IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 13, 2021, accepted August 24, 2021, date of publication August 27, 2021, date of current version September 10, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3108425

A Centralized and Dynamic Network Congestion Classification Approach for Heterogeneous Vehicular Networks

FARNOUSH FALAHATRAFTAR, SAMUEL PIERRE, (Senior Member, IEEE), AND STEVEN CHAMBERLAND, (Member, IEEE)
Mobile Computing and Networking Research Laboratory (LARIM), Department of Computer and Software Engineering, Polytechnique de Montréal, Montreal, QC H3T 1J4, Canada

Corresponding author: Farnoush Falahatraftar (farnoush.falahatraftar@polymtl.ca)

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.

INDEX TERMS Vehicular networks, congestion control, classification methods, network congestion prediction, WAVE.

I. 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 [1]. Minimum human reaction time is 500 ms [2]. 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 solving 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 [2]–[4], and vehicle density [5]–[7]. 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 [2]. The obtained results in the related literature [2], [8], [9] demonstrate that congestion in a dynamic environment, such as a vehicular network with a high number of vehicles, has

The associate editor coordinating the review of this manuscript and approving it for publication was Eyuphan Bulut.

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 [10]–[12] 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 [13], [14]. Fog computing supports mobility, location awareness, and real-time interaction. Well designed and configured it can improve metrics [14], [15]. 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 [16], 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 [14]. 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 [2].

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 [11]. 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 II presents related work. Section III describes the methodology and classification model. Data collection and performance evaluation are presented and discussed in Section IV, and Section V concludes the study and introduces future work.

II. 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 [17]–[20]. The authors in [19] proposed a congestion game to avoid congestion based on scheduling the required services for safety-related applications in HetVNETs. In [2], 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 Inter-frame 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 [21], 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 [18], 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 [19], 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 [19] 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 [20], 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 [22], 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 [23] to propose an approach for controlling congestion using dynamic transmission power control. In [24] 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 [24].

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 [5], 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 [1]. 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 [5]. These challenges exist in all of the proposed decentralized (or distributed) congestion control mechanisms, such as [5] and [25]. In [25], 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 [26], 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 [27], 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.

III. 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 [11], [28]. 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.

A. 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 [29]. 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:

$$\text{Data flowing classes} = \begin{cases} \text{Warning, if:} \\ \quad DDR_{minth} \leq DDR \leq DDR_{maxth} \\ and, \\ \quad RSS_{th} \leq RSS \\ \text{Nonwarning, } otherwise \end{cases} \tag{1}$$

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 [28].

B. 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 [30]. 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:

$$P(c_i|x) = \frac{P(x|c_i)P(c_i)}{P(x)} \tag{2}$$

The Naive Bayesian algorithm calculates as follows:

$$P(w_0|v, dr, tp_{DSRC}, tp_{LTE}, b) = \frac{P(v|w_0)P(dr|w_0)P(tp_{DSRC}|w_0)P(tp_{LTE}|w_0)P(b|w_0)P(w_0)}{P(v, dr, tp_{DSRC}, tp_{LTE}, b)} \tag{3}$$

and

$$P(w_1|v, dr, tp_{DSRC}, tp_{LTE}, b) = \frac{P(v|w_1)P(dr|w_1)P(tp_{DSRC}|w_1)P(tp_{LTE}|w_1)P(b|w_1)P(w_1)}{P(v, dr, tp_{DSRC}, tp_{LTE}, b)} \tag{4}$$

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 (2) can be written as (5):

$$P(c_i|x) \propto P(c_i) \prod_{j=1}^{n} P(x_j|c_i), \tag{5}$$

$i = 1, \ldots, m$.

