telecom

MDPI

Article

A Conditional Generative Adversarial Network Based Approach for Network Slicing in Heterogeneous Vehicular Networks

Farnoush Falahatraftar *, Samuel Pierre and Steven Chamberland

Mobile Computing and Networking Research Laboratory (LARIM), Department of Computer and Software Engineering, Polytechnique de Montréal, Montreal, QC H3T 1J4, Canada; samuel.pierre@polymtl.ca (S.P.); steven.chamberland@polymtl.ca (S.C.)

* Correspondence: farnoush.falahatraftar@polymtl.ca

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

Citation: Falahatraftar, F.; Pierre, S.; Chamberland, S. A Conditional Generative Adversarial Network Based Approach for Network Slicing in Heterogeneous Vehicular Networks. *Telecom* **2021**, *2*, 141–154. https://doi.org/10.3390/telecom2010009

Received: 16 January 2021
Accepted: 15 March 2021
Published: 18 March 2021

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 [1]. 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 [2].

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. [3] 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 [4]. 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 [4–7].

Regarding the network traffic states, which are defined as safe, warning, and congested in [8]; 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 [9–11] 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 [12]. 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 [13,14].

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.

2. Related Work

Network slicing has been recently considered in a number of research works, particularly for vehicular networks. In [15]; the authors gave priority to the safety data. A queuing method was used to categorize the traffic data based on the priority. Fur-

thermore, 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 [16]. 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 [17], 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 [18], 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 [19], 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 [20], 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 [21], 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, [22], 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 [23,24]; however, the vehicular network is not the targeted network in these studies. In [23], 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 [24], deep reinforcement learning was been applied to allocate resources in the network slices. In [25], the authors showed that the network slicing could benefit from the proposed traffic prediction method using AI.

In [26], 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 [27], 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.

3. Methodology

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 [11,28–33]. Because no data sets from real HetVNET are available today, we have generated a data set that contains data records of these parameters [8]. We used OpenStreetMap (OSM) [34] to have a map that is similar to the real

environment in terms of intersections, streets, buildings, etc. Figure 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. [35]. SUMO requires ".osm" file created by OSM to generate the vehicle traffic on the map. Simultaneously, Veins LTE version 1.3 [36] 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 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 1.

Table 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	

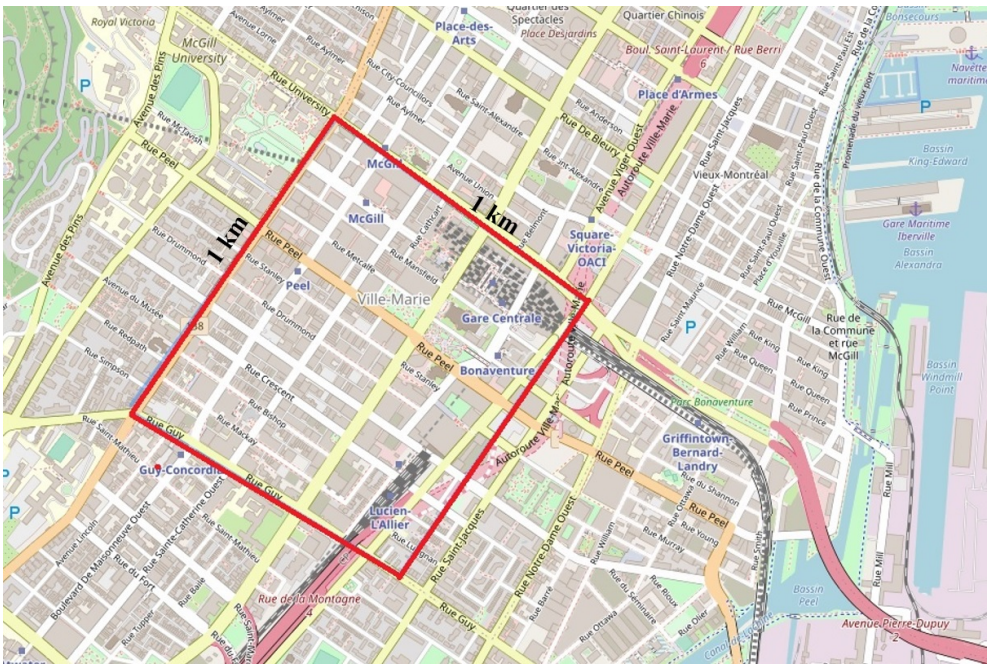

Figure 1. Simulated area is shown by a red square.

