furthermore, urban roads can be divide into several areas. A segment of road may have a transient condition for example for a bad weather condition or an especial event that are not stable situations. Therefore, each part can have a different value of T based on sensitivity of the road. Besides, In such cases, the network manager can consider a temporary value of T, and after passing the particular conditions, the value of T may decrease. Therefore, the network congestion management of HetVNET will be adaptable to the road conditions and the congestion state defined in (5.2) has flexible intervals for each state, based on the road conditions and network management policies.

5.3.2 Designing Structure of Dataset

Basically prediction is about forecasting an event or situation, which we are not sure to happen. Information acquired form previous experiences and knowledge about current network situation are required to generate network congestion prediction model in the HetVNET. The step of designing a dataset with features related to the problem is important in any prediction problem, since erroneous prediction can be an effect of selecting unrelated and useless parameters in training dataset.

Since there is not an accessible dataset contains information of these parameters extracted from a real HetVNET, we generate simulated data. We consider the following five vehicular network parameters in this work: number of vehicles (v), data rate (dr), DSRC transmission power (tp_{DSRC}), LTE transmission power (tp_{LTE}), and LTE bandwidth (b). These are the features of the dataset or predictor variables in the congestion prediction model. In this paper, the considered congestion prediction problem is a regression prediction problem. Therefore, as Fig. 5.1 shows, we do not have any label or class in the dataset, instead the value of utility function is calculated using (5.1). Indeed, we apply value of five predictor variables of a row of dataset shown in Fig. 5.1, to generate a simulation scenario and with this configuration the simulation scenario runs for a specific time t. After terminating the simulation time t, the value of utility function is calculated using (5.1) and the value is inserted in the corresponding row. Since dataset is filled by data extracted from running simulation scenarios in time t, any prediction model trained using the dataset can predict the value of utility function in next t time. In subsection 5.4.1 Simulation Scenario, we will provide more details about generating simulation scenarios, parameters and the values.

5.3.3 Generalized Regression Neural Network (GRNN) Prediction Model

Based on (5.1), the utility function makes an estimate of the network performance using throughput and packet generation rate. The value of utility function is calculated using the recent value of throughput and packet generation rate. We put this value (the value of

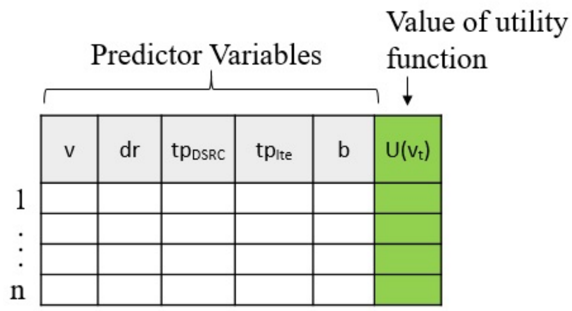

Figure 5.1 Structure of dataset.

$U(v_t)$) in the dataset. GRNN uses the dataset to train and after completing the training step, the prediction model can predict the value of utility function using five predictors variable. In other words, formula (5.1) calculates recent amount of utility function, and the GRNN method predicts the value of utility function in the future and based on the five predictor parameters. Indeed, amount of $\hat{U}(v_t)$ is a predicted future value of $U(v_t)$ without using the utility function and based on the five predictors (here, the value of t in $U(v_t)$ defines that how much does the future far from now).

By analyzing dataset information, our aim in this section of the paper is to define a prediction model to predict the value of $U(v_t)$. In our previous example (in subsection 5.3.1 Proposing Utility Function), if the prediction model predicts that $\hat{U}(v_t)$ is equal to 0.4 (this number is used as an example), network congestion management can infer if it is a warning situation in terms of congestion problem in network or not, using (5.2). In other words, in a part of city, which is not crowded or there is not a serious risk of roads hazard, a low amount for T might be defined by the network manager. Therefore, a value like 0.4 for the $\hat{U}(v_t)$ could be an acceptable value with no worry and put the network state in the safe category. However, in a dense vehicular scenarios that there are road hazards, this amount for $U(v_t)$ can provide troubles for humans life.

To predict quantitative value, like the value of $U(v_t)$, regression methods are widely used [95]. In this paper, $\mathbf{x}= [x_1, x_2, \ldots, x_m]$ is a set containing m number of features, which corresponds to $\mathbf{x}= [x_1, x_2, x_3, x_4, x_5] = [v, dr, tp_{DSRC}, tp_{LTE}, b]$ $(m = 5)$.

GRNN is a type of feed-forwarding neural network with associative memory which contains four layers in its architecture. These layers are input layer, pattern layer, summation layer and output layer [108]. We need a functional form to implement system identification; GRNN applies joint probability density function (pdf) to generate the functional form like $f(x, y)$. Then, it uses a Parzen window [109] to estimate $\hat{f}(x, y)$. Following formula calculates ex-

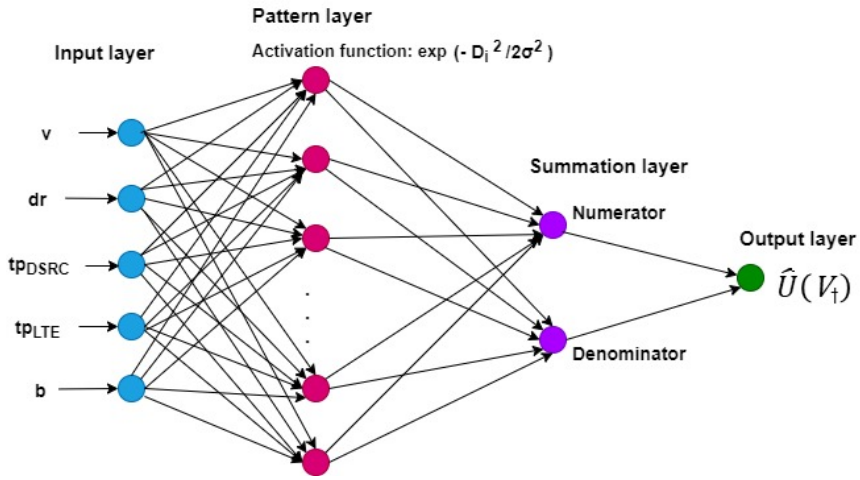

Figure 5.2 GRNN structure for congestion prediction model in HetVNET.

pected value of y using input value such as x:

$$E[y|x] = \frac{\int_{-\infty}^{\infty} yf(x,y)dy}{\int_{-\infty}^{\infty} f(x,y)dy}, \tag{5.3}$$

which:

$$\hat{f}(x,y) = \frac{1}{(2\pi)^{\frac{m+1}{2}}\sigma^{m+1}}.\frac{1}{n}\sum_{i=1}^{n} exp(-\frac{D_i^2}{2\sigma^2}).exp[-\frac{(y-y_i)^2}{2\sigma^2}], \tag{5.4}$$

where, n is the number of observed data records in the dataset and m, the number of predictor variables. GRNN does not use learning rate as many of other neural networks, but it uses smoothing factor (σ) (which $\sigma \in (0.1, 1)$). In (5.4), x and y are random variables. Indeed, we need to estimate value of y using input vector like $\mathbf{x}$, which $\mathbf{x} = [v, dr, tp_{DSRC}, tp_{LTE}, b]$. In addition, x_i and y_i represent the i^{th} sample that we have in dataset.

$$\mathbf{X} = \begin{bmatrix} v_1 & dr_1 & tp_{DSRC1} & tp_{LTE1} & b_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ v_n & dr_n & tp_{DSRCn} & tp_{LTEn} & b_n \end{bmatrix}$$

$$\mathbf{Y} = \begin{bmatrix} U_1(v_t) \\ U_2(v_t) \\ \vdots \\ U_n(v_t) \end{bmatrix}$$

As Fig. 5.2 shows, at the input layer, we need as many neurons as the number of predictor parameters, which we have in the dataset. Therefore, in this paper, we consider five neurons in the first layer, which are $v, dr, tp_{DSRC}, tp_{LTE}$ and b. Besides, the number of neurons in the second layer is same as the number of samples in the training dataset. GRNN uses the Gaussian kernel as the activation function in the pattern layer. In GRNN, data patterns are recognized during the training phase. In the training step, Euclidean distance is used to calculate the difference between input sample vector x and i^{th} training sample x_i using the next formula:

$$D_i^2 = (x - x_i)^T (x - x_i). \tag{5.5}$$

The third layer has two neurons, which are denominator and numerator. For n data records, former totals up the weights and the latter adds up multiplications of weights and the observed values of y. Output value of the numerator is divided by the value of denominator. Finally, GRNN estimates the value of utility function which is $\hat{U}(v_t)$ or $\hat{y}$ using (5.6). Following formula is generated by replacing (5.4) in (5.3) and using (5.5):

$$\hat{y}(X) = \frac{\sum_{i=1}^{n} y_i exp(-\frac{D_i^2}{2\sigma^2})}{\sum_{i=1}^{n} exp(-\frac{D_i^2}{2\sigma^2})}. \tag{5.6}$$

5.3.4 Proposing ICAM

ICAM is a centralized mechanism and can be employed in the controller of SDN in the HetVNET. Congestion prediction model predicts that applying the values of predictors for t unit of time in the HetVNET concludes with a network congestion situation or a warning case or a safe state. The predicted value of the utility function which is $\hat{U}(v_t)$ is the output of GRNN prediction model. Therefore, the value of $\hat{U}(v_t)$ must be applied in (5.2) and in place of $U(v_t)$. Then, the value of tp_{DSRC} is adapted based on the prediction result. In this section, we explain a dynamic DSRC transmission power adaptation approach. The value of tp_{DSRC} must be adjusted based on the predicted network congestion status. It is worth mentioning that the value of DSRC transmission power can not be less than a minimum allowed value ($tp_{DSRCmin}$) and more than a maximum allowed value ($tp_{DSRCmax}$). The network manager is in charge of defining and modifying the value of $tp_{DSRCmin}$, $tp_{DSRCmax}$, T and γ. The pseudo code of ICAN algorithm is presented in Algorithm 2. As shown in this algorithm, in the third line, based on the prediction result generated by proposed GRNN congestion prediction model and using (5.2), one of the following state is true:

- *It is a "safe" state:* the value of DSRC transmission power is doubled. If the new value is more than the predefined value of $tp_{DSRCmax}$, the maximum allowed value for

DSRC transmission power is considered. The GRNN prediction model again is used to confirm that applying the new value of tp_{DSRC} by vehicles in the network would not be a risk of congestion for HetVNET (line 18). Therefore, based on the prediction result and using (5.2), if it is a "safe" state then the new value of tp_{DSRC} is applied in the HetVNET. However, if the prediction result is a "warning" state (line 20), the new value of tp_{DSRC} can not be used and the previous value of tp_{DSRC} (the value before it is doubled) should applied in the network (line 9). Finally, if it is a "congestion" state, the new doubled value of tp_{DSRC} must be ignored and the value of $tp_{DSRCmin}$ must be considered in HetVNET.

- *It is a "warning" state:* the half of the value of DSRC transmission power is considered. If the new value is less than the predefined value of $tp_{DSRCmin}$, the minimum allowed value for DSRC transmission power is considered. The GRNN prediction model again is employed to assure us that applying the new value of tp_{DSRC} in HetVNET would not be a risk of congestion for network (line 17). Therefore, based on the prediction result and using (5.2), if it is a "safe" state then the new value of tp_{DSRC} is applied in the HetVNET. However, if the prediction result is a "warning" state (line 20), the new value of tp_{DSRC} can not be used and this value should be reduced to half of it (line 9). Again, the prediction model must predict until a safe result is obtained. Finally, if it is a "congestion" state, the new value of tp_{DSRC} must be ignored and the value of $tp_{DSRCmin}$ must be considered in HetVNET.

- *It is a "congestion" state:* the value of $tp_{DSRCmin}$ is applied to HetVNET.

For any change in the value of v or dr or tp_{LTE} or b or γ or T, the GRNN prediction model must be used to adjust the value of tp_{DSRC} accordingly (based on the line 25). Changing the value of dr or tp_{LTE} or b or γ or T by the network manager, can be done based on the network management decisions upon any reason that might not necessarily be related to congestion controlling policies.

In the first prediction (line 3), when the result is safe, it means that applying the input value of tp_{DSRC} by vehicles would not make difficulties for data transmission. Using a high value of transmission power, while it does not make a congested network, can help to broadcast messages to a larger area and more number of vehicles can receive the messages. Regarding this issue, when the result of prediction is a safe state (in the line 4), we try to increase the value of tp_{DSRC} (in the line 5 to 7) and see the prediction result using the new value (in the line 17). If intensifying the DSRC transmission power does not change the prediction result of safe to warning or to congestion, the new value of tp_{DSRC} should be applied by vehicles in

HetVNET.

As Algorithm 2 shows, ICAM is a recursive and dynamic algorithm that makes a flexible and elastic network according to data traffic situation in the network and road conditions (as explained in subsection 5.3.1 Proposing Utility Function). Moreover, ICAM is proposed based on a centralized vision, and it is supposed to be employed at the control layer of SDN. We applied the parameters that are compatible with the centralized strategy characteristics. For example, we did not consider channel busy level in V2V communications, as a predictor parameter. Since this parameter can be measured by vehicles, and consequently it is a main parameter in distributed methods such as ETSI DCC based strategies.

Algorithm 2 Pseudo code of the ICAM algorithm.

1: **Input 1:** the value of $v, dr, tp_{DSRC}, tp_{LTE}, b$, γ and T
2: **Input 2:** $tp_{DSRCmin}$ and $tp_{DSRCmax}$
3: **Predict** congestion state by GRNN prediction model and using (5.2)
4: **if** congestion state = "safe" **then**
5:     $tp_{DSRC} = tp_{DSRC} \times 2$
6:     **if** $tp_{DSRC} > tp_{DSRCmax}$ **then**
7:         $tp_{DSRC} = tp_{DSRCmax}$
8:     **end if**
9: **else if** congestion state = "warning" **then**
10:     $tp_{DSRC} = \frac{tp_{DSRC}}{2}$
11:     **if** $tp_{DSRC} < tp_{DSRCmin}$ **then**
12:         $tp_{DSRC} = tp_{DSRCmin}$
13:     **end if**
14: **else if** congestion state = "congestion" **then**
15:     apply $tp_{DSRC} = tp_{DSRCmin}$ and go to the line 25
16: **end if**
17: **Predict** congestion state by GRNN prediction model and using (5.2)
18: **if** congestion state = "safe" **then**
19:     apply tp_{DSRC}
20: **else if** congestion state = "warning" **then**
21:     go to the line 9
22: **else if** congestion state = "congestion" **then**
23:     apply $tp_{DSRCmin}$
24: **end if**
25: **if** the value of v or dr or tp_{LTE} or b or γ or T changed **then**
26:     go to the line 3
27: **end if**

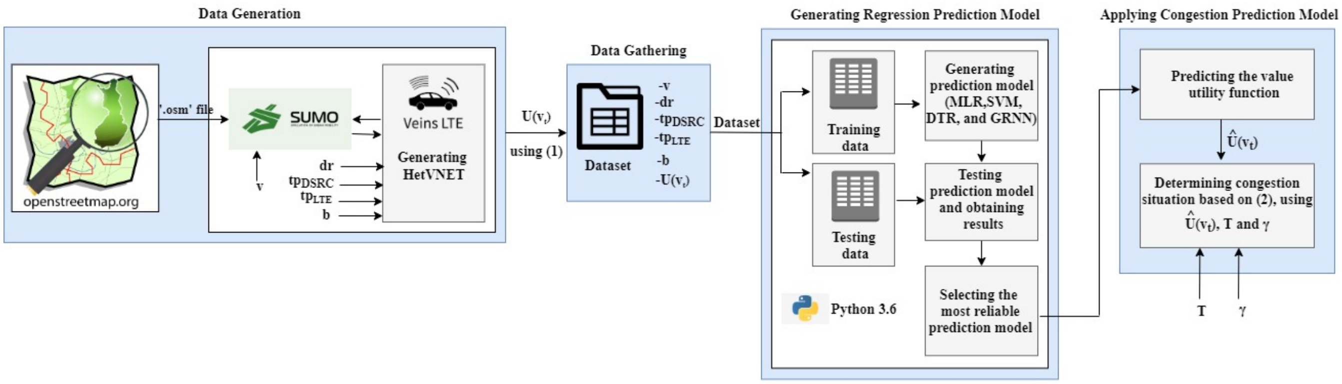

Figure 5.3 Work flow of generating and using congestion prediction for HetVNET.

