sensors

MDPI

Article

An Intelligent Multimode Clustering Mechanism Using Driving Pattern Recognition in Cognitive Internet of Vehicles

Huigang Chang [1,*] and Nianwen Ning [2]

[1] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
[2] School of Artificial Intelligence, Henan University, Zhengzhou 450046, China; nnw@henu.edu.cn
* Correspondence: changhuigang@bupt.edu.cn

Abstract: Connected autonomous vehicles can leverage communication and artificial intelligence technologies to effectively overcome the perceived limitations of individuals and enhance driving safety and stability. However, due to the high dynamics of the vehicular network and frequent interruptions and handovers, it is still challenging to provide stable communication connections between vehicles, which is likely to cause disasters. To address this issue, in this paper, we propose an intelligent clustering mechanism based on driving patterns in heterogeneous Cognitive Internet of Vehicles (CIoVs). In the proposed approach, we analyze the driving mode containing multiple feature parameters to accurately capture the driving characteristics. To ensure the accuracy of pattern recognition, a genetic algorithm-based neural network pattern recognition algorithm is proposed to support the reliable clustering of connected autonomous vehicles. The cognitive engines recognize the driving modes to group vehicles with a similar driving mode into a relatively stable cluster. In addition, we formulate the stability and survival time of clusters and analyze the communication performance of the clustering mechanism. Simulation results show that the proposed mechanism improves the reliable communication throughput and average cluster lifetime by approximately 14.4% and 11.5% respectively compared to the state-of-the-art approaches.

Keywords: Cognitive Internet of Vehicles; artificial intelligence; autonomous driving; genetic algorithm; clustering mechanism

check for updates

Citation: Chang, H.; Ning, N. An Intelligent Multimode Clustering Mechanism Using Driving Pattern Recognition in Cognitive Internet of Vehicles. *Sensors* **2021**, *21*, 7588. https://doi.org/10.3390/s21227588

Academic Editor: Claudia Campolo

Received: 10 October 2021
Accepted: 8 November 2021
Published: 15 November 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The Intelligent Transportation System (ITS) benefits from communication and intelligence technologies to enhance traffic management capabilities and improve vehicle driving efficiency. Autonomous driving (AD) as an important component of ITS has received widespread attention due to its ability to avoid traffic jams, reduce the probability of traffic accidents, and free human hands, which will produce significant economic and social benefits. At present, the Connected Autonomous Vehicles (CAVs) employ vehicular communication technology to exchange environmental and perception information to obtain non-line-of-sight (NLOS) traffic information to strengthen its own perception capabilities [1]. In addition, the analysis of a large amount of sensory data in network space and physical space can effectively assist CAVs in AD decision making [2–5]. However, the traditional Internet of Vehicles (IoV) merely provides communication services and lacks efficient cognition of physical space and network space, thus preventing intelligent vehicle mobility management and collaboration to better support AD services [6,7].

Recently, the Cognitive Internet of Vehicle (CIoVs) has emerged as an innovative paradigm for bringing intelligence and cognitive capabilities to IoV, which can provide smarter, reliable, and proactive services in AD scenarios [8]. CIoVs emphasizes the use of cognitive methods to facilitate intelligent collaboration in physical and network space to ensure traffic safety and quality of communication services. The cognitive engine (CE) can assist decision making and guide the movement of CAVs. In addition, the powerful

CAVs also provide a wealth of onboard communication, sensing, and computing resources for AD scenarios. For instance, CAVs can share safety warnings, perceptual information, and control instructions to ensure the safety of AD. Furthermore, orchestrating CAVs into different clusters through active collaboration can support more efficient and safer swarm intelligence services and reduce fuel consumption [9–11].

Although CIoVs can make AD services and applications become more intelligent and efficient, there are still changes in vehicle mobility management and reliable connectivity of communications. Firstly, dynamic changes in the vehicular network topology may cause unreliable and unstable communication connectivity. Since CIoVs contain a large number of connected devices, many applications are processed locally or at the edge, which requires a more flexible network architecture. In addition, as safety and non-safety services require different network performance, singular communication technology cannot meet the quality of communication services. Furthermore, the mobility of CAVs leads to frequent interruptions and handovers in communication due to changes in access points. It is difficult to guarantee a continuous and consistent communication service, which is essential for the security of network-assisted AD [12–14].

To address these issues, in this paper, we propose an intelligent clustering mechanism based on driving behaviors in heterogeneous CIoVs, which introduces CE to recognize AD mode and perform clustering to better support network-assisted AD scenarios. We consider a connected autonomous driving scenario where the vehicles are CAVs equipped with communication devices to investigate the communication performance gains obtained through intelligent clustering among vehicles. Currently, fully connected AD is not yet completely accessible due to policy, regulation, and technical constraints. However, the powerful sensing capabilities of CAVs and Artificial Intelligence (AI) technologies can already support both non-connected and connected vehicles coexistence scenario. With the further development of technology and improvement of policies and regulations, the penetration rate of CAVs will further increase and eventually achieve a fully driverless scenario.

In the proposed mechanism, the CE analyzes driving behavior to classify CAVs into different clusters based on the similarity metrics of AD modes. In addition, a leadership assessment indicator is designed to select the CAVs with the highest leadership priority as the cluster head (CH). Different communication technologies are utilized inside and outside the cluster, which has dedicated short-range communication (DSRC) and 5G-V2I technologies [15,16]. It enables high connectivity and wide area coverage to meet the communication requirements of both safety and non-safety services [17,18]. The proposed mechanism allows for more stable clusters through CAVs collaboration, reduces communication interruptions and switching caused by random movements, and ensures communication connectivity and reliability. The main contributions of this paper are summarized as follows:

- We design a heterogeneous CIoVs network architecture and propose an intelligent clustering mechanism using AD pattern recognition, which brings CAVs with the same or a similar driving mode together to form a stable cluster to enhance the connectivity of the communication service.
- We establish an AD mode model considering multiple driving parameter factors to perform AD pattern recognition. Then, the stability and survival time metrics of clusters are formulated and analyze the communication performance of the proposed clustering mechanism.
- To ensure the effectiveness of clustering, we propose a genetic algorithm (GA)-based neural network (GANN) AD pattern recognition algorithm to perform accurate AD mode recognition for reliable clustering.
- The simulation results are discussed and analyzed to evaluate the effectiveness of the proposed clustering mechanism compared with existing schemes in terms of AD pattern recognition accuracy, communication throughput, and average cluster lifetime.

The rest of the paper is organized as follows. Related works are introduced in Section 2. Section 3 presents the network architecture, clustering mechanism, and driving behavior modeling. Section 4 describes the GANN algorithm for AD pattern recognition in detail. Section 5 analyzes the performance indicators of the proposed clustering mechanism. Simulation results and discussions are carried out in Section 6. Finally, we summarize this paper in Section 7.

2. Related Work

As important components of smart city and ITS, IoV and AD have attracted extensive interest from both industry and academia. In this section, we discuss related work from the perspective of the vehicular network architecture, the network clustering and switching mechanism in IoV, and vehicle clustering and the driving behavior modeling approach that are closely related to this paper. The comparative study of our work with existing work is listed in Table 1.

Table 1. Comparative study of our work with existing works.

Subject	Ref.	Use Case	Purpose	Contributions
Network architecture	[19]	Vehicular network	Guaranteeing timely data processing or service access	Proposing distributed fog computing for scheduling delay-sensitive applications
	[20]	Vehicular communication	Minimizing transmission costs and fully utilizing communication resources	Proposing a deep Q -learning approach to optimize transmission scheduling
	[21]	Vehicular communication	Improving reliability performance of heterogeneous vehicular communications	Proposing a nearest-first relaying system in the coexistence situation
	[22]	IoV	Evaluating the IoV performance of the scalability and reliability	Modeling and analyzing the performance of V2V and hybrid V2V/V2I networks
	[23]	Vehicular network	Meeting the communication requirements of the ITS	Proposing a heterogeneous vehicular network with V2I and V2V communications
Network clustering and switching	[24]	Data collection	Guaranteeing QoS and reduce cellular bandwidth cost	Proposing a one-hop clustering and data delivery optimization approach
	[25]	Vehicular communication	Providing high reliability, low latency, and wide area coverage	Proposing a self-adaptive clustering method to adjust clusters
	[26]	Vehicular networks	Analyzing the cluster-based heterogeneous vehicular networks	Developing a heterogeneous vehicular networks framework
	[27]	Cognitive radio	Increasing network scalability in a distributed cognitive radio network	Proposing a novel reinforcement learning based trust model to adjust cluster size
	[28]	Network selection	Improving quality of service of vehicles	Developing a network recommendation system through big data analysis
	[29]	Vehicular networks	Making load balancing among heterogeneous base stations	Proposing an online reinforcement learning approach for network load balancing
Vehicle clustering and driving behavior	[30]	Autonomous vehicle	Clustering a wide range of driving encounter scenarios	Proposing a generic unsupervised learing framework
	[31]	IoV	Dealing with large scale network-based trajectory data	Designing a dynamic network representation learning based clustering method
	[32]	Transportation management	Extracting travel patterns from large-scaled vehicle trajectories	Developing a calculation process for clustering vehicle trajectory data
	[33]	ITS	Achieving traffic flow control in Social IoV	Presenting a social vehicle route selection algorithm to select optimal route
	[34]	Autonomous driving	Designing safe, smart, and personalized autonomous driving systems	Proposing a driving model to emulate a human decision making mechanism
	[35]	Driving behavior	Supporting automated driving systems and human-like decision making	Proposing a unified model of driving behavior in different scenarios
	[36]	Driving behavior	Identifying abnormal driving behaviors to prevent traffic accidents	Proposing different abnormal driving behavior recognition algorithms
	[37]	ITS	Monitoring driver behavior to reduce traffic accident risk	Presenting a novel deep learning method for analyzing driver behavior
	Our work	Autonomous driving	Providing stable communication connections between vehicles	Proposing an intelligent clustering mechanism based on driving patterns

2.1. IoV Network Architecture

Lin et al. [19] proposed a software-defined networking (SDN) enabled vehicular network architecture which divided the networks into three layers to improve network management capacity. In [20], the authors proposed a cognitive radio (CR)-based architecture for in-vehicle networks to employ CR technology to alleviate spectrum resource shortages in the presence of highly dynamic typologies and time-varying spectrum utilization. They used reinforcement learning methods to design optimal data transmission scheduling schemes and make full use of V2V and V2I communication resources. To deal with the problem of a lack of communication between vehicles, the authors in [21] proposed a heterogeneous V2V communication architecture and investigated a relaying method for the coexistence of DSRC and LTE-V2V technologies. In [22], the authors investigated the performance of the heterogeneous IoV architecture with multiple communication modes, which included V2V-only and hybrid situations. In [23], the authors surveyed a heterogeneous vehicular network architecture, which utilized multiple wireless communication technologies. Through analyzing and comparing the performance indicators required by safety services and non-safety services, they pointed out that a heterogeneous vehicular network combined with a cellular network and DSRC would be a potential solution to meet the communication requirements of ITS. These articles have put forward some communication optimization schemes from the aspect of vehicular network architecture and data transmission, which include multiple communication technologies and modes. However, the dynamic changes in network topology and communication interruptions and handovers caused by the random mobility of vehicles still challenge the reliability and scalability of IoV.