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:

$$\hat{y} = argmaxP(c_i) \prod_{j=1}^{n} P(x_j|c_i), \tag{6}$$

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

C. 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. 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 (1) by defining the values of DDR_{minth}, DDR_{maxth} and RSS_{th}. Therefore, Fig. 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:

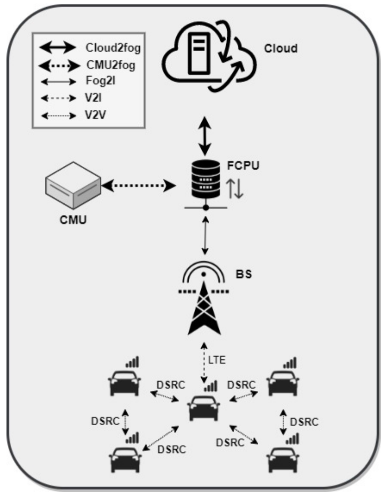


FIGURE 1. A centralized, dynamic and intelligent cloudy-fog congestion prediction architecture of HetVNET.

- 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.

As Fig. 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. 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

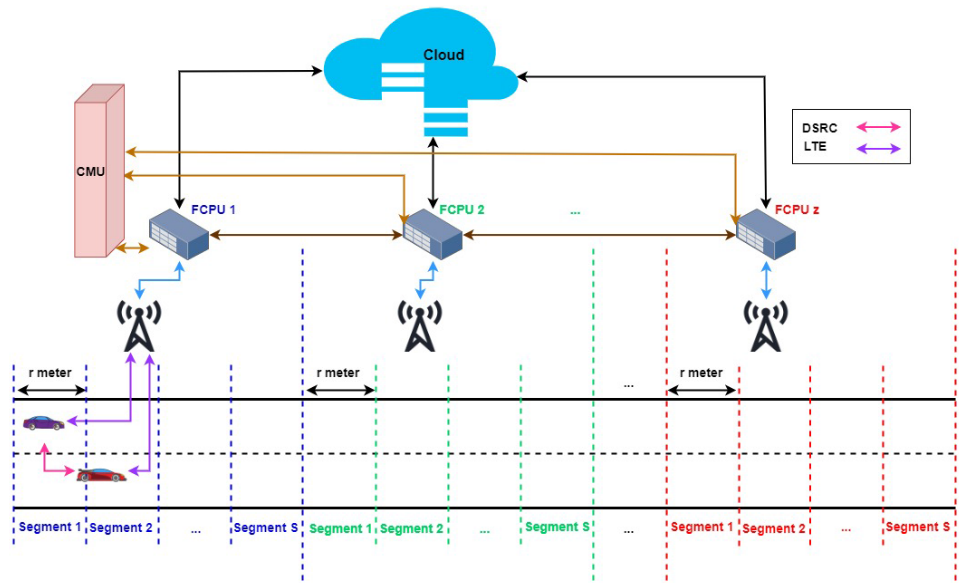


FIGURE 2. Illustration of the intelligent congestion prediction architecture of HetVNET.

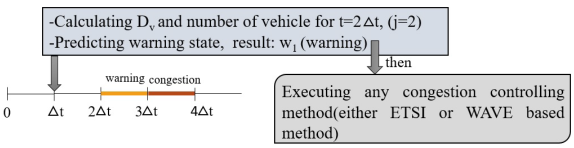


FIGURE 3. Congestion warning state in a HetVNET.

next time unit such as t using the following formula:

$$D_{(v)} = \frac{1}{2}a_{(v)}t^2 + q_{(v)}t, \tag{7}$$

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 $j = \{1, 2, 3, \ldots, N\}$, 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 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. 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.

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.