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 [8]. 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 are selected as a successful scenario [37]. 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 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.

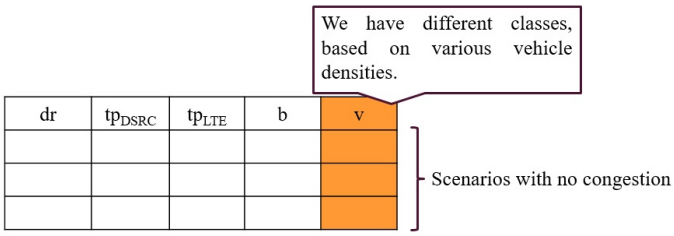

Figure 2. Structure of data set used for the Conditional Generative Adversarial Network (CGAN).

3.2. Proposing CGAN Model for HetVNET

CGAN uses a generative model, like G, to create new data from noise [38]. 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 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 [38], as follows:

$$\min_{G} \max_{D} V(D,G) = \mathbb{E}_{x\sim p_{data(x)}}[logD(x|v)] + \mathbb{E}_{z\sim p_{z(z)}}[log(1-D(G(z|v)))], \quad (1)$$

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.

A uniform distribution $U(0,1)$ is used to generate random noise while using the generator. Moreover, we used the Adam optimizer [39], 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 [39], 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 [39]. Besides, He initialization is applied, which best fits the Leaky ReLU activation function [39]. Tanh and Sigmoid activation functions are used

for the output layer in the generator and the discriminator, respectively. Figure 4 shows the layers of the CGAN.

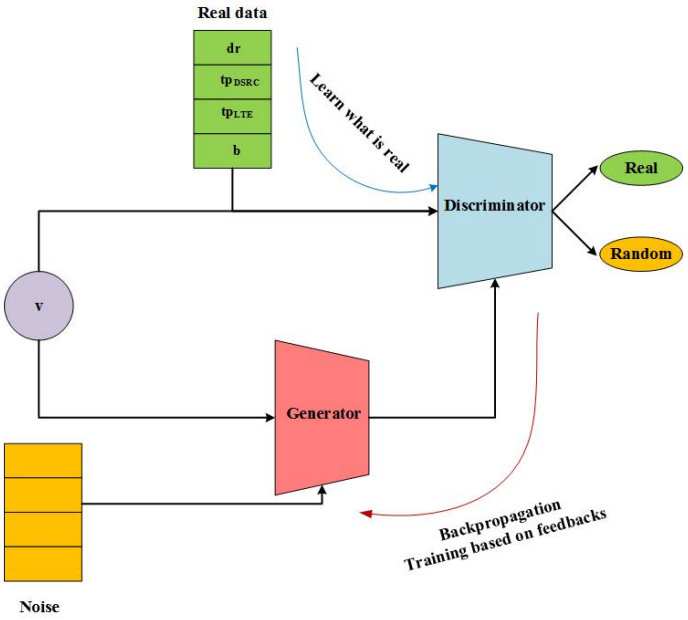

Figure 3. CGAN model for HetVNET.