5.4 Simulation Scenario and Numerical Results

5.4.1 Simulation Scenario

As part of data generation, as shown in Fig. 5.3, towards generating urban simulation scenario, we use OpenStreetMap (OSM) [90] to create map of boroughs of Montreal. Then, we use the ".osm" file as an input file for Simulation of Urban Mobility (SUMO) 0.26.0 [88] to generate the road traffic. SUMO needs ".osm" file as an input to generate environment with all roads, buildings, intersections, and others like it is in reality. Besides, we worked with Veins LTE version 1.3 [89], which is standing on the OMNeT++ (4.6) Network simulator to simulate heterogeneous vehicular network based on IEEE 802.11p and LTE. During the simulation time, vehicles join to /departure from the simulation scenario in random time and speed. SUMO and Veins LTE were running simultaneously to generate simulation scenario in Linux (Ubuntu 16.04), where SUMO is used to generate vehicles traffic and movement and Veins LTE as a network simulator. Vehicles are equipped with both LTE and IEEE 802.11p interfaces. Moreover, we defined a vehicle accident in a specific time (t= 70 s) of running simulation scenario to generate more load of data.

Table 5.1 contains attributes and parameters values, which are applied to generate simulated environment of HetVNET. Simulation time in each run is 1000 s. In each run, we has changed the values of $v, dr, tp_{DSRC}, tp_{LTE}$ and b and we have calculated the value of $U(v_t)$. Besides, the tp_{DSRC} can have a value between -20 dBm and 32 dBm [52].

After generating data extracted from executing simulation scenario and putting it in a shape of a dataset (data gathering part of Fig. 5.3), we use Python version 3.6, in order to create regression congestion prediction models using Linear Regression (MLR), Support Vector

Machine (SVM) for regression, Decision Tree Regression (DTR) and GRNN. Moreover, Classification and Regression Tree (CART) algorithm is applied by Scikit-Learn library of Python to train regression decision trees [94]. The dataset used in this work is not a high dimensional and huge dataset, hence we do not consider deep learning methods. Since, the data related to each of five parameters are different in range and unit, we normalize all of the data records. Then, we statistically analyze the performance of the prediction models to finally finding a congestion prediction model most fitted to the problem (part of generating regression prediction model in Fig. 5.3). Please note that, we used the simulators and the configurations of Table 5.1 two times in this work. First, to generate dataset and second, to evaluate performance of the ICAM.

Table 5.1 Configuration Used to Generate Simulated Environment

Parameter	**Value**
Size of simulated area	1000 m $\times$1000 m
Number of lanes	4 (two in each direction)
Number of vehicles	50, 100, 150, 200
Number of base station (eNB)	1
Bandwidth (LTE)	5 MHz, 10MHz, 20MHz
Transmission power (LTE)	43 dBm, 46 dBm
Bandwidth (IEEE802.11p)	10MHz
Minimum transmission power (IEEE802.11p)	-20 dBm
Maximum transmission power (IEEE802.11p)	32 dBm
Transmission rate (IEEE802.11p)	6-27 Mbps
Maximum transmission range (IEEE802.11p)	1000 m
Message size	400 Bytes
Vehicles speed	0-40 km/h
Propagation model	Nakagami
Simulation time	1000 s

5.4.2 Assessing Congestion Prediction Models

We apply K-fold cross validation technique [95] (which K=10) to evaluate congestion prediction models. The K-fold cross validation is used to prevent data over-fitting and under-fitting trap [94]. Performance of the prediction models generated by MLR, SVM, DTR and GRNN is evaluated using well-known regression metrics, such as Root Mean Square Error (RMSE) and R square (R^2) [94]. In each fold of dataset, the MLR, SVM, DTR, and GRNN algorithms generate a model based on (same) data of a fold, simultaneously. Then, we evaluate performance of the generated congestion prediction models using data of test dataset, which models have not seen before.

In this section accuracy, reliability and stability of the GRNN prediction model are evaluated and compered to MLR, SVM and DTR.

Evaluating the accuracy

RMSE is used to assess the accuracy of the prediction models. RMSE values of congestion prediction models generated by MLR, SVM in regression, DTR and GRNN are shown in Fig. 5.4. Models with higher value of RMSE are those with higher error in prediction results. Thus, a congestion prediction model with a lower value of RMSE is the most trustful model among 40 prediction models. Based on the results presented in Fig. 5.4, GRNN shows better performance than the other three methods. The lowest value of RMSE is 0.07 and belongs to the model number six generated by GRNN. This value shows that the mentioned model could estimate the amount of $\hat{U}(v_t)$ with a value of 0.07 vary from the amount of $U(v_t)$, while SVM and DTR did it with 0.14 approximately differ from the value of $U(v_t)$ using the same data.

Testing the reliability

Fig. 5.5 shows coefficient of determinations (R^2) for each of MLR, SVM, DTR, and GRNN models. R square parameter can show that how much changing in the value of dependent variable like $\hat{U}(v_t)$ is determined by independent variables, such as $v, dr, tp_{DSRC}, tp_{LTE}$ and b [95]. All the ten GRNN congestion prediction models have higher value of R square than the other thirty models generated by the other three methods and using the same training dataset. Again, model number six, which has a minimum RMSE, shows the highest value of the coefficient of determinations as 0.86, which is most close to one among other models. Therefore, model number six is the most capable to express variability in amount of $\hat{U}(v_t)$ among other models. Weak performance of MLR in both of Fig. 5.4 and Fig. 5.5 indicates that the output ($\hat{U}(v_t)$) and the five predictors tend to show a non-linearity relationship. Note that, we applied polynomial kernel of SVM in regression.

Measuring the stability

For each of the four types of methods, we calculate variance of Mean Square Error (MSE) of ten folds. A low value of variance indicates that the value of MSE of ten folds are not spread out far from each other. In other words, a low value in variance of MSE shows stability of the prediction method. As shown in Table 5.2, GRNN flaunts with the lowest amount of variance of MSE, which also confirms that GRNN is more converged than other three types of congestion prediction models.

Moreover, the GRNN congestion prediction model is compared to MLR, SVM and DTR in terms of CPU time in μs. As listed in Table 5.2, although the amounts of CPU time for MLR, SVM and GRNN are very close to each other, GRNN required less execution time among the four methods.

The results shown in Fig. 5.4 and Fig. 5.5 and Table 5.2, indicate that GRNN congestion prediction model is more reliable, more stable and better converged which also could predict more accurate than prediction models of MLR, SVM and DTR.

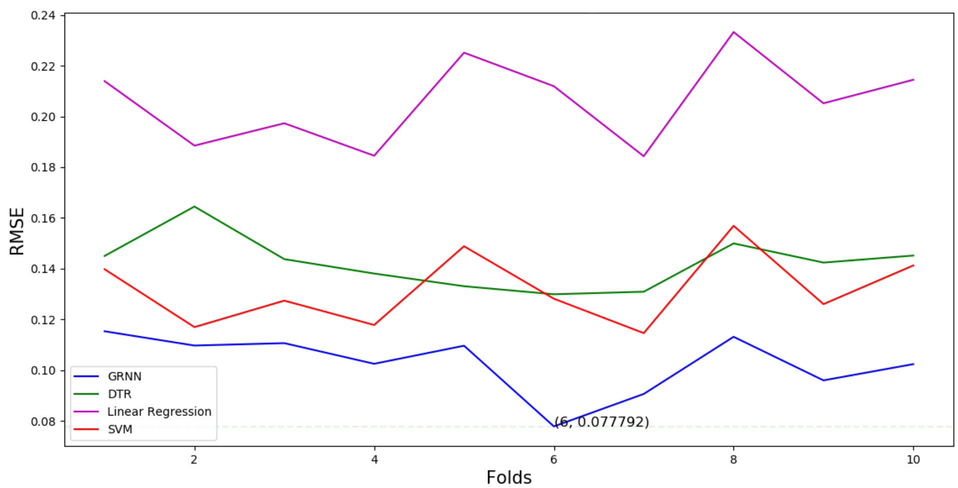

Figure 5.4 RMSE of GRNN, DTR, Linear Regression, and SVM congestion prediction models in HetVNET.

Table 5.2 Variance of MSE and CPU Time of Congestion Prediction Models

Prediction Model	Variance of MSE	CPU Time (μs)
MLR	4.431770e-05	587.5
SVM	1.346291e-05	594
DTR	2.171744e-05	643.4
GRNN	**4.711306e-06**	**582.6**

5.4.3 Smoothing Parameter of GRNN

Changing in value of smoothing factor (σ) affects performance of the GRNN congestion prediction models. Therefore, we need to know what is the best value for (σ) to use in

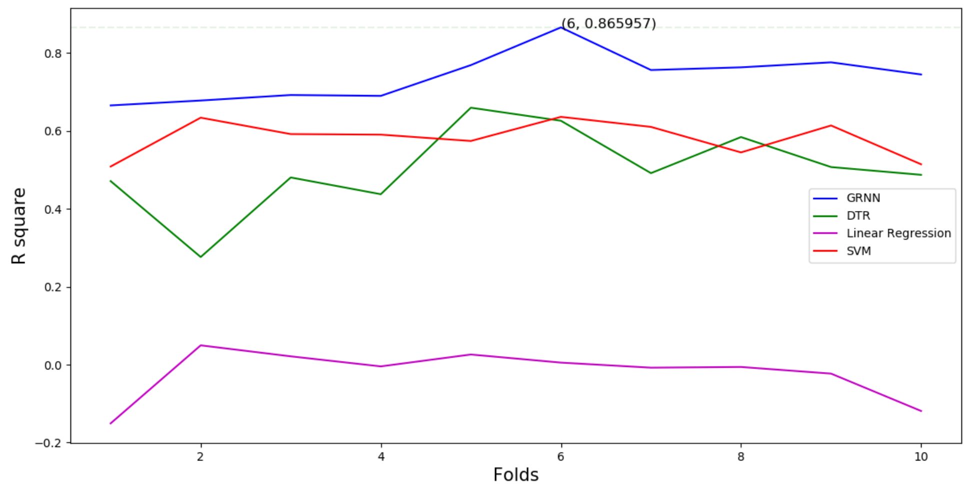

Figure 5.5 R^2 of GRNN, DTR, Linear Regression, and SVM congestion prediction models in HetVNET.

(5.6). We consider multiple values for the smoothing factor in the range of $(0 < \sigma \leq 1)$ and investigate the MSE results of applying each value of (σ) in the GRNN prediction model. In Fig. 5.6, the MSE values of GRNN congestion prediction models, in which the values of smoothing factor are 0.1, 0.2, and 0.3, are close to each other. As the figure shows, the MSE values of GRNN prediction models are risen as we increased the value of (σ) over 0.3 to 1. Fig. 5.6 shows that the best value for smoothing factor in terms of MSE is 0.2. In other words, with this value of the smoothing factor, the GRNN algorithm could predict $\hat{U}(v_t)$ with lower error in prediction results.

5.4.4 Evaluating Performance of ICAM

We consider packet delivery ratio, packet loss ratio and average delay to evaluate performance of ICAM. To generate simulations used in this section, we assume $T = 0.4$ and $\gamma = 0.2$. Since the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) is the default congestion controlling strategy in DSRC, we evaluate performance of the CSMA/CA in this paper. In [23], a "Machine Learning Congestion Control (ML-CC)" method is proposed to control congestion in VANET. In this section, performance of the ML-CC is measured against performance of the ICAM. Regarding that D-FPAV [104] is a well-known power adapting mechanism for controlling congestion in VANET, we compare performance of D-FPAV with ICAM in this paper. In [77], authors proposed a "Requirement of Safety (RoS)" scheduling technique for HetVNET based on a proposed game theory. The authors used Geographic Information System (GIS), Global Position System (GPS), Sensor network and VANET to

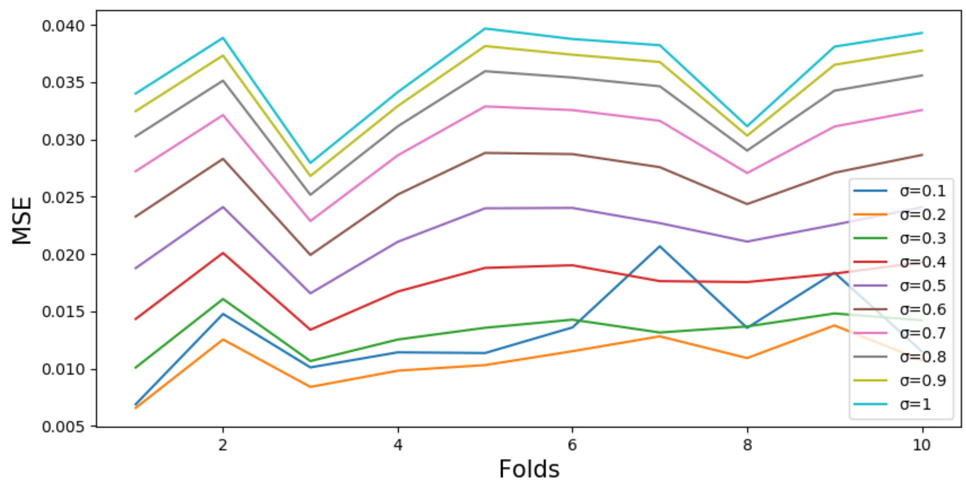

Figure 5.6 Variation of MSE with the folds for various values of Smoothing parameter of GRNN congestion prediction models.

make a heterogeneous network. Moreover, the authors in [77] applied the "Earlier Deadline First (EDF)" for IEEE 802.11p to make comparisons between their proposed method and EDF. In this paper, we compare performance of the proposed ICAM to EDF, and RoS too.

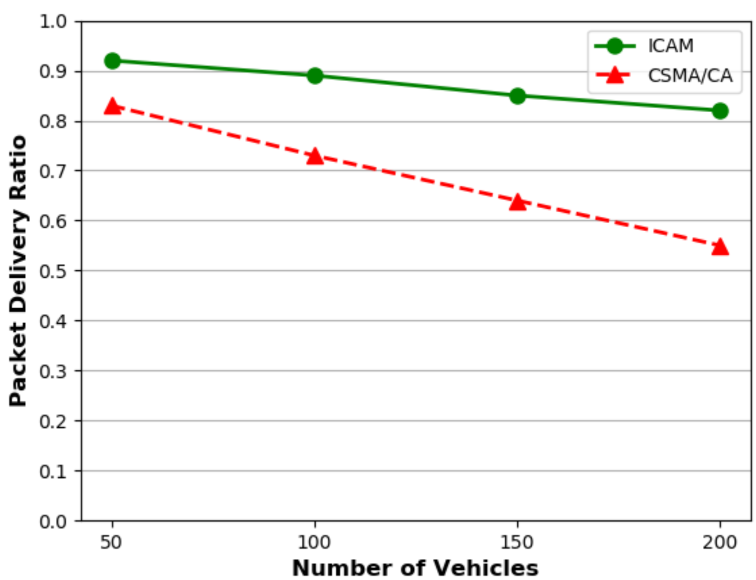

Figure 5.7 Impact of proposed ICAM on Packet delivery ratio in compare to apply CSMA/CA method.