To apply the proposed Naive Bayesian classifier in the centralized and dynamic cloudy-fog-based architecture of HetVNET, as shown in Fig. 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

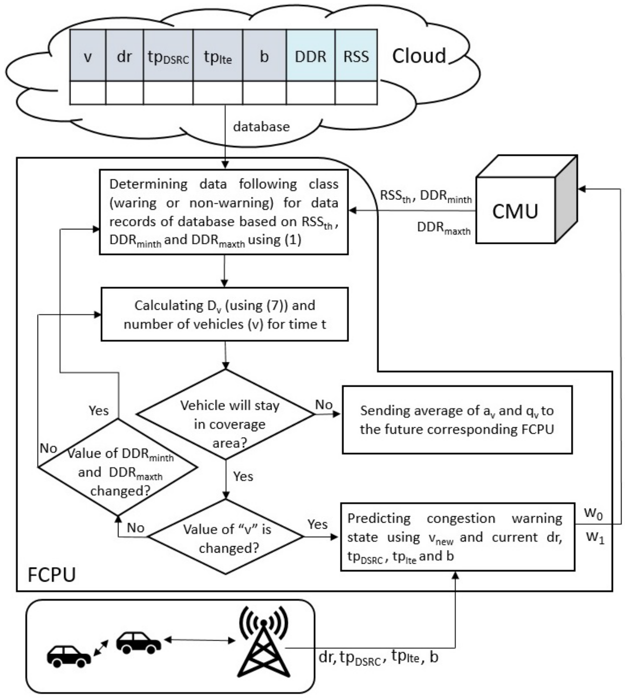


FIGURE 4. A Flowchart of the main steps in an FCPU.

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 (1)).
Information collection locally as x= [v, dr, tp_{DSRC}, tp_{LTE}, b] form HetVNET.
Calculate D_v and v_{new}.
  if $v_{new} \neq v$ **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

---

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 (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) and (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.

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 [1], 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.

D. 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) [31], Dual-α DCC [32], and Dynamic Beaconing (DnyB) protocol [33], 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 [32]. An optimal value for the CBR threshold can prevent the underutilization of the channel [4]. 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 [11]. 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.

IV. DATA COLLECTION AND PERFORMANCE EVALUATION

A. 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 [34] simulator and the Veins LTE version 1.3 [35], both in Linux (Ubuntu 16.04). The boroughs of Montreal city in Canada are considered for simulating vehicular traffic and heterogeneous network environment. ''OpenStreetMap'' [36] 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 [37] in the simulation scenarios. DCC (used in ETSI) is based on changing the value of the data transmission rate and transmission power [4], [31], [38]. Additionally, most of the proposed congestion controls in WAVE standard are based on adapting the transmission power and data transmission rate [28], [39]–[41]. Therefore, in each run, we changed the value of v, dr, tp_{DSRC}, tp_{LTE} and b according to Table 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 [42]. The amounts of generated and transmitted data (during 1000 s of running a simulation scenario) are used to calculate the DDR.

TABLE 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

In [28], 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 [28], 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 (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.

B. PERFORMANCE EVALUATION OF THE CONGESTION CLASSIFICATION METHOD

Table 2 is prepared to clarify the relationship between the actual and predicted classes [43].

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 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

TABLE 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)

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* [44].

Regarding the abovementioned discussion, we evaluated the performance of the proposed Naive Bayesian classifier.

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. 5 and 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 [44].

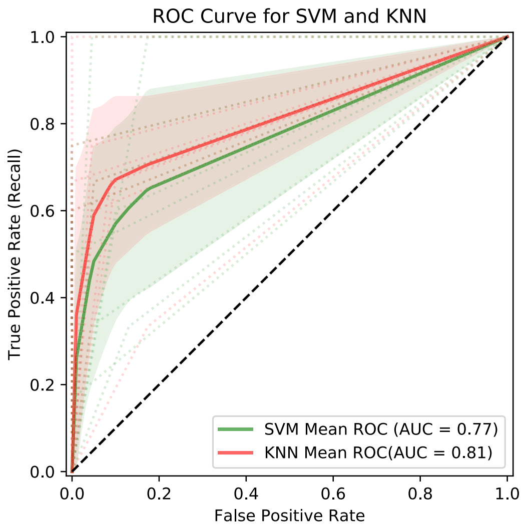


FIGURE 5. ROC curve for SVM and KNN congestion warning classifiers of HetVNET.