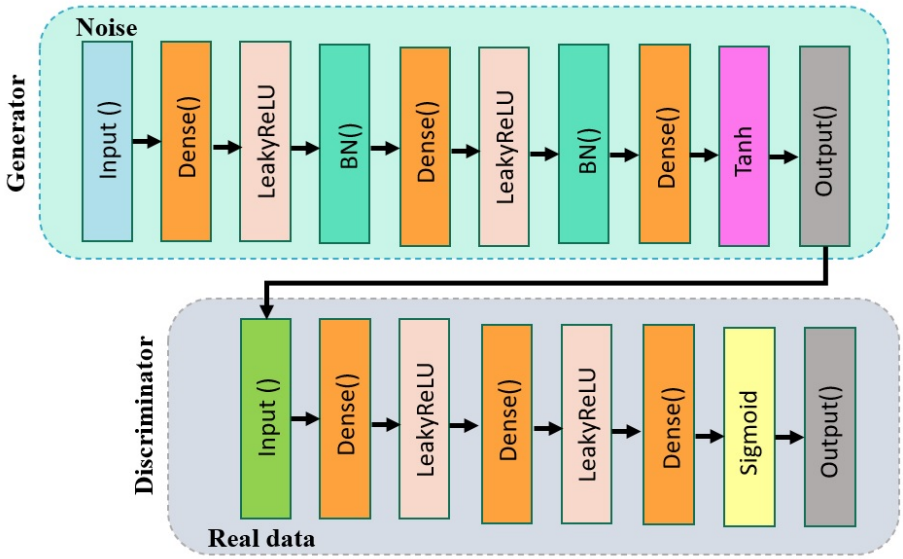

Figure 4. The architecture of the proposed CGAN that shows generator's layers and discriminator's layers.

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 [12].

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 [11,28–33]. 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) [40] 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 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.

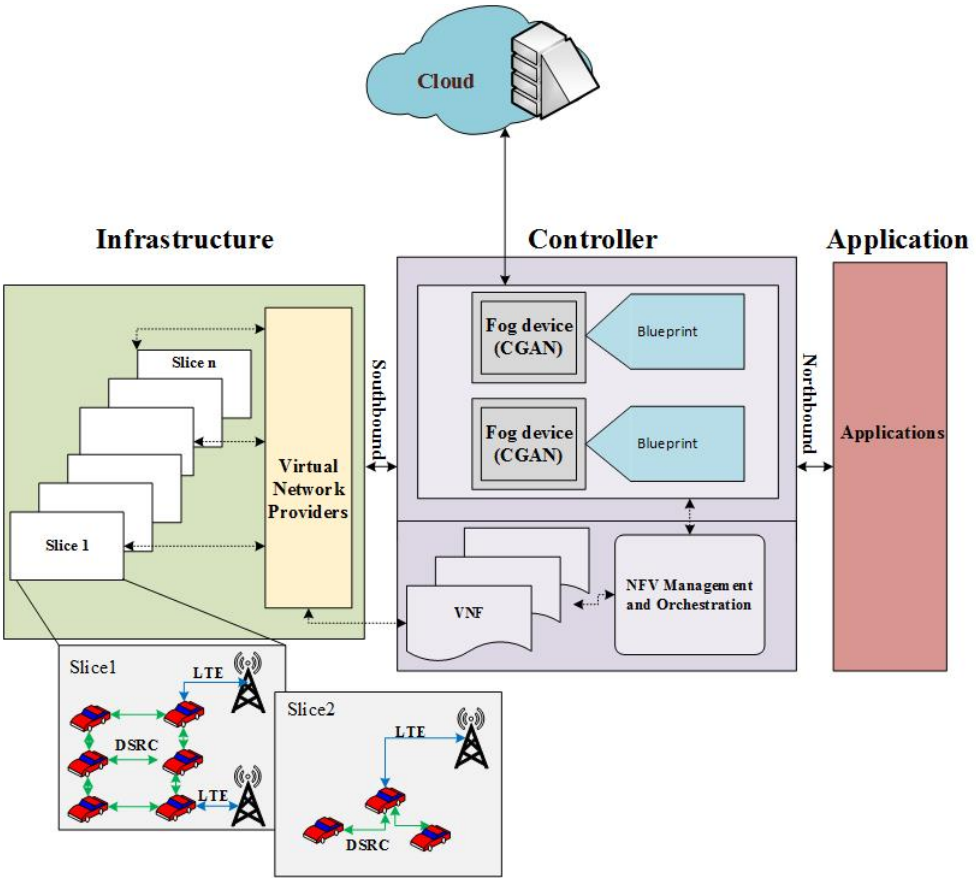

Figure 5. An intelligent hybrid CGAN-Software-Defined Network (CGAN-SDN) architecture for HetVNET.