2.2. Network Clustering and Switching in IoV

Qi et al. [24] proposed a traffic differentiated clustering routing mechanism for vehicular data collection to reduce cellular bandwidth cost in a heterogeneous network, which was compose of DSRC and cellular vehicle-to-everything. To achieve high reliability, low latency, and wide-area coverage communication, the authors in [25] proposed an adaptive clustering method, which utilized DSRC and the cellular network to provide intra-cluster V2V communication and V2I communication outside the cluster, respectively. In [26], the authors proposed a section-based cluster mechanism, which clustered vehicles based on road sections and selected the vehicle closest to the cluster center as the CH, ignoring the driving stability of vehicles. In [27], the authors proposed a dynamic cluster adjustment mechanism to improve the scalability of the vehicular network, where CH discovered malicious cluster member (CM) nodes and adjusted cluster size according to the available spectrum to maximize resource utilization. The authors in [28] proposed a recommendation and switching mechanism in heterogeneous IoV to alleviate the load on the cellular network and improve network performance. In [29], the authors proposed an online reinforcement learning approach, which enabled switching between different heterogeneous vehicular networks by observing the patterns of data traffic in the spatial-temporal dimension. These studies investigated the communication switching problem caused by the dynamic network topology of the IoV from the perspective of clustering of heterogeneous networks. However, they neglected the impact of driving behavior and patterns of vehicles in heterogeneous vehicular networks on the communication performance, which is crucial for the stability of vehicle clusters and the reliability of communication.

2.3. Vehicle Clustering and Driving Behavior Modeling

Wang et al. [30] proposed a classification method for driving encounter scenarios using connected vehicle trajectories. A generic unsupervised learning framework including a feature representation layer and a clustering layer was designed to cluster driving encounter scenes based on multi-vehicle GPS trajectories. In [31], the authors employed machine learning methods to cluster the vehicle trajectories from a network perspective, which constructed IoV based on k-nearest neighbor. In [32], the authors adopted density-based

spatial clustering of application with a noise algorithm to analyze travel patterns in a city and evaluate the similarity between trajectories. In [33], the authors proposed a clustering approach based on history and current driving information for ITS, which utilized the social relationship to perform a clustering algorithm. In addition, a vehicle route selection algorithm based on game evolution was proposed to control the traffic flow. In [34], the authors employed an internal reward function-based driving model to simulate human decision-making mechanisms and proposed a structural hypothesis on human driving behavior focused on discrete potential driving intentions. Tan et al. [35] proposed a new approach to unify the modeling of driving behavior in different scenarios, which combined behavioral theories and field theory. Liu et al. [36] investigated a driving behavior scoring model based on entropy weighting and hierarchical analysis, which analyzed driving behavior data by establishing a driving behavior identification algorithm. Shahverdy et al. [37] proposed a convolutional neural network based method to recognize five types of driving styles according to vehicle signals instead of monitoring driver visual features. These articles focused on vehicle clustering in physical space and behavioral analysis in a single traffic scenario; however, they lacked joint optimization and analysis of communication performance in a connected autonomous driving scenario.

Although much work has been done on IoV, traditional IoV ignores the driving characteristics and intelligence of CAVs, which leads to poor vehicle communication connectivity that cannot serve AD well. In network-assisted AD scenarios, CIoVs can better support AD scenarios by introducing the CE for cognition and collaboration in physical and cyberspace. Specifically, the cognition and clustering of the driving pattern can effectively improve the stability of the cluster, reduce frequent network switching and interruptions, and provide stable and reliable communication. In addition, effective collaboration between CAVs can greatly enhance the safety and efficiency of AD. However, it is very complex and challenging to solve the above problems.

In this paper, we propose an AD pattern recognition algorithm and design an intelligent clustering mechanism based on different AD patterns to provide stable and reliable AD communication services.

3. Network Architecture and Clustering Mechanism

In this section, we describe the proposed architecture and the intelligent clustering mechanism based on AD pattern recognition in detail. First, we present the proposed system architecture of the heterogeneous CIoV, the components that are included, and their functions in detail. We then describe the process of modelling AD patterns and build a training model which serves as the basis for pattern recognition. Finally, we detail the pattern recognition based clustering and CH selection mechanism.

3.1. CIoVs Architecture for AD

The proposed reliability-enhanced hierarchical CIoVs architecture is shown in Figure 1. The architecture mainly contained multiple CAVs with different AD modes, a 5G base station gNB, CE, and a cloud server (CS). Within the fog cell, the gNB collects Floating Car Data (FCD) within its coverage area. The CE processes the collected FCD to perform AD pattern recognition and clustering. Then, the CAVs are divided into different clusters according to AD modes. Among them, CAVs with the same or similar AD mode can form a cluster to maintain a relatively stable state. When the driving characteristics of CAVs are different from the proposed AD modes or a new driving mode appears, it can be switched to the same mode as the cluster member vehicles. Specifically, the constituents and functions are as follows.

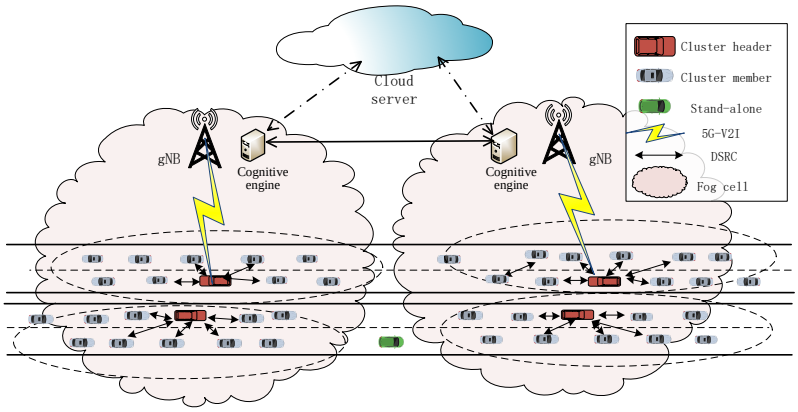


Figure 1. Heterogeneous CIoVs architecture.

(1) CAVs: CAVs are classified into CHs, CMs, and standalone (SA) vehicles, where SA vehicles refer to vehicles that are not clustered. Specifically, DSRC is an effective V2V communication technology, which is used to support the communication between vehicles in the cluster to provide realtime safety services. The SA CAVs communicate directly with the base station.

(2) gNB: The gNB is a centralized communication infrastructure that is responsible for communication with CHs and SA vehicles. For example, outside the cluster, the CH connects with the gNB through 5G-V2I technology, which can use the existing communication infrastructure to access the CS to obtain wider communication coverage and internet services.

(3) CE: The CE can perform AD mode recognition and control the switching of different modes based on context and location information to ensure cluster stability and driving efficiency. For example, when a new mode appears, it can be switched according to the prevailing situation and driving needs.

(4) CS: The cloud server can obtain delay-insensitive information from the gNB and provide global traffic information and services. We adopt two main communication technologies to complete different communication services, namely DSRC and 5G-V2I.

The CAVs set V within a fog cell is expressed as $V = \{V_1, V_2, V_3, \ldots, V_N\}$, where N is the total number of vehicles. In the fog cell, CAVs generate a large amount of FCD and networking data. Processing these data to obtain valuable information can better promote ITS and network-assisted AD. These CAVs are divided into multiple clusters $C = \{C_1, C_2, \ldots, C_L\}$ according to the excavated driving mode information. The vehicles in the fog cell have three different AD modes, and the set of driving modes can be expressed as $V_ID = \{M_1, M_2, M_3\}$, where M_1 represents conservative driving mode, M_2 represents moderate driving mode, and M_3 represents aggressive driving mode. Therefore, the category attribute of the vehicle can be expressed as $V_{i_v} \in V_ID$. The modeling of the AD mode will be introduced in the next subsection.

The gNB collects FCD and passes it to CE and the CE uses machine learning (ML) algorithms for data analysis to obtain the AD mode attributes of CAVs. The ML algorithm was used in this paper, which has strong data fitting and function mapping capabilities, and feature classification can be performed efficiently and reasonably [38]. Then, the CAVs with the same or similar driving patterns are gathered together to form a cluster through control instructions. Finally, each cluster selects the most reliable vehicle as the CH to connect with the gNB according to the stability of the AD mode, location, and communication status. This clustering mechanism can reduce the burden on the base station network and provide reliable communication [39]. Clustering is performed based on the result of AD pattern recognition, and we model the AD mode in the next subsection.