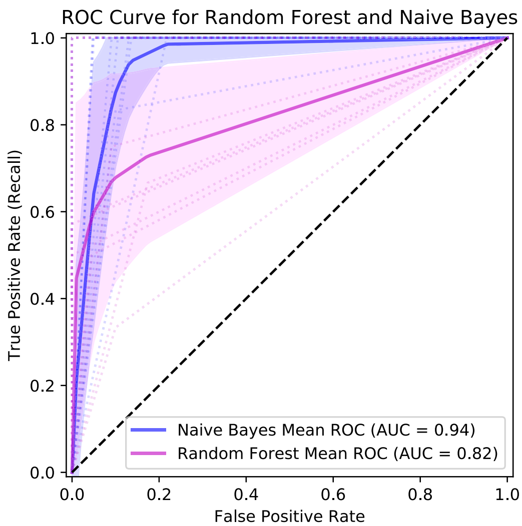


FIGURE 6. ROC curve for Random Forest and Naive Bayesian congestion warning classifiers of HetVNET.

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 [45], 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 5 and 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 [44].

In Fig. 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. 6. However, from the variance area of KNN in Fig. 5 and Random Forest (pink area) in Fig. 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, $1 - specificity$ 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. 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. 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. 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. 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**

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. 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 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. 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. 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. 7, Naive Bayes is a faster learner classifier than SVM, KNN, and Random Forest.

Finally, considering the discussion about the results related to mean ROC curves, which are demonstrated by Figs. 5 and 6, the obtained performance results in Table 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.

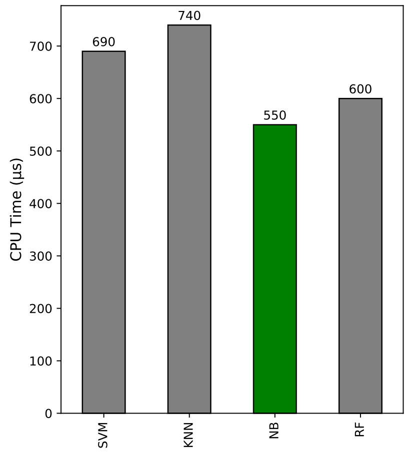


FIGURE 7. CPU time in microseconds for four classifiers.

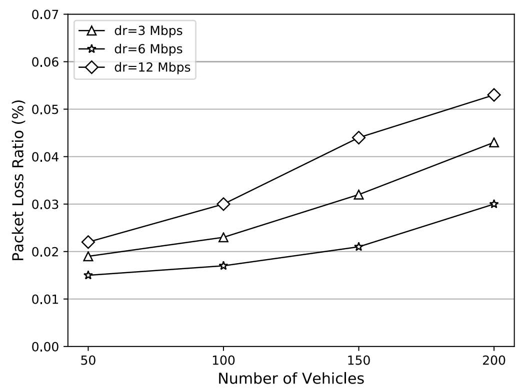


FIGURE 8. Variation in the packet loss ratio using the CNCC mechanism with the numbers of vehicles for various values of the data transmission rate.

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.

C. 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 [19], [45], and the data transmission rate is 3 Mbps. This low data rate is selected to prevent noise and interference [28]. Moreover, based on a study presented in [47], in a moderate

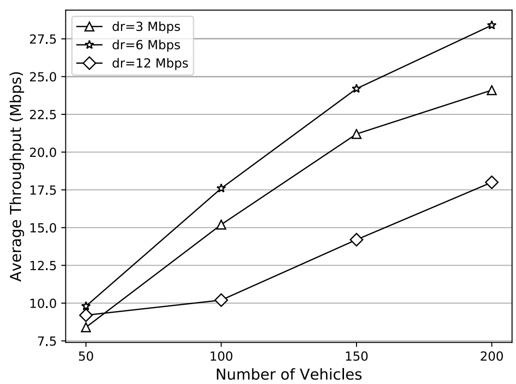


FIGURE 9. Variation of the average throughput using the CNCC mechanism with the numbers of vehicles for various values of the data transmission rate.