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 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 5, if 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 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.

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 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, 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.

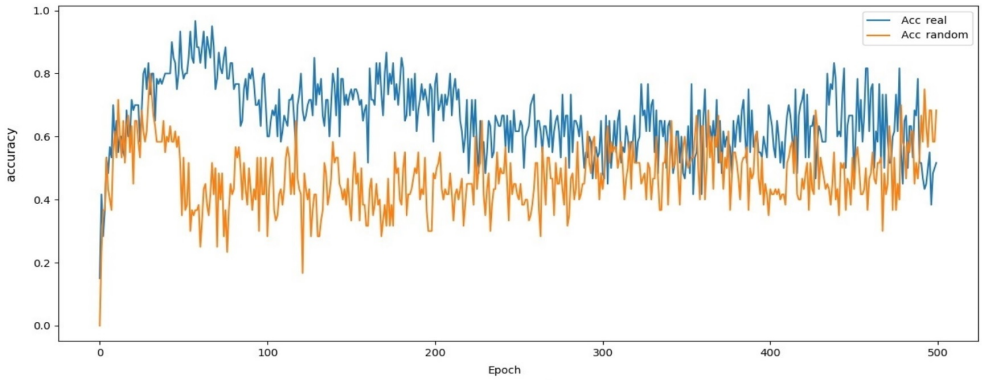

Figure 6. The discriminator's accuracy over time.

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 7 and 8 present the results. Figure 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 [39]; 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 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.

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 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.

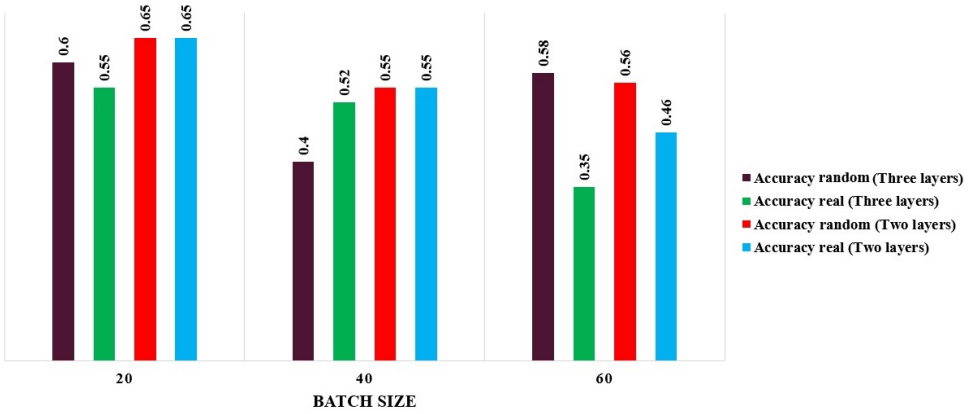

Figure 7. The discriminator's accuracy per various batch sizes and hidden layers.

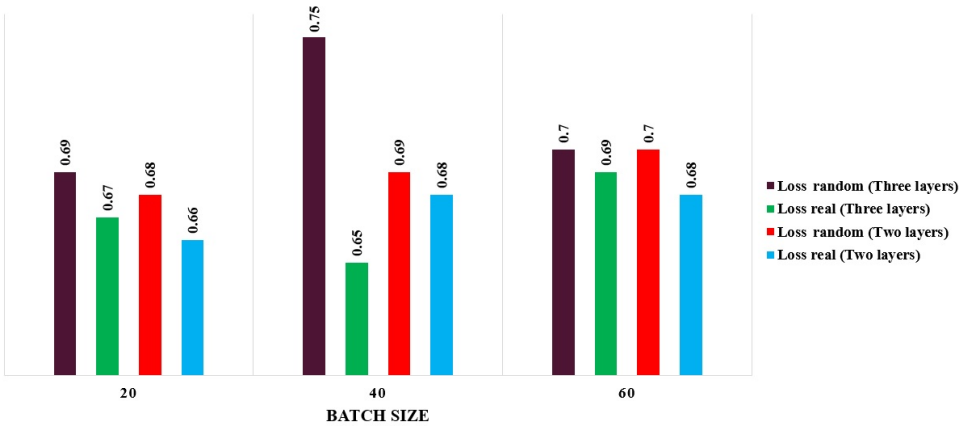

Figure 8. The discriminator's loss per various batch sizes and hidden layers.