3.2. AD Mode Modeling

The AD modes of CAVs should be human-like and need to be adapted to the prevailing situation in order to achieve a comfortable and safe experience [40,41]. Based on the characteristics of driving in real urban traffic conditions, we classified the driving modes into three modes: aggressive, moderate, and conservative.These contain the main driving characteristics and can meet the needs of most driving scenarios. The CAVs with aggressive mode will have a relatively high-speed distribution compared to the CAVs with the other two modes and will overtake the CAVs of the other two modes. Conservative mode is the opposite of aggressive mode, i.e., lower average speed, very steady state of motion, and a tendency to follow. Finally, the moderate mode is between them.

Considering the driving behavior of the vehicle changes dynamically over time, we used VISSIM simulation software to collect the driving data of vehicles in different driving modes over time, which is presented in Section 6.1. Finally, through statistical analysis and preprocessing of these data, we obtained multidimensional data information closely related to the driving modes, i.e., feature vectors. The reflection in speed obeys truncated Gaussian distribution, the speed and overtaking frequency are mainly affected by the independent variables of acceleration and acceleration time. Therefore, based on the above analysis, several eigenvalues most relevant to the AD mode were selected as training data, including normalized velocity, acceleration time, overtaking frequency, etc. Through statistical analysis of the driving data of vehicles v_i in different modes, the preprocessed statistical feature value was used as the training feature tensor. The preprocessed statistical feature values were employed as training feature vectors. The feature matrix can be expressed as

$$F = \begin{bmatrix} A_{v_1} & B_{v_1} & C_{v_1} & D_{v_1} & E_{v_1} & F_{v_1} & \mathbf{T}_1 \\ A_{v_2} & B_{v_2} & C_{v_2} & D_{v_2} & E_{v_2} & F_{v_2} & \mathbf{T}_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ A_{v_i} & B_{v_i} & C_{v_i} & D_{v_i} & E_{v_i} & F_{v_i} & \mathbf{T}_3 \end{bmatrix}^T, \quad (1)$$

where $v_i \in V$ and the meaning of each letter parameter can be expressed as follows: A_{v_i} represents the mean of the velocity distribution of the CAV v_i, B_{v_i} represents the variance of the velocity distribution, C_{v_i} represents the acceleration mean, D_{v_i} represents the acceleration variance, E_{v_i} represents the acceleration time, F_{v_i} represents the overtaking frequency. **T** represents the vector of the output label. The GANN algorithm for AD pattern recognition is proposed in the next section, which utilizes these feature tensors as the training sets to identify the similarity of AD modes for intelligent clustering of CAVs and the clustering mechanism is introduced in the following subsection.

3.3. Clustering Mechanism

The model divided the AD mode into three categories, and the driving mode category label of each CAVs and its probability value in the category were obtained through the proposed AD pattern recognition algorithm. In the same category, the larger the probability value, the more stable the driving mode. It can be expressed as

$$V_ID = \begin{cases} M_1 = 1 \\ M_2 = 2, \\ M_3 = 3 \end{cases} \quad (2)$$

$$P(v_{i_v}) = [P(v_{i_v} \in V_ID)] = GANN\,(v_{i_v}). \quad (3)$$

Among them, V_ID represents the vehicle driving mode category and $P(v_{i_v})$ represents the output category vector of the proposed driving pattern recognition algorithm GANN, $i_v = 1, 2, \ldots, N$. It is a classification probability value, which satisfies $P(v_{i_v} \in M_1) + P(v_{i_v} \in M_2) + P(v_{i_v} \in M_3) = 1$. The vector $V_{i_v} = (V_ID, \max P(v_{i_v}))$ represents the driving mode category attribute of CAV i_v, where $\max P(v_{i_v})$ is the probability value

of vehicle i_v under the category V_ID, which reflects the stability of the driving mode. Therefore, the driving mode category indicator of the vehicle can be expressed as

$$S_{v_{i_v}} = V_ID + \max P(v_{i_v}), \tag{4}$$

where $S_{v_{i_v}}$ represents the quantitative value of the category attribute of the vehicle driving mode. The larger the value of $S_{v_{i_v}}$, the higher the stability of the vehicle in the category. If CAV i_v belongs to M_1 category; $S_{v_{i_v}} \in (1,2)$; if CAV i_v belongs to M_2 category, $S_{v_{i_v}} \in (2,3)$; if CAV i_v belongs to the M_3 category, $S_{v_{i_v}} \in (3,4)$.Then, according to the recognized vehicle mode, the vehicles with the same category are grouped in the same cluster. At the same time, in the same category, the larger the value of $S_{v_{i_v}}$, the higher the priority for being selected as the CH. Based on the proposed clustering mechanism, in the next subsection we will introduce the cluster header selection method.

3.4. Cluster Header Selection

CAVs with the same AD mode will not have frequent overtaking and switching behaviors because the CAVs in the cluster have similar speed distributions and have already met their own driving needs. In addition, to improve cluster stability and flexibility, the CAV can be switched to different AD modes based on traffic and road conditions. Appropriate CH can improve the reliability of communication inside and outside the cluster and avoid frequent switching due to CH changes. When clustering is performed, CAVs with the same V_ID value are grouped together to form a cluster based on position. The leader priority can be defined based on stability, channel quality, and distance from the centroid and it can be expressed as

$$L_{v_{i_v}}(t) = c_1|10(\max P(v_{i_v}) + \rho(t))| + c_2|D_{norm}| \quad t = 1, 2, \ldots, T, \tag{5}$$

where $\rho(t)$ represents the channel quality index, D_{norm} is the normalized distance of each vehicle from the cluster center, and c_1 and c_2 are the weights of each variable, $c_1 + c_2 = 1$. When the value of $\max P(v_{i_v})$ is larger, the driving state of the vehicle in this mode is more stable. It can be seen that when the vehicle with the largest $L_{v_{i_v}}(t)$ is selected as the CH, the subsequent ones will be used as candidate CH.

4. GANN for AD Pattern Recognition

In this section, we establish an AD pattern recognition model optimized by GA that takes full advantage of NN and GA.

4.1. AD Pattern Recognition

4.1.1. NN Model for AD Pattern Recognition

According to the driving model established above, the NN architecture for AD pattern recognition was established as follows:

(a) Number of layers in the NN: The NN consists of an input layer, an output layer, and multiple hidden layers. Theoretical studies have shown that a three-layer network with one hidden layer, which has sufficient neurons can achieve arbitrary nonlinear mapping [42]. We adopted a three-layer structure for pattern recognition.

(b) Number of neurons in the input layer: The input to the NN is the dimension of the feature vector associated with the driving mode. Here we chose six parameters that were closely correlated to our proposed AD mode as the input vector, i.e., the input layer had six neurons. Through the statistical analysis, the six parameters, respectively, correspond to $[A, B, C, D, E, F]$.

(c) Number of neurons in the output layer: The matrix of the target output column vectors corresponding to each category is $[T_1; T_2; T_3]$. Specially, T_1 stands for aggressive mode, which means that the vehicle accelerates frequently and travels at a high speed. T_3 stands for conservative mode, which means that the vehicle tends to actively brake to maintain a longer following distance and travels at a lower speed. T_2 stands for

moderate mode and refers to the transition zone between the above two modes, which usually maintains a steady speed with little fluctuation. The NN target output column vector T_1, T_2, T_3 are $[1,0,0]$, $[0,1,0]$, $[0,0,1]$, respectively. Therefore, the number of neurons in the output layer is three.

The driving pattern recognition model is a fully connected feedforward NN as shown in Figure 2. It works according to the principle that the number of neurons in the hidden layer is larger than that of the input layer and proportional. Considering the tradeoff between benefit and efficiency, the parameters and variables of the proposed GANN for AD pattern recognition are shown in Table 2.

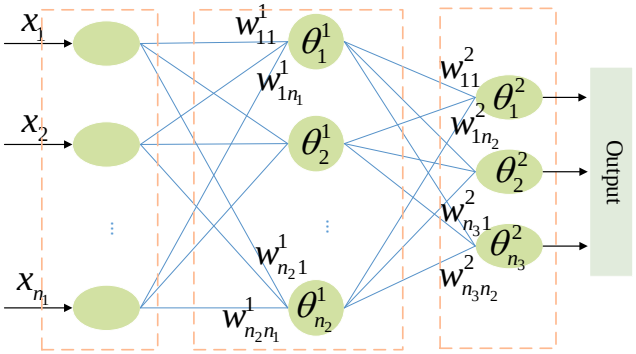


Figure 2. Neural network topology.

Table 2. Neural network variables and parameters.

Network Layer	Input Layer	Hidden Layer	Output Layer
Neuron Number	$n_1 = 6$	$n_2 = 42$	$n_3 = 3$
Transfer Function	*purelin*	*tansig*	*softmax*
Variables	n_1	n_2	n_3
Input	x_i	y_j	z_k
Output		O_j	Z_k
Weights	w_{ji}	w_{kj}	
Thresholds		θ_j	θ_k

4.1.2. Cost Function

According to the proposed network structure, the NN weights and thresholds to be optimized can be expressed as a matrix as follows

$$W = [w^1, \theta^1, w^2, \theta^2], \tag{6}$$

where, W is the total parameter to be optimized; w^1 is the weight between the input layer neurons and the hidden layer neurons; θ^1 is the threshold of the hidden layer neurons; w^2 is the weight between the hidden layer and the output layer neurons; and θ^2 is threshold of the output layer neurons.