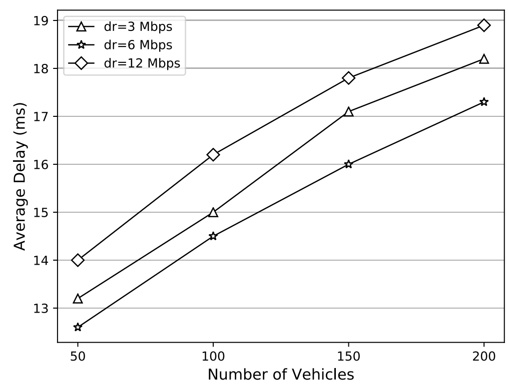


FIGURE 10. Variation in the average delay using the CNCC mechanism with the numbers of vehicles for various values of the data transmission rate.

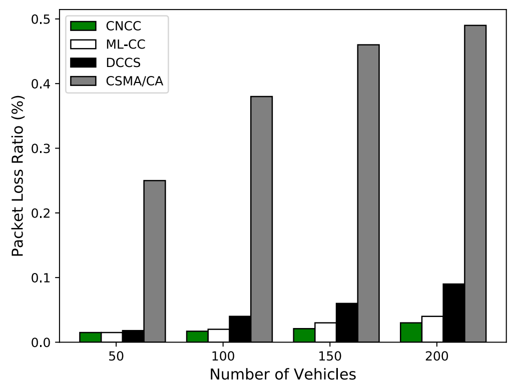


FIGURE 11. Packet loss ratio of the four considered mechanisms with various numbers of vehicles.

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 [19], [46], 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. 8 to 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. 8, 9 and 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. 8, 9 and 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 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 [48]. The FCPU computes $D_{(v)}$ and v_{new} and predicts the congestion state using (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.

In this paper, we compare the performance of the CNCC to contention window-based methods such as CSMA/CA, ML-CC and DCCS. The results of the packet loss ratio are presented in Fig. 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. 11, CNCC could improve the packet loss ratio compared to CSMA/CA, ML-CC and DCCS.

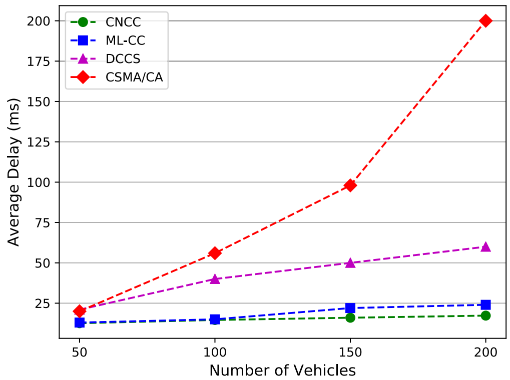


FIGURE 12. Average delay of the four considered mechanisms with various numbers of vehicles.

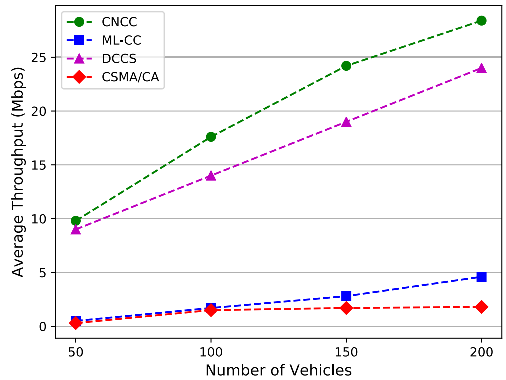


FIGURE 13. Average throughput of the four considered mechanisms with various numbers of vehicles.

Fig. 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. 12, the CNCC could improve the average delay, especially in dense vehicular environments. According to the results shown in Fig. 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. 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. 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.

V. CONCLUSION 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.

REFERENCES

[1] K. Zheng, Q. Zheng, P. Chatzimisios, W. Xiang, and Y. Zhou, "Heterogeneous vehicular networking: A survey on architecture, challenges, and solutions," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2377–2396, 4th Quart., 2015, doi: 10.1109/COMST.2015.2440103.