Figure 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.

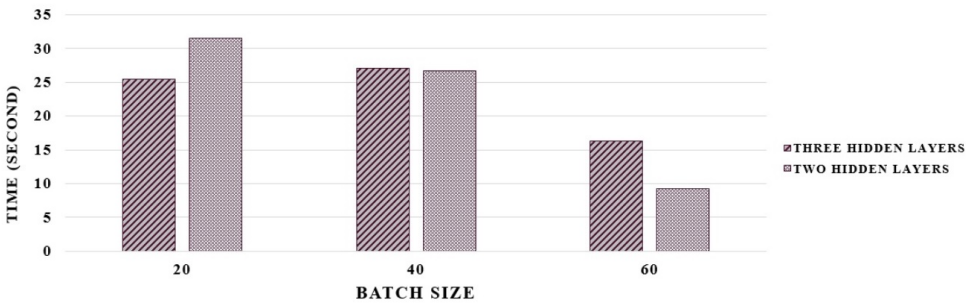

Figure 9. Proposed CGAN model's training time.

Based on Figures 7–9, 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.

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.

Author Contributions: Conceptualization, methodology and analysis, F.F. and S.P.; writing—original draft preparation, F.F.; writing—review and editing, S.P. and S.C.; supervision, S.P. and S.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data available on request. Besides, the OSM data is available under the Open Database License at https://www.openstreetmap.org/copyright.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Duan, Q.; Ansari, N.; Toy, M. Software-defined network virtualization: An architectural framework for integrating SDN and NFV for service provisioning in future networks. *IEEE Netw.* **2016**, *30*, 10–16. [CrossRef]
2. Herrera, J.G.; Botero, J.F. Resource allocation in NFV: A comprehensive survey. *IEEE Trans. Netw. Serv. Manag.* **2016**, *13*, 518–532. [CrossRef]
3. Gomes, R.L.; Bittencourt, L.F.; Madeira, E.R.M.; Cerqueira, E.C.; Gerla, M. Software-defined management of edge as a service networks. *IEEE Trans. Netw. Serv. Manag.* **2016**, *13*, 226–239. [CrossRef]
4. Zhang, H.; Liu, N.; Chu, X.; Long, K.; Aghvami, A.-H.; Leung, V.C. Network slicing based 5g and future mobile networks: mobility, resource management, and challenges. *IEEE Commun. Mag.* **2017**, *55*, 138–145. [CrossRef]
5. Zhou, X.; Li, R.; Chen, T.; Zhang, H. Network slicing as a service: Enabling enterprises' own software-defined cellular networks. *IEEE Commun. Mag.* **2016**, *54*, 146–153. [CrossRef]
6. Rost, P.; Mannweiler, C.; Michalopoulos, D.S.; Sartori, C.; Sciancalepore, V.; Sastry, N.; Holland, O.; Tayade, S.; Han, B.; Bega, D.; et al. Network slicing to enable scalability and flexibility in 5G mobile networks. *IEEE Commun. Mag.* **2017**, *55*, 72–79. [CrossRef]
7. Foukas, X.; Patounas, G.; Elmokashfi, A.; Marina, M.K. Network slicing in 5G: Survey and challenges. *IEEE Commun. Mag.* **2017**, *55*, 94–100. [CrossRef]
8. Falahatraftar, F.; Pierre, S.; Chamberland, S. A Multiple Linear Regression Model for Predicting Congestion in Heterogeneous Vehicular Networks. In Proceedings of the 2020 16th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Thessaloniki, Greece, 12–14 October 2020; pp. 93–98.
9. Soto, I.; Amador, O.; Urueña, M.; Calderon, M. Strengths and weaknesses of the ETSI adaptive DCC algorithm: A proposal for improvement. *IEEE Commun. Lett.* **2019**, *23*, 802–805. [CrossRef]
10. Amador, O.; Soto, I.; Calderon, M.; Urueña, M. Experimental Evaluation of the ETSI DCC Adaptive Approach and Related Algorithms. *IEEE Access* **2020**, *8*, 49798–49811. [CrossRef]