The error of the actual output value and the target value of the network is taken as the objective function. The error between the expected value and the output value can be continuously reduced by using the gradient descent method. The error cost function of NN can be expressed as

$$E = \frac{1}{2m} \sum_{k=1}^{n_3} (Y_k - Z_k)^2, \tag{7}$$

where m represents the number of training samples, Y_k is the target value of the training sample, and Z_k is the actual output value of the NN. In the process of network forward propagation, the weights and thresholds were first initialized. Using vectorization opera-

tion to calculate the output of each layer, the output of the hidden layer neurons and the output of the output layer neurons are, respectively, expressed as

$$O_j = f(w^1 x_i - \theta_j^1) = f(net_j)\ i = 1, \ldots, n_1, \tag{8}$$

$$Z_k = g(w^2 O_j - \theta_k^2) = g(net_k)\ k = 1, \ldots, n_3. \tag{9}$$

where $j = 1, \ldots, n_2$ and $f(\cdot)$ and $g(\cdot)$ are the transfer functions of the hidden layer and the output layer, respectively. After two layers of forwarding transmission, the actual output value of the network can be obtained, and the network error can be obtained by substituting the above formula. The NN error function can be expressed as

$$E = \frac{1}{2m} \sum_{k=1}^{n_3} \left\{ Y_k - g\left[\sum_{j=1}^{n_2} w_{kj}^2 f(\sum_{i=1}^{n_1} w_{ji}^1 x_i - \theta_j^1) - \theta_k^2 \right] \right\}^2. \tag{10}$$

This shows that the NN error is a function of the parameters with respect to the W. Therefore, the pattern recognition error can be reduced by continuously adjusting the W of the NN.

4.1.3. Parameter Update

Using the backpropagation algorithm, the W updates along the direction of the gradient descent of the error function until the error accuracy requirement is met or the maximum number of iterations is reached. The update process of weights and thresholds can be expressed as follows

$$\begin{cases} w(t+1) = w(t) - \Upsilon \frac{\partial E}{\partial w} \\ \theta(t+1) = \theta(t) + \Upsilon \frac{\partial E}{\partial \theta} \end{cases}, \tag{11}$$

where Υ is the learning rate, i.e., the adjustment step size of the weights and thresholds. The partial derivative of the model error function can be calculated using the chain derivation rule. However, the initial W has a large impact on the performance of the NN model, which will result in unstable classification results. In order to obtain the global optimal solution, the initial W needs to be optimized.

4.2. GANN Algorithm for AD Pattern Recognition

The optimization process includes the NN training phase and GA optimization phase. In this subsection, we use the GA algorithm to optimize the initial W. First, the W to be optimized was encoded as the chromosome of individuals, and the NN error norm was used to construct the fitness function. Then, we calculated the fitness value of individuals in the population, and used the operations of selection, crossover, and mutation to generate the next-generation and find the optimal chromosome. The specific steps of the GANN algorithm are as follows:

(1) Population initialization: The population contains M individuals, each of which represents all the parameters to be optimized. Each individual can be coded as

$$Ind = \left\{ w^1, \theta^1, w^2, \theta^2 \right\}, \tag{12}$$

where Ind represents an individual in the population and M is the number of individuals included in the population during initialization. That is, all weights and thresholds are encoded as a chromosome of an individual, which is a row vector. Then a population of size M can be expressed as

$$X = [Ind_1; Ind_2; \ldots; Ind_M], \tag{13}$$

where X represents the initial population of the GA. Len represents the chromosome length of each individual. During the initial coding of the individuals, each parameter is coded in a standard binary code. That is, all the parameters to be optimized for each individual are encoded as a $10 - bit$ binary number, so the length of the chromosome of each individual can be represented as $L_{en} = 10 \times (252 + 42 + 126 + 3) = 4230$.

(2) Fitness function determination: The fitness function is established based on the objective function. According to the principle of the survival of the fittest by GA, the smaller the error of the objective function, the greater the fitness value. Based on the above analysis, the fitness function can be expressed as

$$Fit(X) = \frac{2}{E}, \tag{14}$$

where X represents an individual in the population and E represents the test sample error produced by the individual.

(3) Selection operation: The GA will preferentially select individuals with higher fitness values for inheritance, and the superior genes of the previous generation will be preserved. The probability that a parent individual is selected is

$$P(X_{Ind}) = \frac{Fit_m}{\sum_{m=1}^{M} Fit_m}. \tag{15}$$

The probability of selection is constructed according to the individual fitness value, i.e., individuals with high fitness values are more likely to pass on the genes to the next generation.

(4) Crossover operatio: Using a single point crossover method, each parental individual produces a progeny individual through chromosome crossover. Assuming that the parent individuals are $C_u(gen)$ and $C_v(gen)$, respectively, the progeny individuals generated after the crossover are

$$C_v(gen+1) = \begin{cases} C_v(gen, q) & q < r_c \\ C_u(gen, q) & q \geq r_c \end{cases}, \tag{16}$$

$$C_u(gen+1) = \begin{cases} C_u(gen, q) & q < r_c \\ C_v(gen, q) & q \geq r_c \end{cases}. \tag{17}$$

That is, the parent forms new offspring by exchanging chromosomes, where $r_c \in [1, L_{en}]$ is the crossover coding lengths of chromosomes.

(5) Mutation operation: In order to maintain the global search characteristics of GA and avoid the convergence to the local optimum quickly, the individuals must maintain a certain probability of variation. Mutation produces the number of variant genes with a certain probability, and randomly selects the gene for mutation. If the selected gene is coded as 1, it becomes 0; otherwise, it becomes 1. Variations maintain population diversity to prevent immature convergence.

(6) Termination condition: We set G as the maximum number of genetic generations of the GA. The optimization algorithm terminates when the GA reaches the maximum genetic generation or minimum target error. Then, the individual that has the smallest error in the genetic evolution process is the optimal solution. The output of the final training result is the optimal solution.

4.3. GANN for AD Pattern Recognition Algorithm

Algorithm 1 presents the process of GANN for pattern recognition, which overcomes the defect that NN easily falls into the local optimal during the learning process.

Algorithm 1 GANN for AD pattern recognition

Input: Initial population X, Training samples F, Initialization parameters.
Output: Optimal network parameters Ind, minimum error E.

```
NN structure determination.
Initialize model parameters {n1, n2, n3, G, M, ··· , X}.
X = [Ind1; Ind2; ··· ; IndM].
Convert X to decimal, calculate the objective function value E.
gen = 0.
while gen ≤ G do
    Training of NN;
    Calculate Fit(X);
    Calculate P(X), then selection, crossover, and mutation operation get X'(Ind);
    Convert X'(Ind) to decimal, compute E(X');
    X(gen + 1) = X ← X';
    Convert X(gen + 1) → decimal;
    gen = gen + 1;
    Update trace ← best error E, optimal X;
end while
```

The main idea was to use the test error obtained through NN training as the optimization objective function of GA. We will introduce the NN training part in the next section. The NN model is a multi-classification problem. The *Softmax* function has good multi-classification properties, it is a probability selection mechanism. At the output layer of NN, we used *Softmax* as the transfer function. It can be expressed as

$$p(z_k) = \frac{e^{z_k}}{\sum_{k=1}^{n_3} e^{z_k}}. \tag{18}$$

The sum of the probabilities of the output of the three categories is 1, and the class with the highest probability can be output as the mode category to which the CAV belongs. A larger output value indicates a higher accuracy of the classification, where the NN training process for AD to obtain the error of each iteration is the objective function of GA optimization. The training of NN is to minimize network output errors in each training process by updating the weights and thresholds.

5. Performance Analysis

In this section, we introduce the stability and lifetime of the cluster and analyze the communication performance of the proposed clustering mechanism.

5.1. Cluster Stability and Lifetime Analysis

Considering the CAVs in motion is a time-varying process, each period T is divided into several equal time slots τ. Each time slot is used to collect the beacon information of the vehicle, which can be obtained from the electronic control unit (ECU). Therefore, the vehicle position information $V_{i_v}(t) = [x_{i_v}, y_{i_v}]$ and other driving parameters can be used for cluster stability calculation and driving mode classification. For example, the vehicle speed and the distance between vehicles can be effectively calculated based on the location information of the vehicle at a certain moment.

The relative distance between CH_{j_v} and CM_{i_v} can be obtained through the Euclidean formula using position information. It can be characterized as

$$D_{i_v,j_v}(t) = \sqrt{(x_{i_v} - x_{j_v})^2 + (y_{i_v} - y_{j_v})^2}. \tag{19}$$

The average relative distance between CM_{i_v} and CH_{j_v} in a cluster can be expressed as $D_v(t) = \frac{1}{|c|}\sum_{c}(V_{i_v}(t) - V_{j_v}(t))$. Therefore, the sequence of the average distance variable between CM_{i_v} and CH_{j_v} changing with time is $\{D_v(1), D_v(2), \cdots, D_v(t)\}$, which can be obtained using the Euclidean distance. We can obtain $\Delta D_V(t) = D_v(t+1) - D_v(t)$

by performing the difference operation on the average relative distance $D_v(t)$, then the stability index of the cluster can be derived by calculating the variance of the difference variable $\Delta D_V(t)$. The stability index of the cluster can be expressed as

$$S_{ta} = Var(\Delta D_V(t)). \tag{20}$$

When the CM of the cluster changes, the average distance between CH and CM will change, even the CAV will escape from the current cluster which may cause handover. At the same time, to maintain effective communication within the cluster, the maximum distance between CM and CH should be less than the coverage of the DSRC, i.e., $\max D(V_{CH}, V_{CM}) < range(DSRC)$. If S_{ta} remains a very small value, the time duration τ can represent the lifetime of the cluster. The lifetime of cluster can be expressed as

$$T^c_{lifetime} = Duration(\tau) \;\; s.t. \;\; S_{ta} < \delta. \tag{21}$$

The CE performs AD mode recognition based on the collected FCD of CAVs. Based on the location information, the CAVs with the same or similar AD mode are formed into a cluster and generate an info table of the cluster. When the number of CMs changes greatly, it will cause the average relative distance to change greatly, and the lifetime of the cluster may decrease. Therefore, the smaller the S_{ta}, the more stable the cluster.