Fig. 5.7 shows packet delivery ratios using CSMA/CA and ICAM in various vehicle densities. Based on this figure, packet delivery ratio is improved by employing ICAM. This improvement

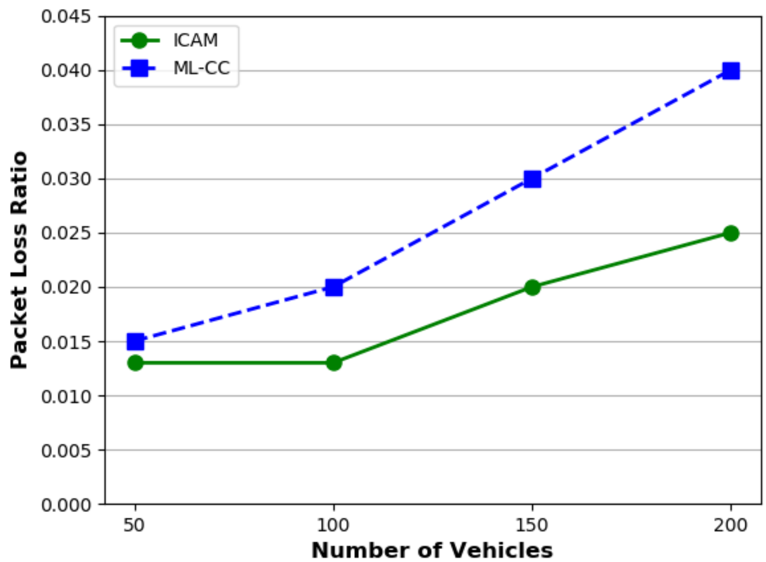

Figure 5.8 Packet loss ratio using ML-CC and ICAM.

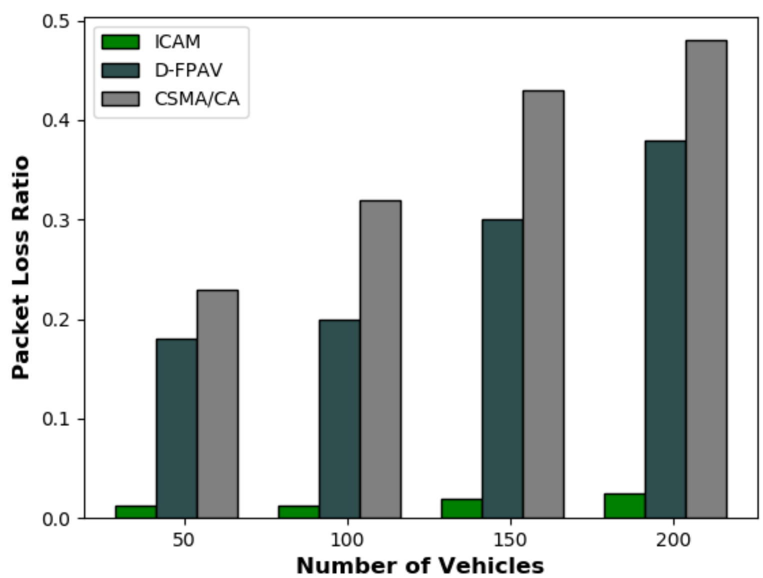

Figure 5.9 Packet loss ratio using CSMA/CA, D-FPAV and ICAM.

is notable by increasing number of vehicles, since packet delivery ratio in the simulation scenario with 200 vehicles is about 0.8 using ICAM and about 0.55 using CSMA/CA. Fig. 5.8 compares packet loss ratio of ML-CC with the proposed ICAM. As shown in this figure, by increasing the number of vehicles from 50 vehicles to 200 vehicles, the ICAM outperforms the ML-CC technique in terms of packet loss ratio. The lower amount of packet loss ratio in ICAM in comparison to ML-CC is due to intelligently adjusting the value of DSRC transmission power. The variation of packet loss ration with vehicle density in ICAM, D-FPAV and CSMA/CA is shown in Fig. 5.9. D-FPAV in compare to CSMA/CA could improve packet loss ratio, however, results of packet loss ratio for ICAM is considerably less than D-FPAV and CSMA/CA. These results indicate that prediction and avoidance mechanism in ICAM could control congestion before it occurs in the network and consequently save the network form having high number of lost packets.

Fig. 5.10 shows variation of average delay with vehicle density, where CSMA/CA, D-FPAV, ML-CC and the ICAM are used separately. As shown in Fig. 5.10, D-FPAV has better performance compared to CSMA/CA in terms of average delay. However, performance of the network in terms of average delay using ML-CC method is more better than D-FPAV, as dashed black and blue lines show in the Fig. 5.10. Above all, the proposed ICAM performs better than ML-CC, D-FPAV and CSMA/CA, especially when the number of vehicles is rising from 100 to 200 vehicles. It is interesting that the behaviour of ICAM shows stability by increasing the number of vehicles. The GRNN prediction approach is the key of this stability. The differences between the performances of ML-CC and ICAM in the both Fig. 5.8 and Fig. 5.10 are related to the concept of prediction, which is in the nature of the proposed method. In the ML-CC, there is no prediction and it is about clustering at the time which vehicles arrived at the intersections. However, in the proposed ICAM, the prediction results help adapt value of DSRC transmission power to the network situation in the future with the aim of avoiding network congestion. Therefore, results in Fig. 5.8 and Fig. 5.10 indicate that the prediction step is the strength point of ICAM in comparison to the ML-CC method.

Fig. 5.11 compares performance of ICAM to RoS and EDF methods. As in [77], two different vehicle densities are selected for this comparison, 31 vehicles and 76 vehicles. Fig. 5.11 shows stability in performance of HetVNET due to apply the proposed ICAM. In the first vehicle density of 31 vehicles, the ICAM could show a low value in average delay like 13.9 ms. For the same number of vehicles, this metric equals 152 and 270 using Ros and EDF, respectively. The average delay's results of RoS show a decrease by increasing the number of vehicles. As shown in Fig. 5.11, by increasing the number of vehicles from 31 to 76 vehicles, which considered in [77], amount of average delay in HetVNET using ICAM has been increased for just 0.3 ms. However, by growing number of vehicle from 31 to 76 vehicles, the amount of average delay remarkably increased from 270 ms to 420 ms using EDF. Indeed, the variation of average delay in ICAM is not significant by rising the number of vehicles. Although for 76 number of vehicles, the RoS could improve average delay to a value of 60 ms, the minimum value for this metric is achieved by applying ICAM. Therefore, as Fig. 5.11 shows, the ICAM outperforms the other two techniques.

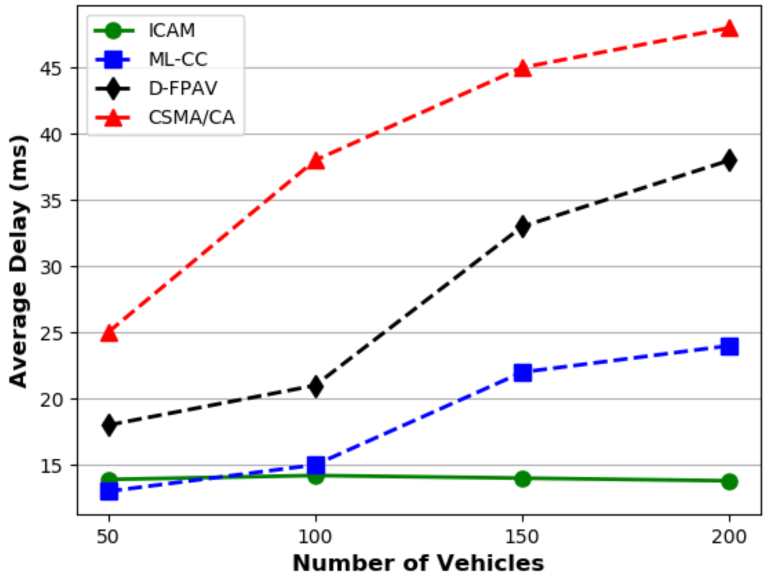

Figure 5.10 Impact of proposed ICAM on average delay (ms) in compare to use CSMA/CA, ML-CC and D-FPAV.

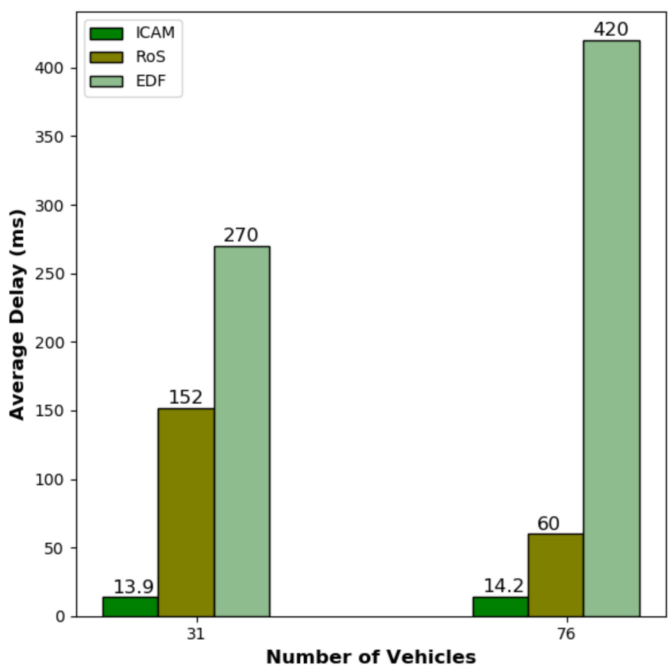

Figure 5.11 Variation of average delay with two different vehicle densities using ICAM, RoS, and EDF in HetVNET.

Based on the results shown in Fig. 5.7 to Fig. 5.11, the ICAM is a reliable method. The stability in the results comes from the two key points of the proposed congestion avoidance strategy. The first is related to the way of defining utility function and congestion states based on the target HetVNET situation. The second is about the proper using of the notion of prediction in the development of congestion avoidance mechanism.

5.5 Conclusion and Future Work

In this paper, we considered the congestion problem in HetVNET. We applied machine learning methods to predict congestion and create an intelligent congestion avoidance mechanism in HetVNET. This approach helps build an intelligent and dynamic congestion management system in vehicular networks. We proposed a congestion prediction model using GRNN, which is a supervised neural network method. K-fold cross-validation approach is used in this work. We used simulated data to train and test the congestion prediction models. We evaluated performance of the proposed GRNN congestion prediction method and compared it to the performance of MLR, SVM and DTR congestion prediction models. Based on the obtained results, GRNN congestion prediction model shows more accuracy, reliability and stability among the considered prediction methods. The result of the GRNN congestion prediction model help dynamically and intelligently adjust the value of DSRC transmission power with aim of preventing network congestion in HetVNET. Moreover, the performance evaluation results indicate that the proposed ICAM outperforms CSMA/CA, D-FPAV, Ml-CC, RoS and EDF. As future work, we will predict congestion occurrence in each of the technologies used in a HetVNET.

CHAPTER 6 ARTICLE 4: A CONDITIONAL GENERATIVE ADVERSARIAL NETWORK BASED APPROACH FOR NETWORK SLICING IN HETEROGENEOUS VEHICULAR NETWORKS

Farnoush Falahatraftar, Samuel Pierre, Steven Chamberland

Mobile Computing and Network Research Laboratory (LARIM)
Department of Computer and Software Engineering, Polytechnique de Montréal
Montreal, QC H3T 1J4, Canada
Email : {farnoush.falahatraftar, samuel.pierre, steven.chamberland}@polymtl.ca.

Status: published in *Telecom*, March 2021.

Abstract Heterogeneous Vehicular Network (HetVNET) is a highly dynamic type of network that changes very quickly. Regarding this feature of HetVNETs and the emerging notion of network slicing in 5G technology, we propose a hybrid intelligent Software-Defined Network (SDN) and Network Functions Virtualization (NFV) based architecture. In this paper, we apply Conditional Generative Adversarial Network (CGAN) to augment the information of successful network scenarios that are related to network congestion and dynamicity. The results show that the proposed CGAN can be trained in order to generate valuable data. The generated data are similar to the real data and they can be used in blueprints of HetVNET slices.

*Keywords:*vehicular network, CGAN, congestion in vehicular network, SDN, network slicing.

6.1 Introduction

The vision of having a central, robust, and intelligent network management can be achieved by taking advantages of learning algorithms in the control layer of Software-Defined Network (SDN). The network virtualization concept is about separating services and infrastructures in order to achieve the provision and development of smooth service by using virtualized network software functions [110]. In the Network Functions Virtualization (NFV) framework, a group of functions and infrastructures (e.g., hardware resources) is formed with the aim of solving network challenges and providing services to cope with critical network situations [111].

The integration of SDN and NFV into an exclusive architecture is an opportunity to take advantage of SDN features, such as having a programmable, intelligent, and dynamic network controller and manager, along with providing efficient response to highly dynamic service demands, such as ultra-low latency, reliability, and more. Gomes et al. [29] proposed an architecture in this matter by combining SDN and network virtualization. In the proposed architecture, Internet Service Providers (ISPs) could tune resource allocation among different Virtual Software Defined Networks (VSDN) that are based on user's demands.

Moreover, with the introduction of fifth Generation (5G) and beyond of mobile systems, it is expected that 5G will support various service requirements [112]. Indeed, with the widespread growth of mobile users and the emerging Internet of Things (IoT) in the near future, 5G is a promising way to connect massive smart devices in various IoT applications with diverse user expectations for quality of service. The 5G resources should be efficiently allocated in order to meet these expectations. On this point, the sharing of physical network resources with several virtual network slices that were isolated from each other was recently proposed as an approach to boosting 5G against various quality of service demands [67, 112–114].

Regarding the network traffic states, which are defined as safe, warning, and congested in [83]; in the literature, the proposed methods of network congestion recognition are mainly based on the computation of the available and the needed resources. For example, the authors in [32, 86, 101] considered the Channel Busy Ratio (CBR) to find congestion in vehicular networks using the European Telecommunications Standards Institute (ETSI) standardization. Finding an optimal value for CBR that prevents communication channel from under-utilization is a serious challenge in these works. Whenever the amount of required network resources is higher than the amount of available resources, the network management system could find out that the smooth data flow may disappear and congestion could even occur. In highly dynamic network systems, like Heterogeneous Vehicular Networks (HetVNETs), the network's topology, the number of users, the type of required services, and the amount of required resources are changing quickly and dynamically. Therefore, following constant and predefined policies for the current situation is not a promising way to provide reliable services. Because network conditions could change rapidly, applying prior generated policies and solutions is not applicable to network problems. In the light of this issue, it is worth giving dynamicity to the proposed solutions. In other words, if we have a very dynamic network, like HetVNET, then it may be a novel idea to propose a method that can provide network resources and requirements quickly and dynamically based on network templates. These templates can be used to dynamically create the HetVNET slices. Integrating Artificial Intelligence (AI) with SDN based architecture is a promising potential solution to the challenges of the dynamicity of heterogeneous networks [115]. To the best of our knowledge, in the current scientific works that are related to vehicular networks, there is a lack of an intelligent SDN-NFV based architecture that could provide network templates in HetVNET environment using computing power of fog objects. Indeed, the use of fog devices to implement intelligent methods (in the heart of SDN and NFV technologies) to ensure smooth data flow in vehicular networks is an open problem that needs to be addressed by both academic and industrial researchers jiacheng2016software,davy2014challenges.