11. Lyamin, N.; Vinel, A.; Smely, D.; Bellalta, B. ETSI DCC: Decentralized Congestion Control in C-ITS. *IEEE Commun. Mag.* **2018**, *56*, 112–118. [CrossRef]
12. Shen, X.; Gao, J.; Wu, W.; Lyu, K.; Li, M.; Zhuang, W.; Li, X.; Rao, J. AI-assisted network-slicing based next-generation wireless networks. *IEEE Open J. Veh. Technol.* **2020**, *1*, 45–66. [CrossRef]
13. Jiacheng, C.; Haibo, Z.; Ning, Z.; Peng, Y.; Lin, G.; ; Sherman, S.X. Software defined internet of vehicles: Architecture, challenges and solutions. *J. Commun. Inf. Netw.* **2016**, *1*, 14–26. [CrossRef]
14. Davy, S.; Famaey, J.; Serrat, J.; Gorricho, J.L.; Miron, A.; Dramitinos, M.; Neves, P.M.; Latre, S.; Goshen, E. Challenges to support edge-as-a-service. *IEEE Commun. Mag.* **2014**, *52*, 132–139. [CrossRef]
15. Tayyaba, S.K.; Khattak, H.A.; Almogren, A.; Shah, M.A.; Din, I.U.; Alkhalifa, I.; Guizani, M. 5G Vehicular Network Resource Management for Improving Radio Access Through Machine Learning. *IEEE Access* **2020**, *8*, 6792–6800.
16. Campolo, C.; Molinaro, A.; Iera, A.; Fontes, R.R.; Rothenberg, C.E. Towards 5G network slicing for the V2X ecosystem. In Proceedings of the 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft), Montreal, QC, Canada, 25–29 June 2018; pp. 400–405.
17. Cui, Y.; Zheng, H.; Wang, H.; Wu, D. An Intelligent Coordinator Design for Network Slicing in Service-Oriented Vehicular Networks. In Proceedings of the GLOBECOM 2020–2020 IEEE Global Communications Conference, Taipei, Taiwan, 7–11 December 2020; pp. 1–6.
18. Khan, H.; Samarakoon, S.; Bennis, M. Enhancing video streaming in vehicular networks via resource slicing. *IEEE Trans. Veh. Technol.* **2020**, *69*, 3513–3522. [CrossRef]
19. Mei, J.; Wang, X.; Zheng, K. Intelligent Network Slicing for V2X Services Toward 5G. *IEEE Netw.* **2019**, *33*, 196–204. [CrossRef]
20. Chen, Y.; Wang, Y.; Liu, M.; Zhang, J.; Jiao, L. Network Slicing Enabled Resource Management for Service-Oriented Ultra-Reliable and Low-Latency Vehicular Networks. *IEEE Trans. Veh. Technol.* **2020**, *69*, 7847–7862. [CrossRef]
21. Khan, H.; Luoto, P.; Bennis, M.; Latva-aho, M. On the Application of Network Slicing for 5G-V2X. In Proceedings of the 24th European Wireless Conference, Catania, Italy, 2–4 May 2018; pp. 1–6.
22. Sanchez-Iborra, R.; santa, J.; Gallego-Madrid, J.; Covaci, S.; Skarmeta, A. Empowering the Internet of vehicles with multi-RAT 5G network slicing. *Sensors* **2019**, *19*, 3107. [CrossRef]
23. Wu, H.W.Y.; Min, G.; Xu, J.; Tang, P. Data-driven dynamic resource scheduling for network slicing: A deep reinforcement learning approach. *Inf. Sci.* **2019**, *498*, 106–116.
24. Koo, J.; Mendiratta, V.B.; Rahman, M.R.; Walid, A. Deep reinforcement learning for network slicing with heterogeneous resource requirements and time varying traffic dynamics. In Proceedings of the 2019 15th International Conference on Network and Service Management (CNSM), Halifax, NS, Canada, 21–25 October 2019; pp. 1–5.