[2] N. Taherkhani and S. Pierre, "Centralized and localized data congestion control strategy for vehicular ad hoc networks using a machine learning clustering algorithm," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 11, pp. 3275–3285, Nov. 2016.

[3] 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 Trans. Intell. Transp. Syst.*, vol. 17, no. 10, pp. 2935–2948, Oct. 2016.

[4] N. Lyamin, A. Vinel, D. Smely, and B. Bellalta, "ETSI DCC: Decentralized congestion control in C-ITS," *IEEE Commun. Mag.*, vol. 56, no. 12, pp. 112–118, Dec. 2018.

[5] S. Zemouri, S. Djahel, and J. Murphy, "A short-term vehicular density prediction scheme for enhanced beaconing control," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, San Diego, CA, USA, Dec. 2015, pp. 1–7.

[6] B. Hassanabadi and S. Valaee, "Reliable periodic safety message broadcasting in VANETs using network coding," *IEEE Trans. Wireless Commun.*, vol. 13, no. 3, pp. 1284–1297, Mar. 2014.

[7] S. R. Kolte and M. S. Madnkar, "A design approach of congestion control for safety critical message transmission in VANET," in *Proc. 4th Int. Conf. Commun. Syst. Netw. Technol.*, Bhopal, India, Apr. 2014, pp. 298–301.

[8] A. Balador, C. Bai, and F. Sedighi, "A comparison of decentralized congestion control algorithms for multiplatooning communications," in *Proc. IEEE Int. Conf. Pervas. Comput. Commun. Workshops (PerCom Workshops)*, Kyoto, Japan, Mar. 2019, pp. 674–680.

[9] C. Chen, J. Hu, J. Sui, and Y. Zhou, "An information congestion control scheme in the internet of vehicles: A bargaining game approach," *Comput. Electr. Eng.*, vol. 58, pp. 282–298, Feb. 2017.

[10] 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-Peer Netw. Appl.*, vol. 14, no. 3, pp. 1778–1805, Sep. 2020, doi: 10.1007/s12083-020-00993-4.

[11] A. Paranjothi, M. S. Khan, and S. Zeadally, "A survey on congestion detection and control in connected vehicles," *Ad Hoc Netw.*, vol. 108, pp. 1–17, Nov. 2020, doi: 10.1016/j.adhoc.2020.102277.

[12] C. Jiang, H. Zhang, Y. Ren, Z. Han, K.-C. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Commun.*, vol. 24, no. 2, pp. 98–105, Apr. 2017.

[13] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[14] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec. 2016.

[15] 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 Commun.*, vol. 23, no. 5, pp. 120–128, May 2016.

[16] 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," *J. Syst. Archit.*, vol. 98, pp. 289–330, Sep. 2019.

[17] 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," *J. Netw. Comput. Appl.*, vol. 85, no. 1, pp. 116–126, May 2017.

[18] N. Taherkhani and S. Pierre, "Prioritizing and scheduling messages for congestion control in vehicular ad hoc networks," *Comput. Netw.*, vol. 108, pp. 15–28, Oct. 2016.

[19] R. L. Gomes, L. F. Bittencourt, E. R. M. Madeira, E. C. Cerqueira, and M. Gerla, "Software-defined management of edge as a service networks," *IEEE Trans. Netw. Service Manage.*, vol. 13, no. 2, pp. 226–239, Jun. 2016.

[20] S. Zemouri, S. Djahel, and J. Murphy, "An altruistic prediction-based congestion control for strict beaconing requirements in urban VANETs," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 49, no. 12, pp. 2582–2597, Dec. 2019.

[21] 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," *Comput. Electr. Eng.*, vol. 72, pp. 774–788, Nov. 2018.

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

[23] 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, 2018.

[24] 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 Trans. Veh. Technol.*, vol. 66, no. 7, pp. 6115–6129, Jul. 2017.