5.2. Communication Performance

The proposed architecture contains multiple clusters, the CH set can be expressed as $CH = \{CH_1, CH_2, \cdots, CH_L\}$, and the CM set of each cluster can be expressed as $CM = \{CM_1, CM_2, \cdots, CM_K\}$. In the analysis of reliable traffic, it is necessary to consider two communication models, within the cluster and outside the cluster. The sum of the reliable traffic of the CAVs is the reliable throughput of the fog cells. Therefore, the V2V reliable traffic between the CM and CH in the cluster can be expressed as

$$C^{V2V}_{i_v} = \eta_0 W_0 R(t) \quad i_v \in C, \tag{22}$$

$$R(t) = \log_2(1 + L_{v_{i_v}}(t)/S_{ta}), \tag{23}$$

where η_0 is the spectrum efficiency, W_0 is the CM bandwidth, and $R(t)$ represents the reliable transmission index.

We assume that the channel quality index $\rho(t)$ is constant within a time period T, it can be detected by the vehicle sensing unit. Through the above CH selection mechanism, the CH can stably and reliably collect vehicle data and upload it to the base station. The V2I reliable traffic between the CH and the infrastructure can be expressed as

$$C^{V2I}_{ch} = \rho(t)\eta_l W_{ch}, \tag{24}$$

where η_l is the spectrum efficiency and W_{ch} is the bandwidth allocated to CH for V2I communication.

Thus, the reliable throughput of the proposed architecture can be expressed as

$$C^L_{reliable} = C^{V2V}_{i_v} + C^{V2I}_{ch} \quad i_v \in V \; ch \in C, \tag{25}$$

substituting Equations (9)–(11) into the Equation (12) can obtain

$$C^L_{reliable} = \left(\eta_0 W_0 \log_2(1 + L_{v_{i_v}}(t)/S_{ta}) + \rho(t)\eta_l W_{ch}\right) \quad v_{i_v} \in V \; ch \in CH. \tag{26}$$

It can be seen that in order to increase reliable throughput, the stability of the cluster needs to be improved. The improvement of cluster stability can effectively avoid the migration of CAVs between different clusters, thus, the handover and interruption can be avoided and improve cluster lifetime. Based on the proposed AD mode clustering and CH selection

mechanism, CAVs can be automatically clustered, which can effectively enhance the cluster stability and reduce handover.

6. Simulation Results and Discussions

In this section, simulations and discussions are carried out. We mainly discuss the accuracy of the AD pattern recognition and the reliability of cluster communication to evaluate our proposed architecture and clustering mechanism. The CAVs were equipped with communication equipment and had large numbers of onboard sensors as data sources, such as acceleration, speed, brakes, LiDAR, and other sensors [43].

6.1. Training and Test Dataset

In order to verify the rationality of the proposed three AD modes, the traffic simulation software VISSIM was used to collect the sample data. Three experimental vehicle models were built in VISSIM, corresponding to the three driving modes in Section 3.2. By setting the operating parameters of the experimental vehicle, including the desired speed distribution, maximum acceleration function, and desired acceleration function, we established the distribution of three vehicle categories corresponding to the three driving modes. Finally, the data obtained were analyzed to obtain the driving characteristics of the vehicles in the different modes. The settings of the simulation parameters in VISSIM are shown in Table 3. A snapshot of the simulation running environment is shown in Figure 3.

Table 3. Simulation configuration in VISSIM.

Simulation Parameter	Value
Road length	1000 m
Number of lanes	3
Road type	Urban motorway
Car following model	Wiedemann 99
Traffic flow	3000
Vehicle mode	3
Simulation period	360 s



Figure 3. One snapshot of the simulation environment.

Through statistical analysis of the dynamic driving data collected during the VISSIM simulation period, the distribution of these data was obtained. These data belonged to a truncated Gaussian distribution. The mean and variance were obtained by preprocessing. For simulation convenience, corresponding feature vector matrices were simulated in MATLAB according to distribution features. A portion of the normalized training samples

utilized for GANN learning optimization are listed in Table 4. In reality, FCD can be transmitted to the CE via CIoVs and processed in CE, or preprocessed locally.

Table 4. Normalized feature factors of examples.

Num / Fac	1	2	3	4	5	6	7	8
A	0.1280	0.0815	0.1104	0.6400	0.5059	0.4983	0.9466	0.8753
B	0.1641	0.6104	0.4600	0.3347	0.3403	0.3024	0.6179	0.4628
C	0.2485	0.0394	0.0001	0.4362	0.3578	0.1700	0.5369	0.8277
D	0.5683	0.1800	0.6096	0.5493	0.7245	0.4920	0.4391	0.7612
E	0.0580	0.0692	0.0350	0.4408	0.4151	0.4485	0.9228	0.9372
F	0.2076	0.0963	0.1011	0.4100	0.3530	0.3778	0.8139	0.8721
Cons	1	1	1	0	0	0	0	0
Mod	0	0	0	1	1	1	0	0
Aggr	0	0	0	0	0	0	1	1

6.2. Performance Analysis of AD Pattern Recognition

The proposed algorithm was trained and validated using the dataset collected by the above method and was compared with the NN without genetic optimization, the deep neural network (DNN) with four layers, and the support vector machine (SVM). The settings of the simulation parameters of the GANN optimization algorithm model are shown in Table 5, which contains the GA part and the NN part. The number of generations and populations must be integers and the values of parameter regarding the probability were decimals between 0 and 1.

Table 5. Simulation parameters.

Simulation Parameter	Value
Population size M	40
Genetic generation G	50
Cross probability px	0.7
Mutation probability pm	0.01
Generation gap $GGAP$	0.95
Training goal Tg	0.01
Learning rate Lr	0.1
DSRC Channel bandwidth [44]	10 MHz
5G V2I Channel bandwidth [45]	20 MHz
Total bandwidth	300 MHz

Figure 4 shows the evolution of the GANN test errors, which was an important indicator of the generalization ability and accuracy of the proposed GANN pattern recognition algorithm. The test error of the test sample was an important indicator of the generalization ability of the GANN. It can be seen from the figure that the error decreased with the increase in genetic generation. The smaller the test error, the stronger the generalization ability of the network for new samples. The error decreased from above 0.35 to below 0.02 and the minimum test error was 0.014. The minimum value of 0.014 was obtained at 47 generations.

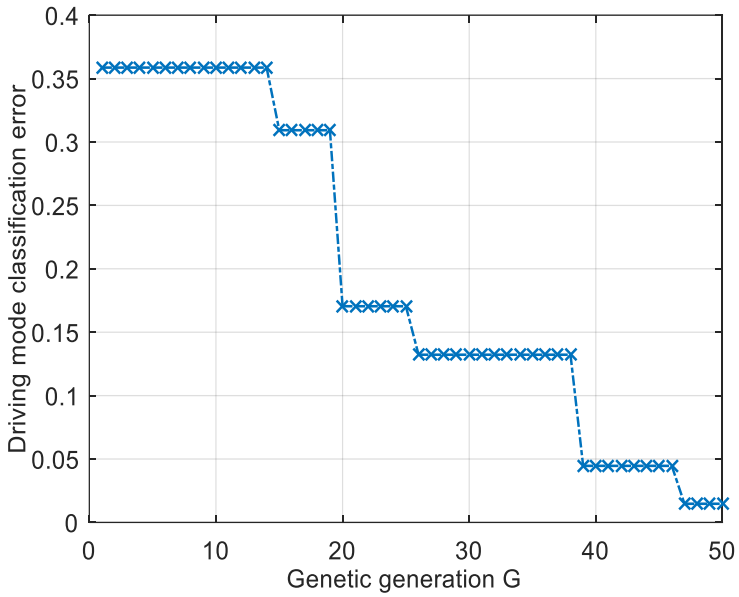


Figure 4. GANN error evolution curve for AD pattern recognition.

The number of experiments took into account the balance between cost and effectiveness to verify the reasonableness of the maximum number of genetic generations. It can be seen from Figure 5 that after 50 iterations for each experiment, a low minimum error was achieved. High recognition accuracy was already achieved under this error condition. Therefore, the maximum number of iterations was chosen to be 50 for the tradeoff between efficiency and accuracy. Figure 6 shows the variation curves of the AD pattern recognition errors of GANN and NN. Since the initialization weights and thresholds of each NN were random, the errors converged to different local optimal solutions, and the pattern recognition error fluctuated each time. The GANN for the AD pattern recognition algorithm, which optimized the training error of the NN and continuously reduced the recognition error with genetic iteration, obtained more accurate recognition.

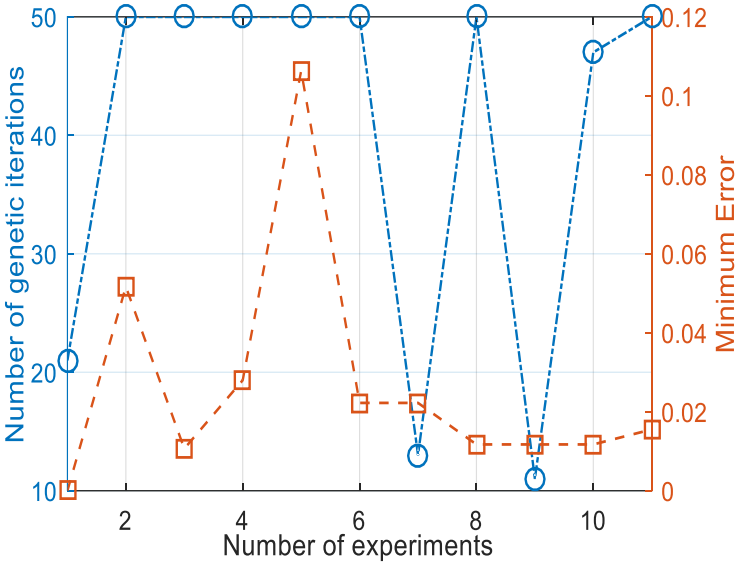


Figure 5. GANN experiment and minimum error.