With regard to the above-mentioned open problems and considering the role of 5G in building IoT use cases and the advantages of using SDN and NFV technologies, the following challenging research question that is related to the HetVNET environment may arise.

- When considering the advantages of SDN and NFV concepts and the notion of network slicing, how can we design an architecture that provides reliable information to create HetVNET slices?

On the basis of the research question and the direct relation between the congestion problem and the resource allocation problem, we propose a novel method that is based on Deep Learning and network slicing technique with the aim of avoiding congestion in HetVNET. Therefore, considering network congestion problem in a high dynamic network environment, like HetVNET, taking advantage of global network view of the SDN, and using information from past successful network experiences, can help us to create an intelligent SDN architecture. In this paper, we apply the Conditional Generative Adversarial Network (CGAN) to augment the data used in creating network slices in HetVNET; and additionally, propose a centralized SDN based architecture with the aim of enhancing flexibility and adaptability of the HetVNET.

CGAN is widely used for text, image, and video generation and prediction works. We will show how the proposed CGAN helps us to intelligently and reliably generate valuable information in order to create network slices with the aim of avoiding congestion in HetVNET. It is the first time that CGAN has been applied in this context to the best of our knowledge.

6.2 Related Work

Network slicing has been recently considered in a number of research works, particularly for vehicular networks. In [116]; the authors gave priority to the safety data. A queuing method was used to categorize the traffic data based on the priority. Furthermore, they applied deep neural network methods for resource allocation in VANET. However, the structure of data set in terms of features, size, and output classes was not clearly explained. Based on the results, Long Short-Term Memory (LSTM) performs better than both Convolutional Neural Network (CNN) and Deep Neural Network (DNN). Unfortunately, the time-based results, such as resource allocation time and detection time, have not been provided. In addition, no further information was provided on the amount of bandwidth dedicated to each of the priority queues and how this amount of bandwidth can be assigned to the vehicles.

Network slicing requirements for Vehicle-to-everything (V2X) communications are investigated in [117]. The authors believe that, apart from infrastructure and management layers, a network slicing V2X architecture also needs to be designed while using business and service layers. These two layers must be considered to determine which services can be provided and which use cases can be supported for each network slice.

In [118], the k-means++ technique is proposed for clustering the services based on the Service Level Agreement (SLA). After clustering the services in three categories of traffic safety, traffic efficiency, and information services, then multiple services are assigned to each slice. In addition, Shared Proportional Fairness Scheme (SPFS) is proposed to schedule network slices based on fairness in resources utilization. Based on the results, wireless resource utilization rate could be improved using the linear programming barrier method and slice scheduling approach.

In [48], the Euclidean distance method is used to find similar vehicles in terms of weak Quality of Experience (QoE). Subsequently, the similar vehicles are gathered in the same cluster. A vehicle that has better QoE and stronger communication links is pronounced as the leader in the slice. Moreover, the Lyapunov optimization technique was applied to provide video frames by the leaders for the other vehicles in the slices. Based on the results, in the slices with high vehicle density, the quality of the selected video is low and the QoE in these slices is reduced. The Lyapunov optimization algorithm selects the low quality videos to guarantee the stability of streaming video. Because the radio resources could not support the high number of vehicles in the slice, the low quality videos are selected by the Lyapunov algorithm. Although it is a wise technique for selecting video data, it could also improve the communication link capacity by dynamically changing the size of the slices and making new slices with a lower number of QoE vehicles.

In [119], the authors proposed an intelligent cloud based architecture for network slicing in vehicular networks. In this architecture, a deep reinforcement learning method is proposed to be applied in the control layer. Collecting, storing, and analyzing the huge data are to be done at the control layer. However, these tasks require powerful computing and storage resources; otherwise, the performance of the intelligent method will be negatively affected. Moreover, using cloud for these tasks is not an optimal solution; instead, the fog devices could be a better choice for these challenges.

In [120], the author proposed a stochastic method for network slicing scheduling and resource allocation. Markov and Lyapunov methods were used for the nonlinear stochastic optimization problem of resource utilization in vehicular networks. The proposed method optimized the value of transmission power and the value of transmission rate in network slices. Indeed, the proposed method is mainly based on tuning the resource allocation and transmission power in the network slices, without any future forecast in the network situation and applying AI methods. Based on their study, computational complexity is polynomial (cubic), which takes time to converge. This indicates that the stochastic solutions may not be appropriate for such a dynamic and fast-changing vehicular network, unless they are applied to strong computational devices, such as fog devices in vehicular networks.

In [121], Signal to Interference plus Noise Ratio (SINR) has been measured for every vehicle in the predefined infotainment and autonomous network slices. Subsequently, the vehicles with higher amount of SINR provide streaming video services for vehicles with lower amount of SINR. The obtained results show an improvement in the throughput of infotainment slices and reduction in packet reception ratio for safety applications of autonomous slices.

In, [122], the authors proposed a network slicing framework for Internet of Vehicle (IoV) based on multiple types of Radio Access Technologies (RATs) and traffics, using cloud computing. The results obtained from real deployment showed the scalability of the proposed framework. However, the application of AI methods to provide a flexible predicted resource allocation mechanism could significantly improve the results. This issue has been considered in [123, 124]; however, the vehicular network is not the targeted network in these studies. In [123], the authors proposed the dynamic resource allocation technique using deep reinforcement learning. The proposed method showed better performance when compared to the other techniques, such as heuristic and random approaches. However, based on the results, slicing performance is negatively affected by an increase in the network load. In [124], deep reinforcement learning was been applied to allocate resources in the network slices. In [125], the authors showed that the network slicing could benefit from the proposed traffic prediction method using AI.

In [126], the authors proposed a method for network resources allocation with respect to the traffic load in vehicular networks. A Monte Carlo tree search utilizing cross entropy with new metric was proposed by the authors. Although the proposed method is not based on prior network information, it shows good performance in simulation scenarios with a large of fog devices and network slices. However, in simulation scenarios with a low number of fog devices and network slices, the performance of the proposed method was not as good as any other comparable method while using Q-learning. Besides, the Mont Carlo tree search requires a lot of memory and it is slow to perform, which is evident in the results of the operation time.

Therefore, the proposed method is not a worthy choice to be applied in a dynamic vehicular network that changes fast. Moreover, it might be better to use more intelligent methods that could be based on analyzing previous network experiences and information than the Mont Carlo tree search method.

In [127], the authors proposed a method for defining clusters of vehicles and vehicle slices of network. The clusters are used to exchange safety related messages via Vehicle-to- Vehicle (V2V) communications. The network slices are used to transmit video streaming data via Vehicle-to-Road Side Unit. From the results, the proposed method performs better in the scenario with a low number of vehicles. This indicates that, with the increasing number of vehicles and crowded urban roads, there is still a problem with the transmission of high data loads, even using the proposed method.

Based on the literature, there are two groups of proposed approaches for network slicing in vehicular networks: non-intelligent methods and intelligent methods. In this section, we explained some of the drawbacks of some studied non-intelligent works. The proposed intelligent methods are mainly based on the machine learning and deep learning methods. Considering vehicular networks, the number of works that used AI methods in the proposed network slicing approach is limited. As mentioned earlier, locally analyzing large data by AI methods requires powerful computing resources that can be provided using fog computing. Proposing a new network slicing method that can intelligently and dynamically change the network slices could therefore have a significant novelty in terms of dynamic resource allocation and HetVNET configuration.

6.3 Methodology

6.3.1 Generating Dataset Using Simulation Scenarios

In this paper, the following five variables were considered: the number of vehicles (v), data rate (dr), DSRC transmission power (tp_{DSRC}), LTE transmission power (tp_{LTE}), and LTE bandwidth (b). Based on the studied literature related to ETSI and the Wireless Access in Vehicular Environments (WAVE) standardization, these parameters have the most effect on network congestion problem [32–38]. Because no data sets from real HetVNET are available today, we have generated a data set that contains data records of these parameters [83]. We used OpenStreetMap (OSM) [90] to have a map that is similar to the real environment in terms of intersections, streets, buildings, etc. Figure 6.1 shows the boroughs of downtown Montreal that we used to generate the simulation scenario. Besides, to generate the road traffic and vehicle movements on the map, we used Simulation of Urban Mobility (SUMO)

0.26.0. [88]. SUMO requires ".osm” file created by OSM to generate the vehicle traffic on the map. Simultaneously, Veins LTE version 1.3 [89] simulator was used to generate heterogeneous vehicular network using IEEE 802.11p for V2V communications and LTE for Vehicle-to-Infrastructure (V2I) communications. SUMO and Veins LTE worked on Ubuntu 16.04. The simulation time was 1000 s and, in order to generate a high load of data, we defined a road accident with a 70 s time of running simulation scenario. Because we considered the downtown of Montreal city, the vehicles maintained the maximum speed limit of 40 km/h assigned to these boroughs. Table 6.1 shows the parameters and their corresponding values used for simulating HetVNET scenarios. In each run, the values of the five attributes were changed based on the information presented in Table 6.1.

Table 6.1 Simulation parameters and corresponding values.

Parameter	**IEEE 802.11 p**	**LTE**
Number of Base Station		1
Number of Resource Blocks		25, 50, 100
Bandwidth	10 MHz	5 MHz, 10 MHz, 20 MHz
Transmission power	30 dBm (Maximally)	43 dBm, 46 dBm
Transmission data rate	6-27 Mbps	
Modulation techniques	QPSK, 16-QAM, 64-QAM	
Simulation time	1000 s	
Simulation runs	500	
Number of vehicles	50, 100, 150, 200	
Simulation area	1000 m × 1000 m	
Number of lanes	4 (two in each direction)	
Maximum speed	40 km/h	
Propagation model	Nakagami	
Size of message	400 Bytes	

Information was extracted after running simulation scenarios of HetVNET. The amount of network throughput over generated data rate can give us a vision of the network performance, as explained in [83]. Indeed, based on this value, we can find out how many of the generated data were successfully received at destination points per unit of time. Therefore, for each run of simulation scenario, the amount of network throughput over load per a unit of time is extracted. This value can be equal to one maximally in the desired case, and equal to zero minimally in the worst case of network performance. Network throughput is the calculated sum of the value of throughput in DSRC and the value of throughput in LTE. To meet the paper objectives, we used information of successful experiences (simulation runs). Simulation scenarios with a network throughput value over the generated data rate of more than 0.6

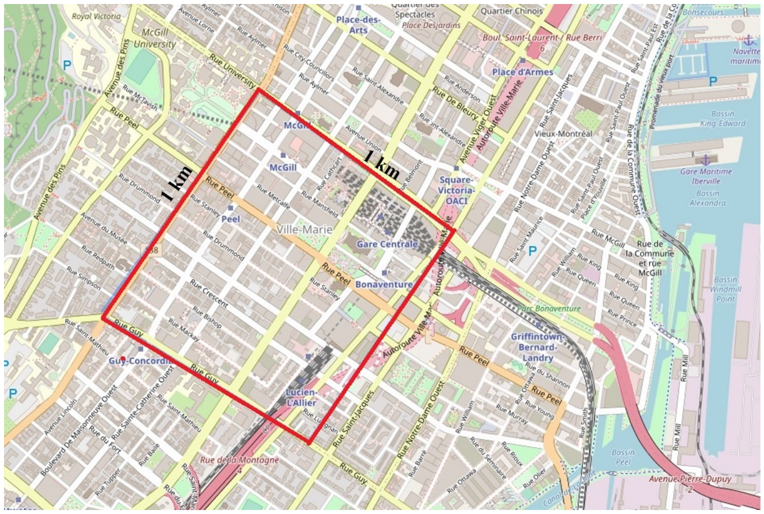

Figure 6.1 Simulated area is shown by a red square.

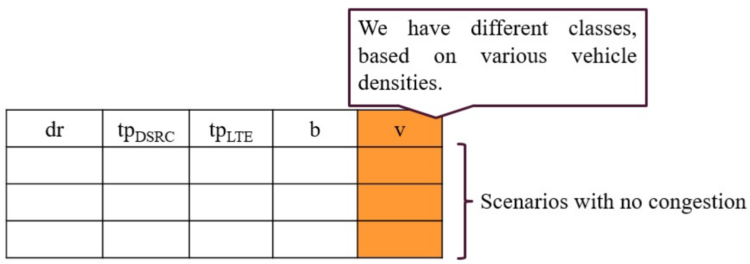

Figure 6.2 Structure of data set used for the Conditional Generative Adversarial Network (CGAN).

are selected as a successful scenario [92]. In fact, we need to extract the best scenarios in HetVNET and augment these successful scenarios. Subsequently, we put the information of these scenarios in the dataset. As shown in Figure 6.2, the data records in the dataset are classified based on the number of vehicles. Based on the five vehicle densities considered, there are five different classes of vehicles in this paper: 30, 50, 100, 150, and 200 vehicles.

6.3.2 Proposing CGAN Model for HetVNET

CGAN uses a generative model, like G, to create new data from noise [128]. The noise is a random data which is similar in structure to real data. The task of the discriminator is to recognize which data are real or not, and come from the generator. In the CGAN, the generator and discriminator have access to class labels, like v, as shown in Figure 6.3.

The discriminator tries to maximize the probability of having a correct label (real or random) in the output. At the same time, the aim of the generator is to make the data very similar to real data, thus misleading the discriminator. In other words, based on the feed backs received from the discriminator, the generator will train time by time, until the discriminator is unable to identify which data are real and which are not (random). Therefore, the generator, like G, and the discriminator, like D, play a min-max game [128], as follows:

$$\underset{G}{min} \ \underset{D}{max} V(D,G) = E_{x \sim p_{data(x)}}[logD(x|v)] + E_{z \sim p_{z(z)}}[log(1-D(G(z|v)))], \tag{6.1}$$

where, $p_{z(z)}$ is prior input noise variable, and $D(x|v)$ is the probability that x with a class label of v is a real data. We assume that z is random data generated by the generator using noise; therefore, $D(G(z|v))$ is the probability that the random data composed of noise and class label comes from the real dataset. From the first moments of learning, the discriminator can easily recognize the random data from the real data. Therefore, the value of $D(G(z|v))$ is low and, as a result, the amount of $log(1-D(G(z|v)))$ is large, which is a desired situation for the discriminator. However, after a while, when the generator trains and algorithm converges, this amount will be minimized.

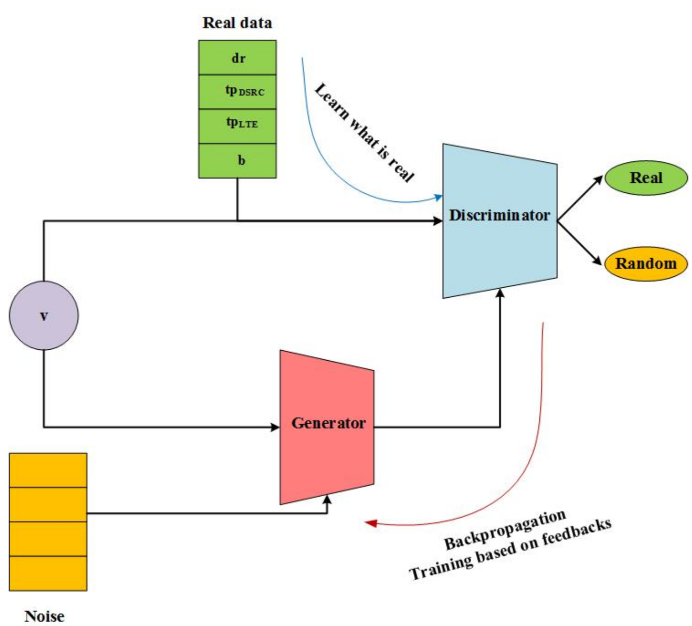

Figure 6.3 CGAN model for HetVNET.