[25] N. Cheng, N. Lu, N. Zhang, X. S. Zhang, X. Shen, and J. W. Mark, "Opportunistic WiFi offloading in vehicular environment: A game-theory approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 7, pp. 1944–1955, Jul. 2016.

[26] C.-M. Huang, M.-S. Chiang, D.-T. Dao, H.-M. Pai, S. Xu, and H. Zhou, "Vehicle-to-infrastructure (V2I) offloading from cellular network to 802.11p Wi-Fi network based on the software-defined network (SDN) architecture," *Veh. Commun.*, vol. 9, pp. 288–300, Jul. 2017.

[27] F. Falahatraftar, S. Pierre, and S. Chamberland, "A multiple linear regression model for predicting congestion in heterogeneous vehicular networks," in *Proc. 16th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Thessaloniki, Greece, Oct. 2020, pp. 93–98.

[28] B.-M. Cho, M.-S. Jang, and K.-J. Park, "Channel-aware congestion control in vehicular cyber-physical systems," *IEEE Access*, vol. 8, pp. 73193–73203, Apr. 2020.

[29] 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 Inf. Syst.*, vol. 2017, pp. 1–14, Jan. 2017, doi: 10.1155/2017/1248578.

[30] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York, NY, USA: Springer, 2009.

[31] G. Bansal, J. B. Kenney, and C. E. Rohrs, "LIMERIC: A linear adaptive message rate algorithm for DSRC congestion control," *IEEE Trans. Veh. Technol.*, vol. 62, no. 9, pp. 4182–4197, Nov. 2013.

[32] I. Soto, O. Amador, M. Urueña, and M. Calderon, "Strengths and weaknesses of the ETSI adaptive DCC algorithm: A proposal for improvement," *IEEE Commun. Lett.*, vol. 23, no. 5, pp. 802–805, May 2019.

[33] 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 Trans. Mobile Comput.*, vol. 14, no. 7, pp. 1411–1421, Oct. 2014.

[34] 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 *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Maui, HI, USA, Nov. 2018, pp. 2575–2582.

[35] F. Hagenauer, F. Dressler, and C. Sommer, "Poster: A simulator for heterogeneous vehicular networks," in *Proc. IEEE Veh. Netw. Conf. (VNC)*, Paderborn, Germany, Dec. 2014, pp. 185–186.

[36] *OpenStreetMap Contributors*. Accessed: May 2021. [Online]. Available: https://www.openstreetmap.org

[37] K. T. Herring, J. W. Holloway, D. H. Staelin, and D. W. Bliss, "Path-loss characteristics of urban wireless channels," *IEEE Trans. Antennas Propag.*, vol. 58, no. 1, pp. 171–177, Jan. 2010.

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

[39] M. Joseph, X. Liu, and A. Jaekel, "An adaptive power level control algorithm for DSRC congestion control," in *Proc. 8th ACM Symp. Design Anal. Intell. Veh. Netw. Appl. (DIVANet)*, Montreal, QC, Canada, Oct. 2018, pp. 57–62, doi: 10.1145/3272036.3272041.

[40] C. Navdeti, C. Giri, and I. Banerjee, "Distributed α-fair transmit power adaptation based congestion control in VANET," in *Proc. 20th Int. Conf. Distrib. Comput. Netw.*, Bangalore, India, Jan. 2019, pp. 253–260, doi: 10.1145/3288599.3288606.

[41] S. A. A. Shah, E. Ahmed, J. J. P. C. Rodrigues, I. Ali, and R. M. Noor, "Shapely value perspective on adapting transmit power for periodic vehicular communications," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 3, pp. 977–986, Mar. 2018.

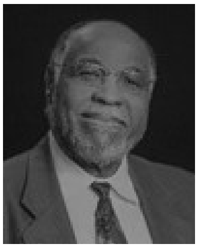
[42] P. Ignaciuk, and A. Bartoszewicz, *Congestion Control in Data Transmission Networks: Sliding Mode and Other Designs*. London, U.K.: Springer, 2013, pp. 3–4.