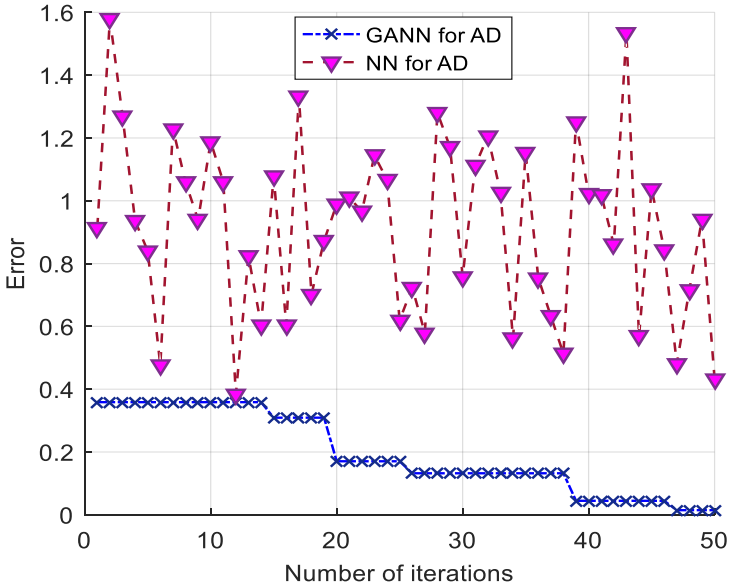


Figure 6. GANN and NN model error iterative evolution.

Figure 7 shows the comparison of GANN with other ML algorithms for conservative driving pattern recognition. The closer the output value of the pattern recognition algorithm to 1, the higher the recognition accuracy. The output value of the GANN for the AD pattern recognition algorithm was almost equal to 1 and remained stable, while the output values of other algorithms fluctuated significantly. This is because GANN can effectively optimize the initial weights and threshold vectors to obtain the global optimal solution, thus reducing the recognition error. Therefore, the GANN algorithm had higher recognition accuracy and stability than the other algorithms.

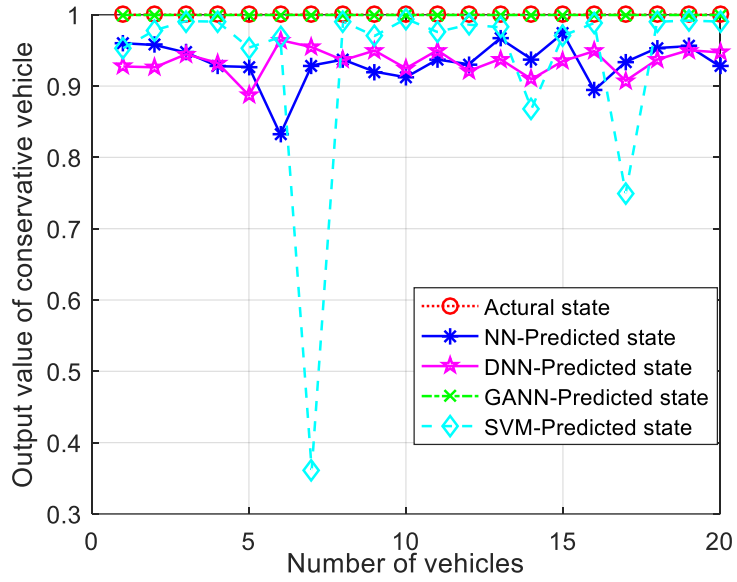


Figure 7. Comparison of output values for conservative CAVs.

Similarly, to verify the generality of the model, the output comparison simulations for the other two drive models in real life are shown in Figures 8 and 9. The NN and DNN models had a large and dramatic fluctuation between the predicted and actual values. The SVM performed very erratically, due to the jitter of the vehicle driving pattern. However, the predicted values of the GANN model were very close to the actual values,

and the fluctuations of the output values were very small. The proposed GANN for AD pattern recognition algorithm had better recognition stability and generalization and was able to perform pattern recognition with high accuracy, which led to more stable clustering.

The test error and training error of different algorithms are shown in Table 6. The proposed GANN for AD pattern recognition algorithm had the smallest error, which could effectively avoid the wrong decisions caused by misjudgments and maintain the stability of the AD clustering system.

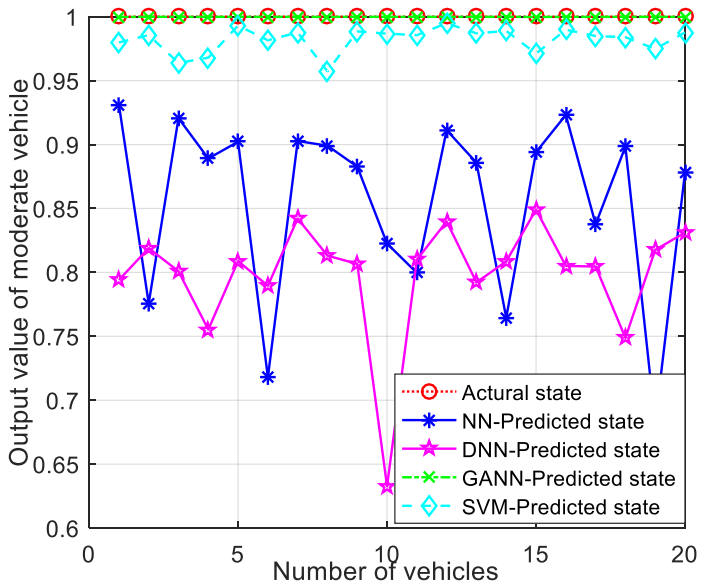


Figure 8. Comparison of output values for moderate CAVs.

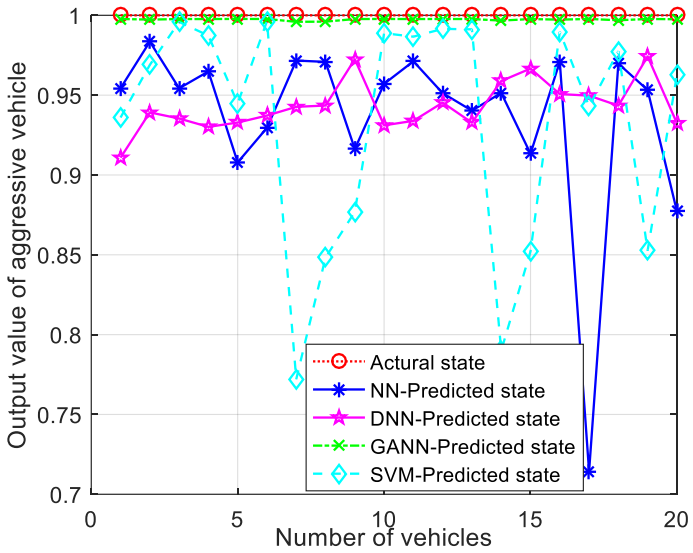


Figure 9. Comparison of output values for aggressive CAVs.

Table 6. Pattern recognition errors.

ML	Test Error	Training Error
NN	1.0568	1.7353
GANN	0.014711	0.023431
DNN	1.1882	1.9773
SVM	0.88613	0.17789

6.3. The Effectiveness of the Proposed Clustering Mechanism

Effective driving pattern recognition and classification to form clusters were the basic components of our proposed architecture, which are very important for reliable communication. To verify the effectiveness of the proposed clustering mechanism based on the AD mode (ADMode-cluster), we adopted six schemes for comparison as follows:

1. The non-cluster scheme: The scheme was not clustered, and each vehicle on the road only communicated with the new base station through the allocated 5G-V2I technology [25];
2. The section-cluster scheme: The scheme divided the road into sections of equal length, and vehicles on the same road section formed a cluster [26]. Then the V2V communication within-cluster and V2I between CH and gNB were performed, respectively;
3. The ADMode-cluster scheme: The CE used the proposed pattern recognition algorithm to gather vehicles with the same driving pattern into a cluster. On the basis of the proposed clustering mechanism, a heterogeneous hierarchical network architecture was used to carry out multimode communication to complete AD services;
4. The Mixmode-cluster scheme: At the same time, the vehicles in the cluster had different driving modes, i.e., the CAVs in the cluster had three driving modes;
5. The ADMix-cluster scheme: CAVs with different driving modes on the same road at the same time were divided into different subclusters based on the driving modes, i.e., the driving modes in each cluster were the same;
6. The Kmeans-cluster scheme: It performed an iterative unsupervised clustering algorithm based on the position of the CAVs, thus clustering the vehicles into different clusters [46].

The communication simulation was performed considering a two-parallel three-lane (2×3 lanes) urban road. The total bandwidth was 300 MHz, of which 200 MHz was utilized for 5G V2I communication, and the remaining 100 Mhz was exploited for DSRC V2V communication. The spectral efficiency of the DSRC was defined as 10 bps/Hz, the spectral efficiency of the 5G V2I was 15 bps/Hz, and the communication range of the DSRC was 200 m. The other simulation parameters are shown in Table 5.

Figure 10 shows the comparison of the average reliable traffic for the six schemes under different vehicle density conditions. The proposed scheme based on the ADMode-cluster was better than the other five schemes. This is due to the fact that the proposed scheme selects CAVs with the best channel quality and stability as CH. In addition, clusters with the same or similar AD mode are able to maintain stable driving, avoid frequent handover, and interruptions of vehicle communication. Since the ADMix-cluster scheme was divided into different subclusters on the same road, it had better performance in terms of stability. However, too many clusters can cause coverage space overlap and reduce channel utilization. Therefore, the performance was slightly lower than that of the ADMode-cluster scheme. The Kmeans-cluster scheme achieved good results because it performed effective clustering based on location. The non-cluster scheme had the lowest reliable traffic due to the very unstable network topology and only 5G V2I communication mode. The Mixmode-cluster scheme decreased its reliable traffic due to the frequent migration and switching of CAVs with different driving modes within the cluster. As the vehicle density increased, the performance of the Section-cluster was higher than that of the non-cluster scheme but lower than that of the Kmeans-cluster scheme. This is mainly due to the fact that although the section-cluster scheme can use DSRC bandwidth by clustering, this fixed section division scheme cannot perform effective clustering to solve the problem of random movement.