25. Cui, Y.; Huang, X.; Wu, D.; Zheng, H. Machine Learning based Resource Allocation Strategy for Network Slicing in Vehicular Networks. In Proceedings of the 2020 IEEE/CIC International Conference on Communications in China (ICCC), Chongqing, China, 9–11 August 2020; pp. 454–459.
26. Xiong, K.; Leng, S.; Hu, J.; Chen, X.; Yang, K. Smart network slicing for vehicular fog-RANs. *IEEE Trans. Veh. Technol.* **2019**, *68*, 3075–3085. [CrossRef]
27. Khan, H.; Luoto, P.; Samarakoon, S.; Bennis, M.; Latva-Aho, M. Network slicing for vehicular communication. *Trans. Emerg. Telecommun. Technol.* **2019**, e3652. [CrossRef]
28. Zemouri, S.; Djahel, S.; Murphy, J. An altruistic prediction-based congestion control for strict beaconing requirements in urban VANETs. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**, *49*, 2582–2597. [CrossRef]
29. Zhang, F.; Du, Y.; Liu, W.; Li, P. Model predictive power control for cooperative vehicle safety system. *IEEE Access* **2018**, *6*, 4797–4810. [CrossRef]
30. Joseph, M.; Liu, X.; Jaekel, A. 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, 25 October 2018; pp. 57–62.
31. Shah, S.A.A.; Ahmed, E.; Rodrigues, J.; Ali, I.; Noor, R.M. Shapely value perspective on adapting transmit power for periodic vehicular communications. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 977–986. [CrossRef]
32. Sharma, S.; Chahal, M.; Harit, S. Transmission Rate-based Congestion Control in Vehicular Ad Hoc Networks. In Proceedings of the 2019 Amity International Conference on Artificial Intelligence (AICAI), Dubai, United Arab Emirates, 4–6 February 2019; pp. 303–307.
33. Cho, B.M.; Jang, M.S.; Park, K.J. Channel-Aware Congestion Control in Vehicular Cyber-Physical Systems. *IEEE Access* **2020**, *8*, 73193–73203. [CrossRef]
34. OpenStreetMap Contributors. 2017. Available online: https://planet.osm.org; https://www.openstreetmap.org (accessed on 15 March 2021).
35. Lopez, P.A.; Behrisch, M.; Bieker-Walz, L.; Erdmann, J.; Flötteröd, Y.P.; Hilbrich, R.; Lücken, L.; Rummel, J.; Wagner, P.; Wiessner, P. Microscopic traffic simulation using sumo. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 2575–2582.
36. Hagenauer, F.; Dressler, F.; Sommer, C. Poster: A simulator for heterogeneous vehicular networks. In Proceedings of the 2014 IEEE Vehicular Networking Conference (VNC), Paderborn, Germany, 3–5 December 2014; pp. 185–186.

37. Ignaciuk, P.; Bartoszewicz, A. Data Transfer Concepts and Congestion. In *Congestion Control in Data Transmission Networks: Sliding Mode and Other Designs*; Springer: London, UK, 2012; pp. 2–12.
38. Mirza, M.; Osindero, S. Conditional generative adversarial nets. *arXiv* **2014**, arXiv:1411.1784.
39. Ǵeron, A. Data Transfer Concepts and Congestion. In *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 2nd ed.; O'Reilly Media: Sebastopol, CA, USA, 2019; pp. 279–598.
40. Alliance, N. *Description of Network Slicing Concept*; NGMN 5G P; Next Generation Mobile Networks Ltd.: Frankfurt, Germany; Hesse, Germany, 2016; Volume 1.