A uniform distribution $U(0,1)$ is used to generate random noise while using the generator. Moreover, we used the Adam optimizer [94], which has high convergence speed and is faster than the gradient descent. Batch normalization is applied in the generator in order to avoid the problem of vanishing gradient in back propagation. Indeed, a batch normalization layer is added after each hidden layer's activation function, with a momentum of 0.9 being recommended in [94], as a suitable value. Furthermore, Leaky ReLU is preferred as the activation function (the learning rate is 0.001), since it is effective in reducing the run-time latency [94]. Besides, He initialization is applied, which best fits the Leaky ReLU activation function [94]. Tanh and Sigmoid activation functions are used for the output layer in the generator and the discriminator, respectively. Figure 6.4 shows the layers of the CGAN.

Figure 6.4 The architecture of the proposed CGAN that shows generator's layers and discriminator's layers.

6.3.3 A Hybrid CGAN-SDN Architecture

HetVNET is a dynamic and stochastic network with a rapidly changing number of users, network topology, and required services. The volume of generated data using safety and infotainment application varies very fast and it is difficult to predict. Moreover, most of the network communication features, such as signal strength, signal attenuation, and path link stability, are vulnerable from other parameters, such as temporary and permanent barriers, vehicles movements, speed and direction. As far as these features of HetVNET are concerned, it is mandatory to have a central intelligent management mechanism in order to meet the requirements of the network. This issue is comprehensively studied and proposed as an open challenge in [115].

Because the velocity of change in HetVNET and requirements is high and varies, using the same slices for a long time is risky; therefore, the ability of the slices to meet all of the network requirements is not guaranteed. Instead, we can generate network slices based on previous successful network experiences. It is like making new generation of HetVNET slices from a strong population. With this vision, on one hand, the risk of having slices with low performance is reduced and, on the other hand, it affects the performance of the entire HetVNET over time by creating a good generation of slices. When considering the literature related to network congestion in vehicular network, using WAVE and ETSI standards, the proposed congestion controlling mechanisms are mostly based on tuning the value of transmission power and data transmission rate [32–38]. Finding the optimal value for these parameters is not a promising solution, due to the high dynamicity of the vehicular network. In other words, there is no optimal fixed value for these parameters, instead, the appropriate value should be applied temporarily for these parameters during a predefined time interval and just based on the current network situation. These values are similar (not equal) to the values that were previously used and made successful network experiences. In this work, the synthetic data generated by the proposed CGAN are the data close (not equal) to the original ones that came from the successful scenarios. Therefore, we create the most similar data with little variance to the previous successful ones to apply in network slices. When considering the wide applications of Generative Adversarial Network (GAN) in producing new images, by our proposed approach, we can generate several images for each part (slice) of vehicular network (let us assume that, in this problem, images are the network various configurations in terms of the five variables). These generated configurations can be applied in various slices with different number of vehicles. Therefore, providing network slices with the appropriate value for the five parameters that are tailored to the current network situation in terms of the number of vehicles is proposed in this work. The proposed CGAN method can quickly generate a volume of information useful in creating and setting many HetVNET slices with respect to the network service needs. When considering the five variables of v, dr, tp_{DSRC}, tp_{LTE}, and b and their possible real values, in each time, the number of possible templates for the slices can be calculated by multiplying the number of possible values that each of these parameters could have. Subsequently, these templates can be categorized based on the number of vehicles.

With the NFV, the necessary and common network functions and services can be deployed and installed in the virtual servers. The Virtual Network Functions (VNFs) are the software functions, which provide network services and functionality virtually. NFV is interesting for network providers, as the installation and updating of the virtual functions using software is less costly and easier than on hardware.

Besides, the Next Generation Mobile Network Alliance (NGMN) [10] proposed network slicing as a method for virtualizing the 5G network physical resources. In this method, each slice is independent of the other slice, and it is formed based on a blueprint. Network functions and resources are used to provide specific service(s), such as low-latency, ultra reliability, etc., by a slice.

In a dynamic environment, like HetVNET, in which topology, the number of users, and the required services are inconsistent, network slices may have to be created and modified very fast. Therefore, it may be necessary, within a period of time, to replace the slices with new slices. The rapid provision of new slices based on the current network requirements needs a plan that could be made up of the necessary metrics, such as transmission power, transmission rate, and required bandwidth. Regarding the above mentioned issues, we propose a hybrid CGAN-SDN architecture.

Figure 6.5 shows the three layers of SDN. In this architecture, fog devices are implanted at the controller in order to execute the CGAN algorithm. The high computing and storage abilities of fog devices are helpful for running the CGAN. The NFV is a fundamental component of network virtualization and the creation of slices. Information extracted from the successful scenarios is augmented at the controller, and this information is useful for making blueprints. The NFV has the required functions to establish the slices. Therefore, the blueprints are passed to the VNFs by NFV management and orchestration.

At the infrastructure layer, based on the orders coming from the virtual functions, the Virtual Network Provider (VNP) must form the necessary slices. Each slice has specific value of data rate and transmission power for DSRC and transmission power and bandwidth for LTE. Because the DSRC is used for V2V communications, the vehicles must communicate with each other using the new value of transmission power and data rate in each slice. Moreover, we can implement several Radio Access Network (RAN) base stations (like road side units) for LTE in the HetVNET. Therefore, these base stations can be configured in terms of transmission power and bandwidth for each slice based on the values that are decided at the control layer.

Because each vector $\mathrm{X} = (v, dr, tp_{DSRC}, tp_{LTE}, b)$ generated by CGAN is applying to create one slice, even if there is more than one LTE RAN station in the slice, all should be configured while using the value of tp_{LTE} and b in the vector X. Accordingly, the value of dr and tp_{DSRC} should be applied for each V2V communication in the slice. For example, when considering slice 1 in Figure 6.5, if $\mathrm{X} = (x_1, x_2, x_3, x_4, x_5)$, then the both LTE RAN stations must provide x_4 dBm transmission power and x_5 MHz bandwidth, and vehicles can use both LTE RAN stations depending on the distance between vehicles and the LTE RAN station. In addition, in

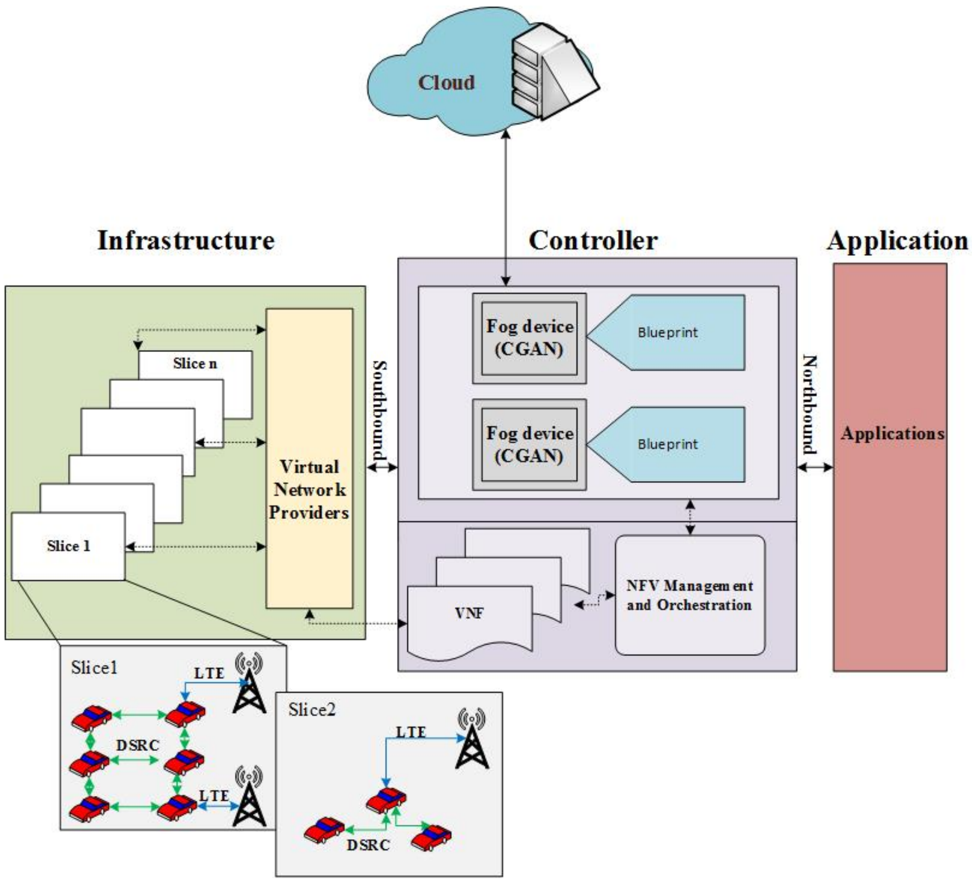

Figure 6.5 An intelligent hybrid CGAN-Software-Defined Network (CGAN-SDN) architecture for HetVNET.

this slice could be x_1 vehicles that should transmit data with x_2 Mbps transmission rates and x_3 dBm transmission power for V2V communications. Therefore, based on this method, slices are defined using the five parameters. Changing in the value of any of the five parameters (based on the new vector X generated by CGAN) means that the previous slice is gone, and a new slice is created.

In a HetVNET, each slice could vary in the size and number of vehicles. Moreover, a VNP is virtually connected to the real network providers. The VNP is authorized to modify and allocate network resources from ISP.

6.4 The Proposed CGAN Model Performance Evaluation

We used Python version 3.6 to simulate the proposed CGAN model. The performance of the proposed CGAN model was evaluated based on the performance of the discriminator.

Because CGAN uses min-max non-cooperative game, if the generator wins, then the discriminator loses. Therefore, the discriminator's performance not only evaluates the discriminator, but it also illustrates how much the generator could improve itself during the training phase. Note that the real data are not at all accessible to the generator, therefore the generator that is only allowed to learn from the gradients comes back from the discriminator via back propagation.

Figure 6.6 shows the accuracy of the discriminator over 500 epochs. In the first 50 epochs, the discriminator could distinguish between random and real data. However, after that, the discriminator made mistakes in finding the random data sent from the generator. This illustrates the improvement of the generator in making data from noise which are similar to real data. At the same time, the discriminator was still successful in finding real data. However, in epoch 300, the CGAN model reached a converging point where the discriminator could not very well recognize the real data. At this point in time, and as shown in Figure 6.6, the accuracy of the discriminator is reduced to approximately 50%, which indicates that the discriminator randomly generates output labels. Therefore, the generator trained itself very well while using feedback from the discriminator, and finally the CGAN model converges at this point in time.

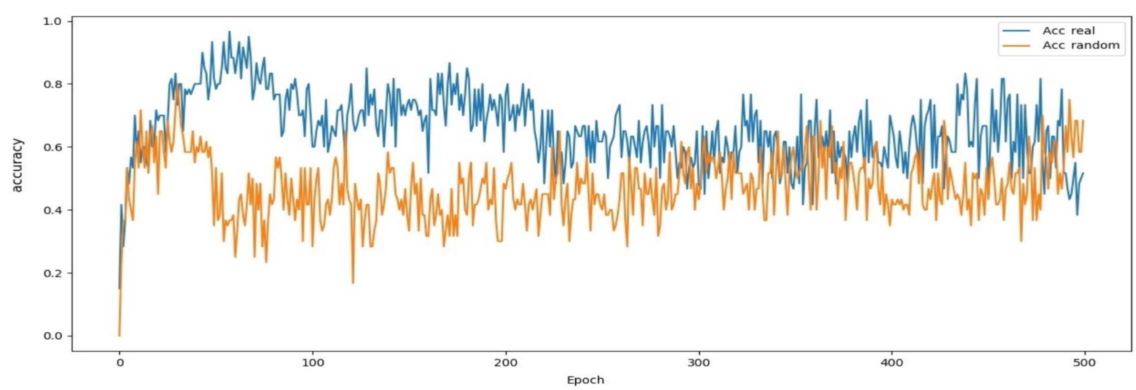

Figure 6.6 The discriminator's accuracy over time.

CGANs with two and three hidden layers were separately considered and, for each, the accuracy and loss of the discriminator was evaluated using three batch sizes of 20, 40, and 60. Finally, Figures 6.7 and 6.8 present the results. Figure 6.7 shows the variation in the accuracy of the discriminator at the converging point of the CGAN for real and random data, whiel using two and three hidden layers and three different batch sizes. The best state for a CGAN is when the discriminator randomly makes labels [94]; therefore, the accuracy of the discriminator will be approximately 50% for both the real and random data. As far as this is concerned, and as Figure 6.7 shows, this has happened with 40 batch size and two hidden layers. At this state of the converged CGAN, the discriminator with 55% accuracy guesses whether the data are real or random.

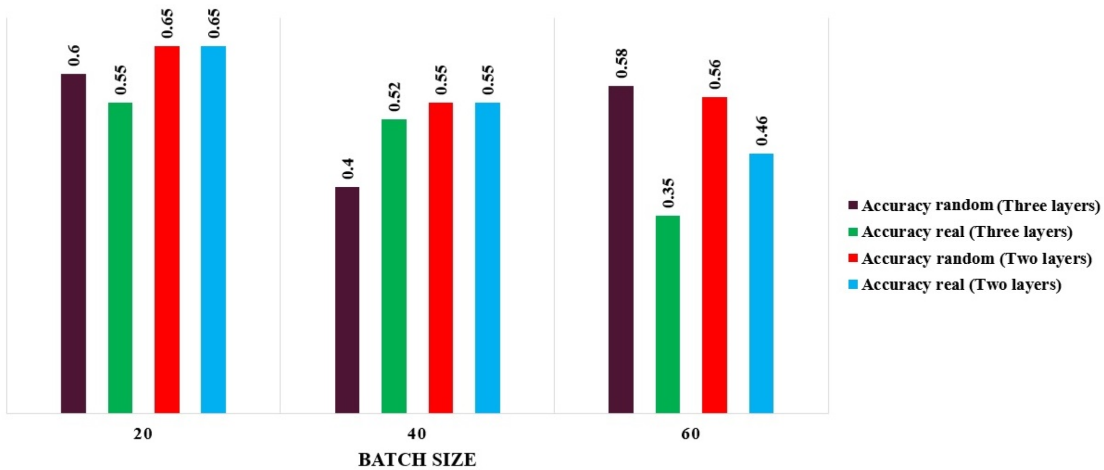

Figure 6.7 The discriminator's accuracy per various batch sizes and hidden layers.

We use binary cross-entropy to calculate the loss of the discriminator. The best state is when the loss of the discriminator is at the lowest value for both real and random data. The loss of the discriminator for three hidden layers of 40 batch size is a good value like 0.65; however, the loss for random data is as high as 0.75, as shown in Figure 6.8. In other words, the discriminator has more faults in finding random data and performs better in recognizing real data. This is a green sign that indicates that the generator is well trained. However, as compared to the same batch size, but with two hidden layers, this is a hasty conclusion. Because, with the same batch size and two hidden layers, the loss values for real and random data are very close 0.68 and 0.69, respectively. These values show that the discriminator has almost the same performance in labeling the real and the random data. Therefore, we can say that the discriminator has an error in finding both the real data and the random data at a same level. This indicates that the generator is well trained in making the discriminator's mistakes.