Figure 11 shows the reliable throughputs of the six schemes with different vehicle densities. The throughput of the proposed ADMode-cluster scheme was significantly higher than the other schemes. This is due to the consistent driving mode of the CAVs in the cluster, thus avoiding the frequent replacement of CAVs. Similarly, the ADMix-cluster scheme also achieved high reliable throughput. The section-cluster scheme is a fixed clustering method that divided CAVs into different clusters based on fixed road

sections. In this way, the CAVs in the cluster were likely to move out and replace frequently, which would lead to frequent handovers. This phenomenon becomes more pronounced when the vehicle density increases, which will lead to a decrease in reliable throughput. As vehicle density increases, the reliable throughput tends to increase and then decrease. This is mainly because the increase of cluster members results in better spectrum gain but above a certain level will cause an increase in collision probability and overload, thus affecting performance. The non-cluster scheme had the smallest reliable throughput, and the variation with vehicle density was also small since no clustering communication was performed.

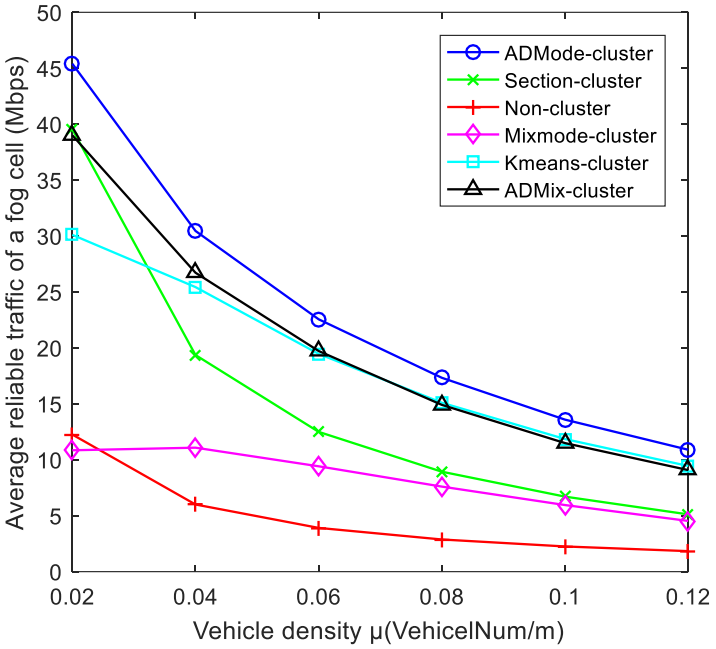


Figure 10. Comparison of average reliable traffic under different vehicle densities.

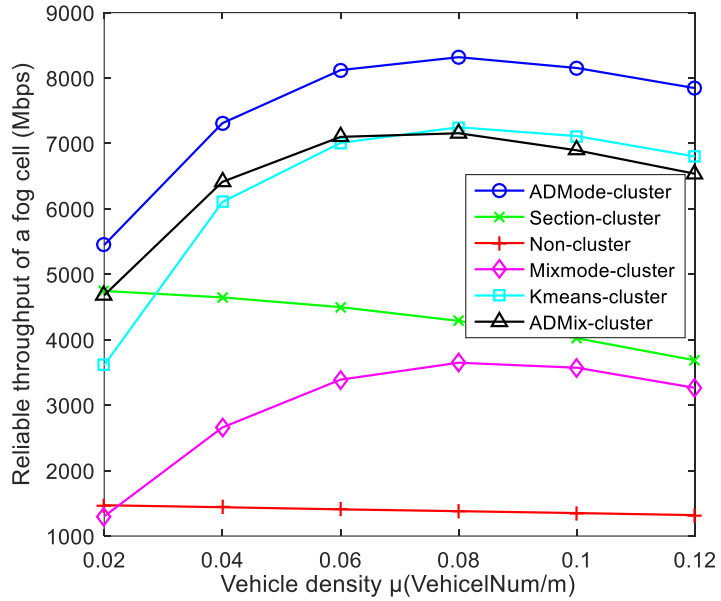


Figure 11. Comparison of reliable throughput under different vehicle densities.

Figure 12 shows the average lifetime of different schemes. The ADMode-cluster scheme and ADMix-cluster scheme had the highest average lifetime of clusters. The reason is that the clusters can be maintained for a long time as the CAVs move in the same

driving mode and the driving stability is higher than the hybrid model. This means that CAVs can maintain connections within a cluster for a long time while avoiding frequent handovers. The Kmeans-cluster scheme also showed a good cluster lifetime, since clusters can be kept relatively stable by continuous iterative clustering. Figure 12 shows that the average lifetime of clusters for all schemes tended to increase as the CAVs density increased. In addition, the Mixmode-cluster scheme and Section-cluster scheme were greatly affected by vehicle density. This is because when the vehicle density is higher, there will be more members in the cluster. Therefore, the CAVs are less likely to switch their AD mode, because it becomes difficult to change lanes.

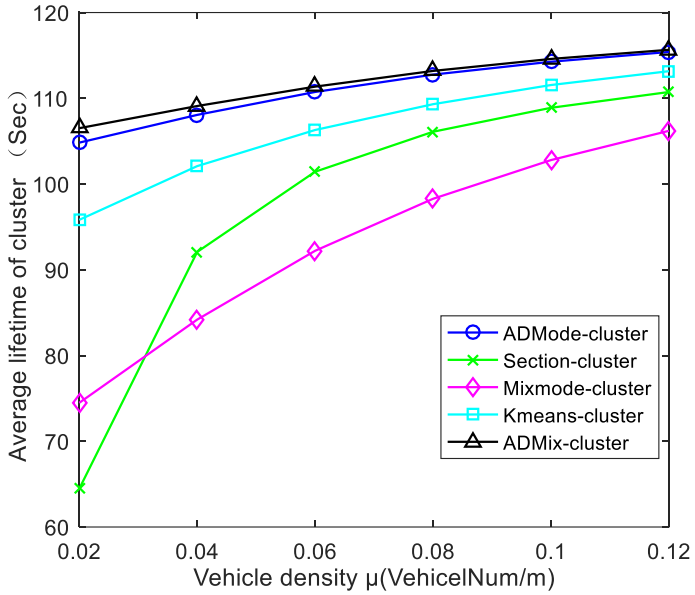


Figure 12. Comparison of average lifetime of cluster under different vehicle densities.

7. Conclusions and Future Work

In this paper, we proposed an intelligent clustering mechanism in heterogeneous CIoVs which orchestrated the CAVs into different clusters based on AD mode to enhance the reliability of vehicular communication. In the proposed heterogeneous clustered CIoVs architecture, multiple communication technologies were exploited to improve connectivity and coverage area. We established an AD mode model and proposed a GANN-based AD pattern recognition algorithm to improve the accuracy of the clustering mechanism. Simulation results showed that our approach outperformed the state-of-the-art methods in terms of pattern recognition accuracy, reliable throughput, and cluster lifetime. In future work, we will investigate the blockchain-enabled CIoVs to establish a secure and trustworthy intelligence sharing and incentive platform to facilitate the collaboration of CAVs.

Author Contributions: Conceptualization, H.C. and N.N.; methodology, H.C.; software, H.C.; formal analysis, H.C.; Validation, H.C.; data curation, H.C.; investigation, H.C.; writing—original draft preparation, H.C.; writing—review and editing, H.C. and N.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Key Area R&D Program of Guangdong Province under Grant No. 2018B030338001 and the National Natural Science Foundation of China under Grant No. 62001050.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request.

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

Abbreviations

CIoVs	Cognitive Internet of Vehicles
ITS	Intelligent Transportation System
AD	Autonomous Driving
CAVs	Connected Autonomous Vehicles
NLOS	Non Line of Sight
IoV	Internet of Vehicles
CE	Cognitive Engine
AI	Artificial Intelligence
CH	Cluster Head
DSRC	Dedicated Short Range Communication
V2I	Vehicle to Infrastructure
GA	Genetic Algorithm
NN	Neural Networks
SDN	Software Defined Network
RSU	Roadside Unit
CR	Cognitive Radio
V2V	Vehicle to Vehicle
CM	Cluster Member
gNB	5G Base Station
CS	Cloud Server
FCD	Floating Car Data
ML	Machine Learning
ECU	Electronic Control Unit
DNN	Deep Neural Network
SVM	Support Vector Machine