[43] J. Davis and M. Goadrich, "The relationship between precision-recall and ROC curves," in *Proc. 23rd Int. Conf. Mach. Learn. (ICML)*, Pittsburgh, PA, USA, 2006, pp. 233–240, doi: 10.1145/1143844.1143874.

[44] A. Géron, *Hands-On Machine Learning With Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2019, pp. 88–100.

[45] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: With Applications in R*. New York, NY, USA: Springer, 2013, pp. 181–183.
[46] N. Taherkhani, "Congestion control in vehicular ad hoc networks," Ph.D. dissertation, École Polytechnique de Montréal, Montreal, QC, Canada, 2015.
[47] L. Delgrossi, and T. Zhang, *Vehicle Safety Communications: Protocols, Security, and Privacy*. Hoboken, NJ, USA: Wiley, 2012, pp. 85–91.
[48] Q. D. La, M. V. Ngo, T. Q. Dinh, T. Q. S. Quek, and H. Shin, "Enabling intelligence in fog computing to achieve energy and latency reduction," *Digital Commun. Netw.*, vol. 5, no. 1, pp. 3–9, Feb. 2019.

FARNOUSH FALAHATRAFTAR received the B.Sc. degree in computer software engineering from the South Tehran Branch of Islamic Azad University, Tehran, Iran, in 2009, and the M.Sc. degree in computer engineering from Eastern Mediterranean University, Famagusta, North Cyprus, in 2013. She is currently pursuing the Ph.D. degree with École Polytechnique de Montréal, Montreal, QC, Canada. Her research interests include controlling congestion in heterogeneous vehicular networks using machine learning and deep learning methods, SDN, network slicing, and virtualization in 5G and beyond.



SAMUEL PIERRE (Senior Member, IEEE) received an Honorary Doctorate degree from the Université du Quebec à Trois-Rivieres (UQTR), in May 2014, and an Honorary Doctorate degree from the Université du Quebec en Outaouais (UQO), in November 2016. He is currently a Professor with the Department of Computer and Software Engineering, Polytechnique Montreal, and the Director of the Mobile Computing and Networking Research Laboratory (LARIM). He has authored or coauthored more than 550 technical publications, including articles in refereed archival journals, textbooks, patents, and book chapters. His research interests include wired and wireless communications, mobile computing and networking, cloud computing, and e-learning. He is a fellow of the Engineering Institute of Canada, in 2003, and the Canadian Academy of Engineering, in 2008. He was appointed as a member of the Order of Canada, in December 2011. He has received several awards, including the Prix Poly 1873 for Excellence in Teaching and Training, in 2001 and 2005, and the Knight of the National Order of Quebec, in 2009. In 2017, he received the El Fasi Prize from the Agence Universitaire de la Francophonie (AUF) to highlight the action of a person who has exerted a significant influence through the quality of his expertise and the innovative nature of his achievements at the international level in the fields of research, training, development and international cooperation, governance and/or transfer of knowledge or skills. In 2020, he received the Grand Prize for Professional Excellence from the Order of Engineers of Quebec (OIQ). In 2021, he received the Gold Medal from Engineers Canada.

STEVEN CHAMBERLAND (Member, IEEE) received the Electrical Engineering degree and the Ph.D. degree in operations research from the École Polytechnique de Montréal, Montreal, QC, Canada, in 1994 and 1998, respectively. From 1998 to 1999, he was with the Network Architecture Group, Bell Canada, and from 1999 to 2001, he was a Professor with the École de Technologie Supérieure, Montreal. Since 2001, he has been an Assistant Professor and an Associate Professor with the École Polytechnique de Montréal, where he is currently a Full Professor of computer engineering. His research interests include telecommunication network planning and design and network architecture. He is a member of the Association for Computing Machinery (ACM). He also serves on the Editorial Board for *Computers and Operations Research* and as an Associate Editor for *Information Systems and Operational Research*.