Figure 6.9 compares the training time that is required by the proposed CGAN to reach a converging point for two and three hidden layers using different batch sizes. The training time for the CGAN using a batch size of 60 and two hidden layers is the lowest, with a value of less than 10 s. The reason for this low training time could be related to the high amount of data that the CGAN has during the training phase as compared to using the other batch with small sizes.

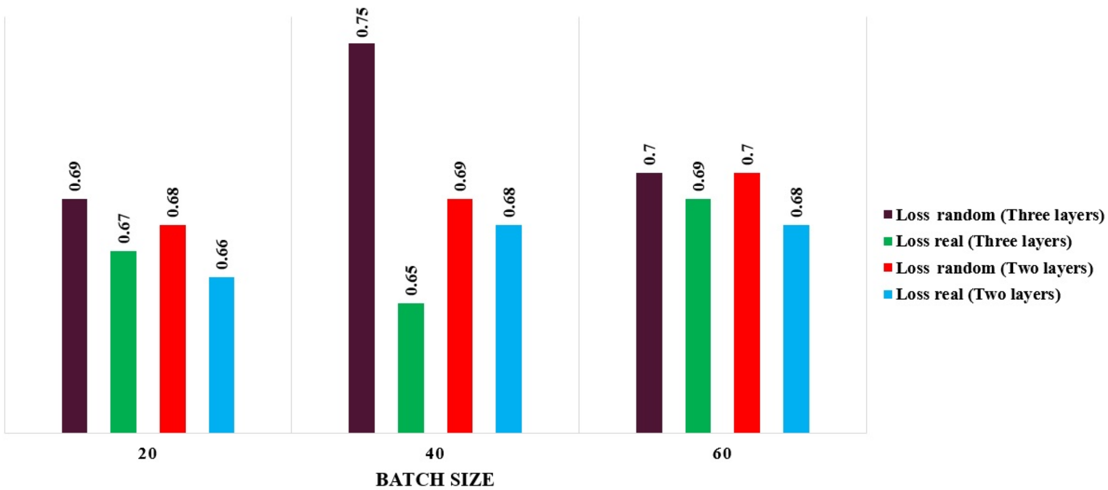

Figure 6.8 The discriminator's loss per various batch sizes and hidden layers.

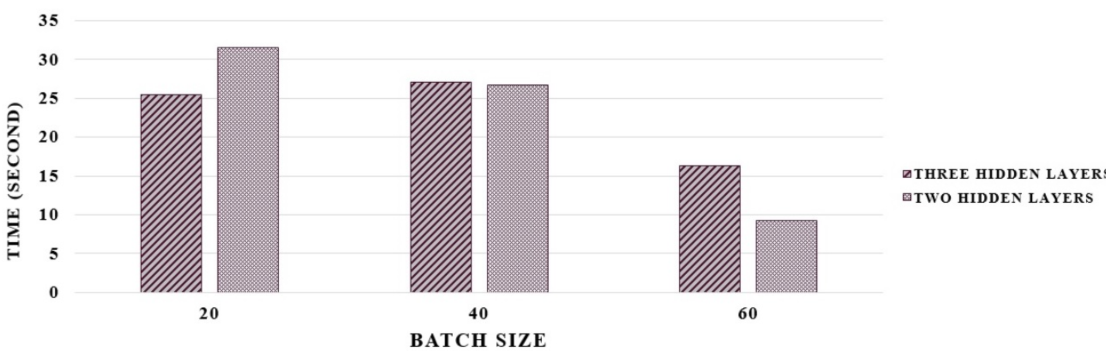

Figure 6.9 Proposed CGAN model's training time.

Based on Figure 6.7 to Figure 6.9, we can infer that using a batch size of 40 with two hidden layers is a good setting for the proposed CGAN. If there is a strict time restriction for the network, it may be better to consider a 60 batch size for the proposed CGAN.

Because we could not find the same paper in the literature as the benchmark, we could not compare the results with other HetVNET-related works.

6.5 Conclusions

When considering the dynamic nature of HetVNET, the HetVNET slices must be rapidly generated and modified. Besides, the slices must be provided in accordance with the user requirements. In this paper, we proposed an intelligent hybrid CGAN-SDN architecture for network slicing, while considering the network congestion problem in HetVNET. We proposed a CGAN based method for augmenting the data used in dynamically creating HetVNET slices. In this method, information on a number of network metrics, which have a significant impact on the smooth flow of data, has been extracted from network scenarios with a good performance and results, in terms of ratio of throughput to data generation rate. Subsequently, these data records are classified based on various vehicle densities, and the proposed CGAN method is applied to generate similar information. The augmented information can be used in the control layer of the proposed intelligent hybrid CGAN-SDN architecture to dynamically generate HetVNET slices. We evaluated the performance of the CGAN method and, based on the obtained results and discussions, the proposed CGAN method is a reliable way to generate data that are similar to the real data.

In the future, we will apply a reinforcement learning method to propose an agent in a hybrid SDN-based architecture, which can intelligently produce network slices.

CHAPTER 7 GENERAL DISCUSSION

In this chapter, the proposed methods are discussed. In vehicular networks, network congestion and road congestion are two concepts that should be distinguished from each other. Road congestion can occur due to several factors such as accidents, road closure, special events, rush time of day, special weather conditions and many more. These factors are not networking factors. We can recognize road congestion when we see that the road is closed and the vehicles movements are very slow or even they stop. However, we do not have the same approach to find out that a network congestion occurred. Thus, we need to infer network congestion from the performance of the network. Predicting road congestion using machine learning and deep learning methods has been considered by authors and valuable works have been done in this area by the researchers. However, progress in the field of applying machine learning and deep learning methods to predict network congestion in vehicular networks are still insufficient and this area of research has open challenges for researchers [1]. Congestion in the network has a negative impact on network throughput. It also increases the packet loss and decreases the data delivery ratio in the network [1]. Therefore, network throughput and data delivery ratio are considered to infer congestion in the HetVNET.

Transmission power and data transmission rate have considerable impact on the congestion in the network [1]. Range of transmission increases by increasing the amount of the transmission power. Consequently, the number of vehicles that can receive the messages rises, and it might lead to more collisions in the network, specially in a dense vehicular network [96]. High transmission rate means more packet transmission rate. Moreover, the high data rate needs high transmission power in order to have a good result in network performance [50]. Therefore, tuning the amount of transmission power and the value of data transmission rate has been considered in existing congestion control strategies [1]. Number of vehicles is another parameter that is important in the network congestion problem. Growing in number of vehicles leads to more data generation and more traffic load in the network and could be the cause of congestion in the network. Hence, in this dissertation, the number of vehicles, the transmission power and the data rate have been considered as parameters that affect the congestion in the network.

The warring states in terms of congestion in the HetVNET are defined, using DDR and RSS. These states are considered as classes in Chapter 3. Several supervised machine learning classification methods such as SVM, KNN, Random Forest and Naive Bayes are investigated in this dissertation. However, based on the performance evaluation results of the algorithms, Naive Bayes is considered for the proposed classification problem in this dissertation. Performance of Naive Bayes classifier shows that it is faster and more accurate than the other considered algorithms. A centralized management unit segments the roads. Estimating the vehicle density for each segment in the next time t and making classification prediction for congestion warning state of HetVNET, are the tasks of the fog congestion predictor unit in a cloudy-fog architecture of HetVNET.

Network throughput and data generation rate are components of the proposed utility function in Chapters 4 and 5. The proposed formula of the utility function shows the current situation of the network in terms of smooth data flow in the HetVNET. However, the aim is to predict the value of the utility function (without using the proposed formula of the utility function) with considering the five predictors. Therefore, this problem is defined as a regression problem and the supervised regression prediction methods such as Multiple linear regression, SVM, Decision Tree and GRNN are investigated in this dissertation. In 5, the obtained results show that the proposed GRNN is more accurate and more reliable than the other considered algorithms. Therefore, GRNN is selected to apply in the proposed intelligent congestion avoidance mechanism. As results indicate, the neural network predicting approach could prevent the network from a congestion situation and notably improve the network performance.

Network slicing with the aim of dynamically providing network slices while considering degrading network congestion in the created slices are presented in Chapter 6. CGAN is a deep learning method which is widely used in image, audio, video and text generation. Augmenting data of past successful experiences in terms of occurring congestion in the HetVNET are considered in this chapter. CGAN is applied to augment the data used for configuration of new network slices. Since in CGAN the generator and discriminator are playing against each other, we can see instabilities in the results during the training phase, in fact, they push against each other continuously. This issue is a natural behavior of GAN methods [94]. However, at the point that this unstable manner in generator and discriminator comes around 50% in terms of accuracy (converged point) we can trust that the model is trained well, since at this point, the CGAN can recognize noise data (random data) and real data randomly.

Analyzing data is a key part of supervised algorithms. In this dissertation, data is created using simulators. SUMO is used to generate vehicle movements and road traffic. In parallel, Veins LTE used to generate network environments and create data traffic in the network. In this part of the dissertation, other network simulators such as NS3 and NETSIM could be used. However, it was important that all the vehicles had the opportunity of using DSRC and LTE at the same time. Regarding this issue and after investigating the network simulators, the Veins LTE simulator is selected for doing this task. Veins LTE is an extension of OMNET ++ and is released to simulate the heterogeneous type of vehicular network.

CHAPTER 8 CONCLUSION

8.1 Summary of Contributions

Congestion controlling mechanisms are based on measuring the current state of the network. For example, defining a threshold for the busy level of the communication channel or putting a threshold for the number of lost packets in the network. The congestion controlling mechanisms run when the threshold is met. This strategy is the main structure of the current congestion control mechanisms.

The use of AI-based methods in various fields of science has been increased due to their ability of learning from data and solving the problems. Accordingly, the network congestion problem in HetVNET is not an exception. Autonomous networks are able to self-diagnose, self provisioning and self-healing thanks to AI methods. Toward creating a self adaptive and autonomous HetVNET, proposing AI-based approaches to predict and control the network congestion before it occurs in the HetVENT is the main body of the contributions in this dissertation. The adaptive network is a stable network, since it could estimate the problem before happening and adapt itself with the future challenging situation. Stability in the performance of the network that applies the proposed methods is another contribution.

Analyzing huge data generated by vehicular users needs high storage, significant computing power and functionality requirements. Recent cutting edge technologies such as SDN, fog computing and NFV provide the possibility of applying AI-based methods in the network more than before. Programmability, having a global view of the network and agility from the SDN along with high power of fog computing devices in computation, storage and networking will provide a perfect package of answers to the requirements toward having intelligent congestion control management and using AI-based approaches. Designing hybrid architectures using the named technologies with the aim of having intelligent congestion management in HetVNET is another contribution in this dissertation.

The congestion prediction problem is proposed in both forms of classification and regression problem. According to these two named types of problems and using supervised machine learning algorithms, congestion prediction methods are investigated and proposed in this dissertation. Here, Naive Bayes method is proposed to predict the warning class of network congestion in HetVNET. The AUC in the ROC curve for the proposed classification Naive Bayes method is 94% which is a good value and it also could show a significant accuracy of 91.87% in the results. In the regression congestion prediction problem, the GRNN prediction model shows the lowest RMSE value of 0.07 and the highest R square value of 0.86 among other considered prediction methods. Obtained results show that the proposed intelligent congestion avoidance mechanism using GRNN could improve packet loss ratio and average delay in comparison to other mechanisms (such as CSMA/CA, D-FPAV, ML-CC, RoS, and EDF), especially for the high number of vehicles.

Finally, applying a deep learning method for network slicing in HetVNET is considered in this dissertation. Regarding network slicing in vehicular networks, proposing a CGAN method to augment data of past positive experiences in terms of not having congestion in the network; then, applying the CGAN in a SDN-NFV architecture are other contributions in this dissertation. Generator and discriminator are the two components of the CGAN, and they play a min-max game against each other. The CGAN is converged at the point that the amount of discriminator's accuracy is around 50%. The CGAN converged after 300 epochs. Moreover, based on the results the performance of the CGAN with two hidden layers is better than with the three hidden layers.

8.2 Limitations

The biggest limitation in this dissertation is the lack of analyzing real data generated in a real HetVNET. Although SUMO and Veins LTE are popular simulators in modeling of the vehicular network, simulation based data is not as convincing as the real data. However, there was not any alternative way except generating data by simulators. Since the data is extracted from running simulation scenarios, the amount of generated data is not in the scale of big data. Therefore, the choice of prediction algorithms is limited among the machine learning (not deep learning) prediction methods.

Moreover, to have a dataset containing real data, we need a HetVNET implemented in an urban area. Then the data gathered during a time like several days or months. In this condition, we could even consider more features such as the effect of rush hour, city events and especial weather conditions on the network performance in HetVNET. To have such a real test bed, we need to equip a large number of vehicles to be able to communicate with DSRC technology. The cellular and DSRC network providers should cooperate in order to create a real HetVNET on a big scale.

Regarding simulating the HetVNET, the number of vehicular network simulators that work smoothly and are compatible with SUMO are very limited. The Veins LTE was not user friendly and more specifically it is hard to find a good tutorial resource about using Veins LTE. Nevertheless, it was the best choice at the time. Because other powerful simulators like

Network Simulator (NS), did not support heterogeneity of the vehicular networks. Indeed, in HetVNET it is required that all vehicles could communicate by both DSRC and cellular network without any preconditions. In NS we should specify that vehicles communicate with cellular networks during a specific time and after that they could use DSRC, or a group of vehicles uses DSRC and other vehicles apply cellular networks. However, none of these are not a HetVNET. In HetVNET, vehicles must be free to use the DSRC or cellular network.

In Chapter 3 location of the vehicle should be extracted by a GPS device installed in the vehicles. Moreover, for implementing the proposed approach, the fog computing units should have the security authority to access the location of the vehicles.

In Chapter 6, the network infrastructures and resources in the slices belong to the network operators. However, the virtual network operators provide the network services of one or more slices, using the infrastructures. Indeed, the virtual network operators should have enough authority from real network operators in this issue. Therefore, network slicing is considered by relying on cooperation between virtual network operators and real network operators.

8.3 Future Work

Regarding this dissertation, several future works are listed as follows:

- The future work can be predicting the occurrence of congestion in DSRC or 6G of future heterogeneous vehicular networks based on supervised machine learning methods.

- Since simulated data is used in this dissertation, applying the proposed methods using data that come from a real world HetVNET can be considered as a future work. In this regard, big data extracted from a real HetVNET can be used as input of a deep learning algorithm.

- Proposing the network congestion prediction methods which could be implemented using cloud computing and storage services (e.g., Amazon Web Services (AWS), Microsoft Azure and Google Cloud Platform (GCP)) to analyze the big data and predicting the network behaviour in real-time.

- Considering other factors such as mobility model, modulation technique, complexity of scenarios (urban, rural and straight highway), number of eNBs, number of resource blocks and etc. as predictors and generating congestion prediction model for HetVNET.

- The time series data can be the data generated in a network during various time slices. Depending on the network situation, the amount of generated data could be various in time slices. For example, the data generated during the time that an accident occurs is less than data generated in the time that there is not any road hazard (normal situations). Considering time series data and applying LSTM to generate the network congestion prediction model.