References

1. Yu, C.; Lin, B.; Guo, P.; Zhang, W.; Li, S.; He, R. Deployment and dimensioning of fog computing-based internet of vehicle infrastructure for autonomous driving. *IEEE Internet Things J.* **2018**, *6*, 149–160. [CrossRef]
2. Fagnant, D.J.; Kockelman, K. Preparing a nation for autonomous vehicles: Opportunities, barriers and policy recommendations. *Transp. Res. Part A Policy Pract.* **2015**, *77*, 167–181. [CrossRef]
3. Saini, M.; Alelaiwi, A.; Saddik, A.E. How close are we to realizing a pragmatic VANET solution? A meta-survey. *ACM Comput. Surv. (CSUR)* **2015**, *48*, 1–40. [CrossRef]
4. Siddiqa, A.; Shah, M.A.; Khattak, H.A.; Akhunzada, A.; Ali, I.; Razak, Z.B.; Gani, A. Social internet of vehicles: Complexity, adaptivity, issues and beyond. *IEEE Access* **2018**, *6*, 62089–62106. [CrossRef]
5. Ang, L.M.; Seng, K.P.; Ijemaru, G.K.; Zungeru, A.M. Deployment of IoV for smart cities: Applications, architecture, and challenges. *IEEE Access* **2018**, *7*, 6473–6492. [CrossRef]
6. Kaiwartya, O.; Abdullah, A.H.; Cao, Y.; Altameem, A.; Prasad, M.; Lin, C.T.; Liu, X. Internet of vehicles: Motivation, layered architecture, network model, challenges, and future aspects. *IEEE Access* **2016**, *4*, 5356–5373. [CrossRef]
7. Yang, F.; Wang, S.; Li, J.; Liu, Z.; Sun, Q. An overview of internet of vehicles. *China Commun.* **2014**, *11*, 1–15. [CrossRef]
8. Chen, M.; Tian, Y.; Fortino, G.; Zhang, J.; Humar, I. Cognitive internet of vehicles. *Comput. Commun.* **2018**, *120*, 58–70. [CrossRef]
9. Lu, H.; Liu, Q.; Tian, D.; Li, Y.; Kim, H.; Serikawa, S. The cognitive internet of vehicles for autonomous driving. *IEEE Netw.* **2019**, *33*, 65–73. [CrossRef]
10. Xiao, Y.; Ren, D. A hierarchical decision architecture for network-assisted automatic driving. In Proceedings of the 2018 IEEE International Conference on Energy Internet (ICEI), Beijing, China, 21–25 May 2018; pp. 35–37.
11. Yuan, Q.; Zhou, H.; Li, J.; Liu, Z.; Yang, F.; Shen, X.S. Toward efficient content delivery for automated driving services: An edge computing solution. *IEEE Netw.* **2018**, *32*, 80–86. [CrossRef]
12. Kuutti, S.; Fallah, S.; Katsaros, K.; Dianati, M.; Mccullough, F.; Mouzakitis, A. A survey of the state-of-the-art localization techniques and their potentials for autonomous vehicle applications. *IEEE Internet Things J.* **2018**, *5*, 829–846. [CrossRef]
13. Kim, S.W.; Ko, K.; Ko, H.; Leung, V.C. Edge-Network-Assisted Real-Time Object Detection Framework for Autonomous Driving. *IEEE Netw.* **2021**, *35*, 177–183. [CrossRef]
14. Wang, J.; Liu, J.; Kato, N. Networking and communications in autonomous driving: A survey. *IEEE Commun. Surv. Tutor.* **2018**, *21*, 1243–1274. [CrossRef]
15. Won, M. L-Platooning: A Protocol for Managing a Long Platoon With DSRC. *IEEE Trans. Intell. Transp. Syst.* **2021**. [CrossRef]

16. Chen, S.; Hu, J.; Shi, Y.; Zhao, L.; Li, W. A vision of C-V2X: Technologies, field testing, and challenges with chinese development. *IEEE Internet Things J.* **2020**, *7*, 3872–3881. [CrossRef]
17. Molina-Masegosa, R.; Gozalvez, J.; Sepulcre, M. Comparison of IEEE 802.11 p and LTE-V2X: An evaluation with periodic and aperiodic messages of constant and variable size. *IEEE Access* **2020**, *8*, 121526–121548. [CrossRef]
18. Klapez, M.; Grazia, C.A.; Casoni, M. Experimental Evaluation of IEEE 802.11 p in High-Speed Trials for Safety-Related Applications. *IEEE Trans. Veh. Technol.* **2021**. [CrossRef]
19. Lin, C.; Han, G.; Qi, X.; Guizani, M.; Shu, L. A distributed mobile fog computing scheme for mobile delay-sensitive applications in SDN-enabled vehicular networks. *IEEE Trans. Veh. Technol.* **2020**, *69*, 5481–5493. [CrossRef]
20. Zhang, K.; Leng, S.; Peng, X.; Pan, L.; Maharjan, S.; Zhang, Y. Artificial intelligence inspired transmission scheduling in cognitive vehicular communications and networks. *IEEE Internet Things J.* **2018**, *6*, 1987–1997. [CrossRef]
21. Kim, B.; Kim, S.; Yoon, H.; Hwang, S.; Punithan, M.X.; Jo, B.R.; Choi, S. Nearest-first: Efficient relaying scheme in heterogeneous V2V communication environments. *IEEE Access* **2019**, *7*, 23615–23627. [CrossRef]
22. Ni, Y.; Cai, L.; He, J.; Vinel, A.; Li, Y.; Mosavat-Jahromi, H.; Pan, J. Toward reliable and scalable internet of vehicles: Performance analysis and resource management. *Proc. IEEE* **2019**, *108*, 324–340. [CrossRef]
23. Zheng, K.; Zheng, Q.; Chatzimisios, P.; Xiang, W.; Zhou, Y. Heterogeneous vehicular networking: A survey on architecture, challenges, and solutions. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 2377–2396. [CrossRef]
24. Qi, W.; Landfeldt, B.; Song, Q.; Guo, L.; Jamalipour, A. Traffic differentiated clustering routing in DSRC and C-V2X hybrid vehicular networks. *IEEE Trans. Veh. Technol.* **2020**, *69*, 7723–7734. [CrossRef]
25. Wang, T.; Cao, X.; Wang, S. Self-adaptive clustering and load-bandwidth management for uplink enhancement in heterogeneous vehicular networks. *IEEE Internet Things J.* **2019**, *6*, 5607–5617. [CrossRef]
26. Zheng, Q.; Zheng, K.; Sun, L.; Leung, V.C. Dynamic performance analysis of uplink transmission in cluster-based heterogeneous vehicular networks. *IEEE Trans. Veh. Technol.* **2015**, *64*, 5584–5595. [CrossRef]
27. Ling, M.H.; Yau, K.L.A.; Qadir, J.; Ni, Q. A reinforcement learning-based trust model for cluster size adjustment scheme in distributed cognitive radio networks. *IEEE Trans. Cogn. Commun. Netw.* **2018**, *5*, 28–43. [CrossRef]
28. Liu, Y.; Chen, X.; Chen, C.; Guan, X. Traffic big data analysis supporting vehicular network access recommendation. In Proceedings of the 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia, 22–27 May 2016; pp. 1–6.
29. Li, Z.; Wang, C.; Jiang, C.J. User association for load balancing in vehicular networks: An online reinforcement learning approach. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 2217–2228. [CrossRef]
30. Wang, W.; Ramesh, A.; Zhu, J.; Li, J.; Zhao, D. Clustering of driving encounter scenarios using connected vehicle trajectories. *IEEE Trans. Intell. Veh.* **2020**, *5*, 485–496. [CrossRef]
31. Wang, W.; Xia, F.; Nie, H.; Chen, Z.; Gong, Z.; Kong, X.; Wei, W. Vehicle trajectory clustering based on dynamic representation learning of internet of vehicles. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 3567–3576. [CrossRef]
32. Tang, J.; Bi, W.; Liu, F.; Zhang, W. Exploring urban travel patterns using density-based clustering with multi-attributes from large-scaled vehicle trajectories. *Phys. A Stat. Mech. Appl.* **2021**, *561*, 125301. [CrossRef]
33. Lin, K.; Li, C.; Fortino, G.; Rodrigues, J.J. Vehicle route selection based on game evolution in social internet of vehicles. *IEEE Internet Things J.* **2018**, *5*, 2423–2430. [CrossRef]
34. Huang, Z.; Wu, J.; Lv, C. Driving Behavior Modeling Using Naturalistic Human Driving Data With Inverse Reinforcement Learning. *IEEE Trans. Intell. Transp. Syst.* **2021**. [CrossRef]
35. Tan, H.; Lu, G.; Liu, M. Risk Field Model of Driving and Its Application in Modeling Car-Following Behavior. *IEEE Trans. Intell. Transp. Syst.* **2021**. [CrossRef]
36. Liu, T.; Yang, G.; Shi, D. Construction of Driving Behavior Scoring Model based on OBD Terminal Data Analysis. In Proceedings of the 2020 5th International Conference on Information Science, Computer Technology and Transportation (ISCTT), Shenyang, China, 13–15 November 2020; pp. 24–27.
37. Shahverdy, M.; Fathy, M.; Berangi, R.; Sabokrou, M. Driver behavior detection and classification using deep convolutional neural networks. *Expert Syst. Appl.* **2020**, *149*, 113240. [CrossRef]
38. Chen, C.; Xiang, H.; Qiu, T.; Wang, C.; Zhou, Y.; Chang, V. A rear-end collision prediction scheme based on deep learning in the Internet of Vehicles. *J. Parallel Distrib. Comput.* **2018**, *117*, 192–204. [CrossRef]
39. Singh, R.; Saluja, D.; Kumar, S. Reliability improvement in clustering-based vehicular ad-hoc network. *IEEE Commun. Lett.* **2020**, *24*, 1351–1355. [CrossRef]
40. Zhu, B.; Yan, S.; Zhao, J.; Deng, W. Personalized lane-change assistance system with driver behavior identification. *IEEE Trans. Veh. Technol.* **2018**, *67*, 10293–10306. [CrossRef]
41. Xu, L.; Hu, J.; Jiang, H.; Meng, W. Establishing style-oriented driver models by imitating human driving behaviors. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 2522–2530. [CrossRef]
42. Li, J.; Cheng, J.h.; Shi, J.y.; Huang, F. Brief introduction of back propagation (BP) neural network algorithm and its improvement. In *Advances in Computer Science and Information Engineering*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 553–558.
43. Kim, S.; Shrestha, R. Intelligent autonomous vehicle. In *Automotive Cyber Security*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 15–33.
44. Kenney, J.B. Dedicated short-range communications (DSRC) standards in the United States. *Proc. IEEE* **2011**, *99*, 1162–1182. [CrossRef]

45. 3GPP TR 21.914 V14.0.0. Summary of Rel-14 Work Items[S]. *3GPP Release 14*. 2018. Available online: https://www.3gpp.org/release-14/ (accessed on 10 November 2021).
46. Shi, C.; Zhou, Y.; Li, W.; Li, H.; Lu, N.; Cheng, N.; Yang, T. A centralized clustering based hybrid vehicular networking architecture for safety data delivery. In Proceedings of the GLOBECOM 2017—2017 IEEE Global Communications Conference, Singapore, 4–8 December 2017; pp. 1–6.