- Regarding the approach proposed in Chapter 6 of this dissertation, proposing a network slice scheduling mechanism to calculate the lifetime of the created slices using the CGAN.

- Implementing the proposed mechanisms in the real world test bed, since the simulated environment can not be exactly the same to the real world situations.

- Considering the data generated by infotainment applications and proposing the prediction model that can analyze the huge data and predict network congestion quickly and accurately.

REFERENCES

[1] A. Paranjothi, M. S. Khan, and S. Zeadally, “A survey on congestion detection and control in connected vehicles,” *Ad Hoc Networks*, vol. 108, p. 102277, Nov. 2020.

[2] K. Zheng, Q. Zheng, P. Chatzimisios, W. Xiang, and Y. Zhou, “Heterogeneous vehicular networking: a survey on architecture, challenges, and solutions,” *IEEE communications surveys tutorials*, vol. 17, no. 4, pp. 2377–2396, June 2015.

[3] F. Hu, Q. Hao, and K. Bao, “A survey on software-defined network and openflow: From concept to implementation,” *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 2181–2206, May 2014.

[4] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, “Software-defined networking: A comprehensive survey,” *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, Dec. 2014.

[5] J. Liu, Y. Li, M. Chen, W. Dong, and D. Jin, “Software-defined internet of things for smart urban sensing,” *IEEE communications magazine*, vol. 53, no. 9, pp. 55–63, Sept. 2015.

[6] M. Chiang and T. Zhang, “Fog and iot: An overview of research opportunities,” *IEEE Internet of things journal*, vol. 3, no. 6, pp. 854–864, June 2016.

[7] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge computing: Vision and challenges,” *IEEE internet of things journal*, vol. 3, no. 5, pp. 637–646, June 2016.

[8] X. Masip-Bruin, E. Marín-Tordera, G. Tashakor, A. Jukan, and G.-J. Ren, “Foggy clouds and cloudy fogs: a real need for coordinated management of fog-to-cloud computing systems,” *IEEE Wireless Communications*, vol. 23, no. 5, pp. 120–128, Nov. 2016.

[9] H. F. Atlam, R. J. Walters, and G. B. Wills, “Fog computing and the internet of things: A review,” *big data and cognitive computing*, vol. 2, no. 2, p. 10, June 2018.

[10] N. Alliance, “Description of network slicing concept,” *NGMN 5G P*, vol. 1, no. 1, Jan. 2016.

[11] M. Taruk, E. Budiman, H. J. Setyadi, *et al.*, "Comparison of tcp variants in long term evolution (lte)," in *2017 5th International Conference on Electrical, Electronics and Information Engineering (ICEEIE)*, Malang, Indonesia, 6 2017.

[12] R. Sanchez-Iborra and M.-D. Cano, "On the similarities between urban traffic management and communication networks: Application of the random early detection algorithm for self-regulating intersections," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 4, pp. 48–61, Oct. 2017.

[13] W.-c. Feng, K. G. Shin, D. D. Kandlur, and D. Saha, "The blue active queue management algorithms," *IEEE/ACM transactions on networking*, vol. 10, no. 4, pp. 513–528, Nov. 2002.

[14] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on networking*, vol. 1, no. 4, pp. 397–413, Aug. 1993.

[15] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "Bbr: congestion-based congestion control," *Communications of the ACM*, vol. 60, no. 2, pp. 58–66, Jan. 2017.

[16] E. Kohler, M. Handley, S. Floyd, and J. Padhye, "Datagram congestion control protocol (dccp)," 2006.

[17] M. Handley, S. Floyd, J. Padhye, and J. Widmer, "Tcp friendly rate control (tfrc): Protocol specification," RFC, Tech. Rep., 2003.

[18] S. R. Kolte and M. S. Madnkar, "A design approach of congestion control for safety critical message transmission in vanet," in *2014 Fourth International Conference on Communication Systems and Network Technologies*. Malang, Indonesia: IEEE, 6-8 Oct 2014, pp. 298–301. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6821405

[19] S. Djahel and Y. Ghamri-Doudane, "A robust congestion control scheme for fast and reliable dissemination of safety messages in vanets," in *2012 IEEE Wireless Communications and Networking Conference (WCNC)*. Paris, France: IEEE, 1-4 April 2012, pp. 2264–2269. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6214170

[20] S. Zemouri, S. Djahel, and J. Murphy, "A short-term vehicular density prediction scheme for enhanced beaconing control," in *2015 IEEE Global Communications*

Conference (GLOBECOM). San Diego, CA, USA: IEEE, 6-10 Dec 2015, pp. 1–7. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7417684

[21] O. Chakroun and S. Cherkaoui, "Overhead-free congestion control and data dissemination for 802.11 p vanets," *Vehicular Communications*, vol. 1, no. 3, pp. 123–133, July 2014.

[22] N. Cheng, N. Lu, N. Zhang, X. Zhang, X. S. Shen, and J. W. Mark, "Opportunistic wifi offloading in vehicular environment: A game-theory approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 7, pp. 1944–1955, Jan. 2016.

[23] N. Taherkhani and S. Pierre, "Centralized and localized data congestion control strategy for vehicular ad hoc networks using a machine learning clustering algorithm," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 11, pp. 3275–3285, Apr. 2016.

[24] ——, "Prioritizing and scheduling messages for congestion control in vehicular ad hoc networks," *Computer Networks*, vol. 108, pp. 15–28, Oct 2016.

[25] M. Díaz, C. Martín, and B. Rubio, "State-of-the-art, challenges, and open issues in the integration of internet of things and cloud computing," *Journal of Network and Computer applications*, vol. 67, pp. 99–117, May 2016.

[26] C. Alippi, R. Fantacci, D. Marabissi, and M. Roveri, "A cloud to the ground: The new frontier of intelligent and autonomous networks of things," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 14–20, Dec. 2016.

[27] B. Hassanabadi and S. Valaee, "Reliable periodic safety message broadcasting in vanets using network coding," *IEEE transactions on wireless communications*, vol. 13, no. 3, pp. 1284–1297, Jan. 2014.

[28] K. Xu, K.-C. Wang, R. Amin, J. Martin, and R. Izard, "A fast cloud-based network selection scheme using coalition formation games in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 11, pp. 5327–5339, Dec. 2014.

[29] R. L. Gomes, L. F. Bittencourt, E. R. Madeira, E. C. Cerqueira, and M. Gerla, "Software-defined management of edge as a service networks," *IEEE Transactions on Network and Service Management*, vol. 13, no. 2, pp. 226–239, Mar. 2016.

[30] A. A. Gómez and C. F. Mecklenbräuker, "Dependability of decentralized congestion control for varying vanet density," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 11, pp. 9153–9167, Jan. 2016.

[31] R. Sathya and A. Abraham, "Comparison of supervised and unsupervised learning algorithms for pattern classification," *International Journal of Advanced Research in Artificial Intelligence*, vol. 2, no. 2, pp. 34–38, Feb. 2013.

[32] N. Lyamin, A. Vinel, D. Smely, and B. Bellalta, "Etsi dcc: Decentralized congestion control in c-its," *IEEE Communications Magazine*, vol. 56, no. 12, pp. 112–118, Nov. 2018.

[33] S. Zemouri, S. Djahel, and J. Murphy, "An altruistic prediction-based congestion control for strict beaconing requirements in urban vanets," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 12, pp. 2582–2597, Jan. 2018.

[34] F. Zhang, Y. Du, W. Liu, and P. Li, "Model predictive power control for cooperative vehicle safety systems," *IEEE Access*, vol. 6, pp. 4797–4810, Jan. 2018.

[35] M. Joseph, X. Liu, and A. Jaekel, "An adaptive power level control algorithm for dsrc congestion control," in *Proceedings of the 8th ACM Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications*, Montreal, QC, Canada, Oct 2018, pp. 57–62. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/3272036.3272041

[36] S. A. A. Shah, E. Ahmed, J. J. Rodrigues, I. Ali, and R. M. Noor, "Shapely value perspective on adapting transmit power for periodic vehicular communications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 977–986, Jan. 2018.

[37] S. Sharma, M. Chahal, and S. Harit, "Transmission rate-based congestion control in vehicular ad hoc networks," in *2019 Amity International Conference on Artificial Intelligence (AICAI)*. Dubai, United Arab Emirates: IEEE, 4-6 Feb 2019, pp. 303–307. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8701304

[38] B.-M. Cho, M.-S. Jang, and K.-J. Park, "Channel-aware congestion control in vehicular cyber-physical systems," *IEEE Access*, vol. 8, pp. 73 193–73 203, Apr. 2020.

[39] M. Allman, V. Paxson, W. Stevens, *et al.*, "Tcp congestion control," 1999.

[40] E. Cinque, F. Valentini, A. Iovine, and M. Pratesi, "An adaptive strategy to mitigate instability in the etsi dcc: Experimental validation," in *2017 15th International Conference on ITS Telecommunications (ITST)*. Warsaw, Poland: IEEE, 29-31 May 2017, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7972223

[41] C. Navdeti, C. Giri, and I. Banerjee, "Distributed α-fair transmit power adaptation based congestion control in vanet," in *Proceedings of the 20th International Conference on Distributed Computing and Networking*, Bangalore, India, Jan 2019, pp. 253–260. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/3288599.3288606

[42] A. Rostami, B. Cheng, G. Bansal, K. Sjöberg, M. Gruteser, and J. B. Kenney, "Stability challenges and enhancements for vehicular channel congestion control approaches," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 10, pp. 2935–2948, Apr. 2016.

[43] N. Lyamin, B. Bellalta, and A. Vinel, "Age-of-information-aware decentralized congestion control in vanets," *IEEE Networking Letters*, vol. 2, no. 1, pp. 33–37, Feb. 2020.

[44] G. Bansal, J. B. Kenney, and C. E. Rohrs, "Limeric: A linear adaptive message rate algorithm for dsrc congestion control," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 9, pp. 4182–4197, Jul 2013.

[45] T. Tielert, D. Jiang, Q. Chen, L. Delgrossi, and H. Hartenstein, "Design methodology and evaluation of rate adaptation based congestion control for vehicle safety communications," in *2011 IEEE vehicular networking conference (VNC)*, 14-16 Nov. 2011, pp. 116–123.

[46] B. Aygun, M. Boban, and A. M. Wyglinski, "Ecpr: Environment-and context-aware combined power and rate distributed congestion control for vehicular communications," *Computer Communications*, vol. 93, pp. 3–16, Nov 2016.

[47] E. Egea-Lopez, J. J. Alcaraz, J. Vales-Alonso, A. Festag, and J. Garcia-Haro, "Statistical beaconing congestion control for vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 9, pp. 4162–4181, May 2013.

[48] M. I. Khan, J. Härri, and S. Sesia, "Enhancing etsi dcc for multi-service vehicular safety communication," in *2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*. Victoria, BC, Canada: IEEE, 18 Nov.-16 Dec 2020, pp. 1–5.

[49] C. B. Math, H. Li, S. H. de Groot, and I. Niemegeers, "A combined fair decentralized message-rate and data-rate congestion control for v2v communication," in *2017 IEEE Vehicular Networking Conference (VNC)*. Turin, Italy: IEEE, 27-29 Nov 2017, pp. 271–278.

[50] L. Delgrossi and T. Zhang, *Vehicle safety communications: protocols, security, and privacy.* Hoboken, New Jersey, US: John Wiley Sons, Inc., 2012. [Online]. Available: http://onlinelibrary.wiley.com/book/10.1002/9781118452189

[51] R. Ramanathan *et al.*, "An empirical study on mac layer in ieee 802.11 p/wave based vehicular ad hoc networks," *Procedia computer science*, vol. 143, pp. 720–727, Jun 2018.

[52] D. B. Rawat, D. C. Popescu, G. Yan, and S. Olariu, "Enhancing vanet performance by joint adaptation of transmission power and contention window size," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 9, pp. 1528–1535, Jan 2011.

[53] Y. Lu, Z. Ling, S. Zhu, and L. Tang, "Sdtcp: Towards datacenter tcp congestion control with sdn for iot applications," *Sensors*, vol. 17, no. 1, p. 109, Jan. 2017.

[54] D. Wu, D. Zhang, and L. Sun, "An aggregate parameter for congestion detection in vanets," in *2012 Fifth International Conference on Information and Computing Science*. Liverpool, UK: IEEE, 24-25 July 2012, pp. 95–98. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6258080

[55] Y. Zang, L. Stibor, X. Cheng, H.-J. Reumerman, A. Paruzel, and A. Barroso, "Congestion control in wireless networks for vehicular safety applications," in *Proceedings of the 8th European Wireless Conference*, vol. 7, Liverpool, UK, Apr 2007, p. 1. [Online]. Available: http://confs.comelec.enst.fr/EW2007/papers/1569014654.pdf

[56] P. Kumar, H. S. Kataria, and T. Ghosh, "Congestion control approach by reducing the number of messages in vanet," in *2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO)(Trends and Future Directions)*. Noida, India: IEEE, 2-4 Sept 2015, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7359297

[57] L. Zhang and S. Valaee, "Congestion control for vehicular networks with safety-awareness," *IEEE/ACM Transactions on Networking*, vol. 24, no. 6, pp. 3290–3299, Feb. 2016.

[58] E. Egea-Lopez and P. Pavon-Mariño, "Fair congestion control in vehicular networks with beaconing rate adaptation at multiple transmit powers," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 3888–3903, Mar. 2016.

[59] C. Choe, J. Choi, J. Ahn, D. Park, and S. Ahn, "Multiple channel access using deep reinforcement learning for congested vehicular networks," in *2020 IEEE 91st Vehicular*

Technology Conference (VTC2020-Spring). Antwerp, Belgium: IEEE, 25-28 May 2020, pp. 1–6.

[60] C.-X. Song, J.-Z. Li, and N. Ding, "Joint power and contention window adaptive control for reliable and efficient broadcast of safety-related services in vanets," *Journal of Computers*, vol. 30, no. 6, pp. 84–99, Dec. 2019.

[61] S. Bai, J. Oh, and J.-i. Jung, "Context awareness beacon scheduling scheme for congestion control in vehicle to vehicle safety communication," *Ad Hoc Networks*, vol. 11, no. 7, pp. 2049–2058, Sep. 2013.

[62] N. Taherkhani and S. Pierre, "Congestion control in vehicular ad hoc networks using meta-heuristic techniques," in *Proceedings of the second ACM international symposium on Design and analysis of intelligent vehicular networks and applications*. New York, NY, USA: Association for Computing Machinery, Oct. 2012, pp. 47–54.

[63] ——, "Improving dynamic and distributed congestion control in vehicular ad hoc networks," *Ad Hoc Networks*, vol. 33, pp. 112–125, Oct. 2015.

[64] P. Si, Y. He, H. Yao, R. Yang, and Y. Zhang, "Dave: Offloading delay-tolerant data traffic to connected vehicle networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 3941–3953, Apr. 2016.

[65] C. Campolo, A. Molinaro, A. Iera, and F. Menichella, "5g network slicing for vehicle-to-everything services," *IEEE Wireless Communications*, vol. 24, no. 6, pp. 38–45, Dec. 2017.

[66] K. Samdanis, X. Costa-Perez, and V. Sciancalepore, "From network sharing to multi-tenancy: The 5g network slice broker," *IEEE Communications Magazine*, vol. 54, no. 7, pp. 32–39, July 2016.

[67] X. Zhou, R. Li, T. Chen, and H. Zhang, "Network slicing as a service: enabling enterprises' own software-defined cellular networks," *IEEE Communications Magazine*, vol. 54, no. 7, pp. 146–153, July 2016.

[68] R. Su, D. Zhang, R. Venkatesan, Z. Gong, C. Li, F. Ding, F. Jiang, and Z. Zhu, "Resource allocation for network slicing in 5g telecommunication networks: A survey of principles and models," *IEEE Network*, vol. 33, no. 6, pp. 172–179, July 2019.

[69] X. Li, B.-J. Hu, H. Chen, G. Andrieux, Y. Wang, and Z.-H. Wei, "An rsu-coordinated synchronous multi-channel mac scheme for vehicular ad hoc networks," *IEEE Access*, vol. 3, pp. 2794–2802, Dec. 2015.

[70] S. A. A. Shah, E. Ahmed, M. Imran, and S. Zeadally, "5g for vehicular communications," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 111–117, Jan. 2018.

[71] S. E. Elayoubi, S. B. Jemaa, Z. Altman, and A. Galindo-Serrano, "5g ran slicing for verticals: Enablers and challenges," *IEEE Communications Magazine*, vol. 57, no. 1, pp. 28–34, Jan. 2019.

[72] A. Balador, C. Bai, and F. Sedighi, "A comparison of decentralized congestion control algorithms for multiplatooning communications," in *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. Kyoto, Japan: IEEE, 11-15 March 2019, pp. 674–680.

[73] C. Chen, J. Hu, J. Sui, and Y. Zhou, "An information congestion control scheme in the internet of vehicles: A bargaining game approach," *Computers & Electrical Engineering*, vol. 58, pp. 282–298, 2017.

[74] S. Khatri, H. Vachhani, S. Shah, J. Bhatia, M. Chaturvedi, S. Tanwar, and N. Kumar, "Machine learning models and techniques for vanet based traffic management: Implementation issues and challenges," *Peer-to-Peer Networking and Applications*, vol. 14, no. 3, pp. 1778–1805, May 2021.

[75] C. Jiang, H. Zhang, Y. Ren, Z. Han, K.-C. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Communications*, vol. 24, no. 2, pp. 98–105, Dec. 2016.

[76] A. Yousefpour, C. Fung, T. Nguyen, K. Kadiyala, F. Jalali, A. Niakanlahiji, J. Kong, and J. P. Jue, "All one needs to know about fog computing and related edge computing paradigms: A complete survey," *Journal of Systems Architecture*, vol. 98, pp. 289–330, Sep. 2019.

[77] C. Chen, T. Qiu, J. Hu, Z. Ren, Y. Zhou, and A. K. Sangaiah, "A congestion avoidance game for information exchange on intersections in heterogeneous vehicular networks," *Journal of Network and Computer Applications*, vol. 85, pp. 116–126, May 2017.

[78] K. N. Qureshi, A. H. Abdullah, O. Kaiwartya, S. Iqbal, R. A. Butt, and F. Bashir, "A dynamic congestion control scheme for safety applications in vehicular ad hoc networks," *Computers & Electrical Engineering*, vol. 72, pp. 774–788, Nov. 2018.

[79] R. Regin and T. Menakadevi, "Dynamic clustering mechanism to avoid congestion control in vehicular ad hoc networks based on node density," *Wireless Personal Communications*, vol. 107, no. 4, pp. 1911–1931, Aug. 2019.

[80] F. Zhang, G. Tan, C. Yu, N. Ding, C. Song, and M. Liu, "Fair transmission rate adjustment in cooperative vehicle safety systems based on multi-agent model predictive control," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 7, pp. 6115–6129, Dec. 2016.

[81] K. Zheng, Q. Zheng, P. Chatzimisios, W. Xiang, and Y. Zhou, "Heterogeneous vehicular networking: A survey on architecture, challenges, and solutions," *IEEE communications surveys & tutorials*, vol. 17, no. 4, pp. 2377–2396, Jan. 2015.

[82] C. Huang, M. Chiang, D. Dao, H. Pai, S. Xu, and H. Zhou, "Vehicle-to-infrastructure (v2i) offloading from cellular network to 802.11 p wi-fi network based on the software-defined network (sdn) architecture," *Vehicular Communications*, vol. 9, pp. 288–300, Jul. 2017.

[83] F. Falahatraftar, S. Pierre, and S. Chamberland, "A multiple linear regression model for predicting congestion in heterogeneous vehicular networks," in *2020 16th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)(50308)*. Thessaloniki, Greece: IEEE, 12-14 Oct 2020, pp. 93–98.

[84] Z. Tang, Y. Zhao, L. Yang, S. Qi, D. Fang, X. Chen, X. Gong, and Z. Wang, "Exploiting wireless received signal strength indicators to detect evil-twin attacks in smart homes," *Mobile Information Systems*, vol. 2017, Jan. 2017.

[85] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2017.

[86] I. Soto, O. Amador, M. Urueña, and M. Calderon, "Strengths and weaknesses of the etsi adaptive dcc algorithm: A proposal for improvement," *IEEE Communications Letters*, vol. 23, no. 5, pp. 802–805, Mar. 2019.

[87] C. Sommer, S. Joerer, M. Segata, O. K. Tonguz, R. L. Cigno, and F. Dressler, "How shadowing hurts vehicular communications and how dynamic beaconing can help," *IEEE Transactions on Mobile Computing*, vol. 14, no. 7, pp. 1411–1421, Oct. 2014.

[88] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using sumo," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. Maui, HI, USA: IEEE, 4-7 Nov 2018, pp. 2575–2582. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8569938

[89] F. Hagenauer, F. Dressler, and C. Sommer, "Poster: A simulator for heterogeneous vehicular networks," in *2014 IEEE Vehicular Networking Conference (VNC)*, Paderborn, Germany, 3-5 Dec 2014.

[90] O. contributors. (2017) Planet dump retrieved from https://planet.osm.org. [Online]. Available: https://www.openstreetmap.org.

[91] K. T. Herring, J. W. Holloway, D. H. Staelin, and D. W. Bliss, "Path-loss characteristics of urban wireless channels," *IEEE Transactions on Antennas and Propagation*, vol. 58, no. 1, pp. 171–177, Nov. 2009.

[92] P. Ignaciuk and A. Bartoszewicz, *Congestion control in data transmission networks: sliding mode and other designs*. London, England, United Kingdom: Springer-Verlag, 2012.

[93] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *Proceedings of the 23rd international conference on Machine learning*, New York, NY, United States, Jun 2006.

[94] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 2nd ed. Sebastopol, CA, US: O'Reilly Media, 2019.

[95] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*. New York, USA: Springer, 2013, vol. 112.

[96] N. Taherkhani, "Congestion control in vehicular ad hoc networks," thèse de doctorat, Dép. de génie informatique et génie logiciel, École Polytechnique de Montréal, Montréal, QC, 2015. [Online]. Available: https://publications.polymtl.ca/1971/.

[97] Q. D. La, M. V. Ngo, T. Q. Dinh, T. Q. Quek, and H. Shin, "Enabling intelligence in fog computing to achieve energy and latency reduction," *Digital Communications and Networks*, vol. 5, no. 1, pp. 3–9, Feb. 2019.

[98] K. Zheng, X. Wang, Y. Li, and P. Chatzimisios, "Ieee access special section editorial: Communication, control, and computation issues in heterogeneous vehicular networks," *IEEE Access*, vol. 6, pp. 79 285–79 287, Dec. 2018.

[99] S. Weisberg, *Applied linear regression*, vol. 528.

[100] C. J. Willmott, "Some comments on the evaluation of model performance," *Bulletin of the American Meteorological Society*, vol. 63, no. 11, pp. 1309–1313, Nov. 1982.

[101] O. Amador, I. Soto, M. Calderón, and M. Urueña, "Experimental evaluation of the etsi dcc adaptive approach and related algorithms," *IEEE Access*, vol. 8, pp. 49 798–49 811, Mar. 2020.

[102] F. Goudarzi and H. Asgari, "Non-cooperative beacon power control for vanets," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 2, pp. 777–782, Apr. 2018.

[103] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "Bbr: Congestion-based congestion control," *Queue*, vol. 14, no. 5, pp. 20–53, Jan. 2016.

[104] M. Torrent-Moreno, P. Santi, and H. Hartenstein, "Distributed fair transmit power adjustment for vehicular ad hoc networks," in *2006 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks*, vol. 2. Reston, VA, USA: IEEE, 28-28 Sept. 2006, pp. 479–488.

[105] S. Kuhlmorgen, A. Festag, and G. Fettweis, "Evaluation of multi-hop packet prioritization for decentralized congestion control in vanets," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*. San Francisco, CA, USA: IEEE, 19-22 Mar. 2017, pp. 1–6.

[106] A. Autolitano, C. Campolo, A. Molinaro, R. M. Scopigno, and A. Vesco, "An insight into decentralized congestion control techniques for vanets from etsi ts 102 687 v1. 1.1," in *2013 IFIP Wireless Days (WD)*. Valencia, Spain: IEEE, 13-15 Nov. 2013, pp. 1–6.

[107] N. Haouari, S. Moussaoui, and S.-M. Senouci, "Application reliability analysis of density-aware congestion control in vanets," in *2018 IEEE International Conference on Communications (ICC)*. Kansas City, MO, USA: IEEE, 20-24 May 2018, pp. 1–6.

[108] D. F. Specht, "A general regression neural network," *IEEE transactions on neural networks*, vol. 2, no. 6, pp. 568–576, Nov. 1991.

[109] E. Parzen, "On estimation of a probability density function and mode," *The annals of mathematical statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.

[110] Q. Duan, N. Ansari, and M. Toy, "Software-defined network virtualization: An architectural framework for integrating sdn and nfv for service provisioning in future networks," *IEEE Network*, vol. 30, no. 5, pp. 10–16, 2016.

[111] J. G. Herrera and J. F. Botero, "Resource allocation in nfv: A comprehensive survey," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 518–532, Aug. 2016.

[112] H. Zhang, N. Liu, X. Chu, K. Long, A.-H. Aghvami, and V. C. Leung, “Network slicing based 5g and future mobile networks: mobility, resource management, and challenges,” *IEEE communications magazine*, vol. 55, no. 8, pp. 138–145, Aug. 2017.

[113] P. Rost, C. Mannweiler, D. S. Michalopoulos, C. Sartori, V. Sciancalepore, N. Sastry, O. Holland, S. Tayade, B. Han, D. Bega, *et al.*, “Network slicing to enable scalability and flexibility in 5g mobile networks,” *IEEE Communications magazine*, vol. 55, no. 5, pp. 72–79, May 2017.

[114] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, “Network slicing in 5g: Survey and challenges,” *IEEE Communications Magazine*, vol. 55, no. 5, pp. 94–100, May 2017.

[115] X. Shen, J. Gao, W. Wu, K. Lyu, M. Li, W. Zhuang, X. Li, and J. Rao, “Ai-assisted network-slicing based next-generation wireless networks,” *IEEE Open Journal of Vehicular Technology*, vol. 1, pp. 45–66, Jan. 2020.

[116] S. K. Tayyaba, H. A. Khattak, A. Almogren, M. A. Shah, I. U. Din, I. Alkhalifa, and M. Guizani, “5g vehicular network resource management for improving radio access through machine learning,” *IEEE Access*, vol. 8, pp. 6792–6800, Jan. 2020.

[117] C. Campolo, A. Molinaro, A. Iera, R. R. Fontes, and C. E. Rothenberg, “Towards 5g network slicing for the v2x ecosystem,” in *2018 4th IEEE conference on network softwarization and workshops (NetSoft)*, Montreal, QC, Canada, 2018.

[118] Y. Cui, H. Zheng, H. Wang, and D. Wu, “An intelligent coordinator design for network slicing in service-oriented vehicular networks,” in *GLOBECOM 2020-2020 IEEE Global Communications Conference*, Taipei, Taiwan, 7-11 Dec 2020.

[119] J. Mei, X. Wang, and K. Zheng, “Intelligent network slicing for v2x services toward 5g,” *IEEE Network*, vol. 33, no. 6, pp. 196–204, Oct. 2019.

[120] Y. Chen, Y. Wang, M. Liu, J. Zhang, and L. Jiao, “Network slicing enabled resource management for service-oriented ultra-reliable and low-latency vehicular networks,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 7, pp. 7847–7862, May 2020.

[121] H. Khan, P. Luoto, M. Bennis, and M. Latva-aho, “On the application of network slicing for 5g-v2x,” in *European Wireless 2018; 24th European Wireless Conference*, Catania, Italy, 2-4 May 2018.

[122] R. Sanchez-Iborra, J. Santa, J. Gallego-Madrid, S. Covaci, and A. Skarmeta, "Empowering the internet of vehicles with multi-rat 5g network slicing," *Sensors*, vol. 19, no. 14, p. 3107, Jan. 2019.

[123] H. Wang, Y. Wu, G. Min, J. Xu, and P. Tang, "Data-driven dynamic resource scheduling for network slicing: A deep reinforcement learning approach," *Information Sciences*, vol. 498, pp. 106–116, Sept. 2019.

[124] J. Koo, V. B. Mendiratta, M. R. Rahman, and A. Walid, "Deep reinforcement learning for network slicing with heterogeneous resource requirements and time varying traffic dynamics," in *2019 15th International Conference on Network and Service Management (CNSM)*. Halifax, NS, Canada: IEEE, 21-25 Oct 2019, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9012702

[125] Y. Cui, X. Huang, D. Wu, and H. Zheng, "Machine learning based resource allocation strategy for network slicing in vehicular networks," in *2020 IEEE/CIC International Conference on Communications in China (ICCC)*. Chongqing, China: IEEE, 9-11 Aug 2020, pp. 454–459. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9238991

[126] K. Xiong, S. Leng, J. Hu, X. Chen, and K. Yang, "Smart network slicing for vehicular fog-rans," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3075–3085, Feb. 2019.

[127] H. Khan, P. Luoto, S. Samarakoon, M. Bennis, and M. Latva-Aho, "Network slicing for vehicular communication," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 1, p. e3652, Jan. 2021.

[128] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, Nov. 2014. [Online]. Available: https://arxiv.org/abs/1411.1784

APPENDIX A CLASSIFICATION PERFORMANCE METRICS

$$Recall = \frac{TP}{TP + FN} \tag{A.1}$$

$$Precision = \frac{TP}{TP + FP} \tag{A.2}$$

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{A.3}$$

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{A.4}$$

APPENDIX B REGRESSION PERFORMANCE METRICS

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{B.1}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{B.2}$$

$$RSS = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{B.3}$$

$$RSE = \sqrt{\frac{1}{(n-p-1)}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{B.4}$$

$$TSS = \sum_{i=1}^{n}(y_i - \overline{y}_i)^2 \tag{B.5}$$

$$R^2 = 1 - \frac{RSS}{TSS} \tag{B.6}$$

$$F - statistic = \frac{(TSS - RSS) \times \frac{1}{p}}{RSS \times \frac{1}{n-p-1}} \tag{B.7}$$

$$SE(\hat{\beta}_0)^2 = \sigma^2[\frac{1}{n} + \frac{\overline{x}^2}{\sum_{i=1}^{n}(x_i - \overline{x})^2}] \tag{B.8}$$

$$SE(\hat{\beta}_1)^2 = \frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \overline{x})^2} \tag{B.9}$$

$$t - statistic = \frac{\hat{\beta}_1}{SE(\hat{\beta}_1)} \tag{B.10}$$

APPENDIX C CROSS-ENTROPY

$$H_i = -\sum_{k=1}^{n} P_{i,k} \log(P_{i,k}) \tag{C.1}$$

$$P_{i,k} \neq 0$$