

# Novel Online Sequential Learning-based Adaptive Routing for Edge Software-Defined Vehicular Networks

Liang Zhao, *Member, IEEE*, Weiliang Zhao, Ammar Hawbani, Ahmed Al-Dubai, *Senior Member, IEEE*  
Geyong Min, Albert Y. Zomaya, *Fellow, IEEE* and Changqing Gong

**Abstract**—To provide efficient networking services at the edge of Internet-of-Vehicles (IoV), Software-Defined Vehicular Network (SDVN) has been a promising technology to enable intelligent data exchange without giving additional duties to the resource constrained vehicles. Compared with conventional centralized SDVNs, hybrid SDVNs combine the centralized control of SDVNs and self-organized distributed routing of Vehicular Ad-hoc NETWORKS (VANETs) to mitigate the burden on the central controller caused by the frequent uplink and downlink transmissions. Although a wide variety of routing protocols have been developed, existing protocols are designed for specific scenarios without considering flexibility and adaptivity in dynamic vehicular networks. To address this problem, we propose an efficient online sequential learning-based adaptive routing scheme, namely, Penicillium reproduction-based Online Learning Adaptive Routing scheme (POLAR) for hybrid SDVNs. By utilizing the computational power of edge servers, this scheme can dynamically select a routing strategy for a specific traffic scenario by learning the pattern from network traffic. Firstly, this paper applies Geohash to divide the large geographical area into multiple grids, which facilitates the collection and processing of real-time traffic data for regional management in controller. Secondly, a new Penicillium Reproduction Algorithm (PRA) with outstanding optimization capabilities is designed to improve the learning effectiveness of Online Sequential Extreme Learning Machine (OS-ELM). Finally, POLAR is deployed in control plane to generate decision-making model (i.e., routing policy). Based on the real-time featured data, this scheme can choose the optimal routing strategy for a specific area. Extensive simulation results show that POLAR is superior to a single traditional routing protocol in terms of packet delivery ratio and latency.

**Index Terms**—hybrid SDVNs, VANETs, adaptive routing scheme, penicillium reproduction algorithm, OS-ELM.

Manuscript received xx xx, 2020; revised xx xx, 2020; accepted xx xx, 2020. Date of publication xx xx, 202x; date of current version December 18, 2020. This work is supported by the National Natural Science Foundation for Young Scientists of China (61701322), the Young and Middle-aged Science and Technology Innovation Talent Support Plan of Shenyang (RC190026), and Liaoning Provincial Department of Education Science Foundation (JYT19052). (Corresponding author: Ammar Hawbani; Weiliang Zhao).

Liang Zhao, Weiliang Zhao, and Changqing Gong are with the School of Computer Science, Shenyang Aerospace University, Shenyang, China (e-mail: lzhaos@sau.edu.cn, sau\_zwl@163.com, gongchangqing@sau.edu.cn).

Ammar Hawbani is with the School of Computer Science and Technology, University of Science and Technology of China, Hefei, China (e-mail: ammande@ustc.edu.cn).

Ahmed Al-Dubai is with the School of Computing; Edinburgh Napier University, Edinburgh, U.K. (e-mail: a.al-dubai@napier.ac.uk).

Geyong Min is with the Department of Mathematics and Computer Science, University of Exeter, Exeter, U.K. (e-mail: g.min@exeter.ac.uk).

Albert Y. Zomaya is with the School of Computer Science in University of Sydney, Australia (e-mail: albert.zomaya@sydney.edu.au).

## I. INTRODUCTION

MOBILE Edge Computing (MEC) brings new opportunities to Internet of Vehicles (IoV) by providing computational-rich resource for processing vehicular and transport related services. In particular, with MEC, the workload of connected and autonomous vehicles can be migrated to the resource comparably sufficient edge. With such benefits, intelligent services and relevant data exchanges can be realized among the edge and vehicles. In addition, one of the beneficial services is to intelligentize Vehicular Networks (VNs), in which the networking scheme will be enhanced by learning the fruitful traffic and network data from the IoV [1], [2].

VNs have been the cornerstone of IoV. In VNs, each vehicle node is equipped with an On-Board Unit (OBU) for implementing Vehicle-to-Everything (V2X) in the form of Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), Vehicle-to-Pedestrian (V2P), Vehicle-to-Edge (V2E), Vehicle-to-Cloud (V2C), Vehicle-to-Home (V2H), and Vehicle-to-Network (V2N). VNs are of great significance in strengthening traffic management, improving road safety, and providing entertainment content for users. The first important step towards the implementation of VANET (Vehicular Ad-hoc NETWORK) have been taken several standardization bodies, including the Society of Automotive Engineers (SAE) in the United States, the European Committee for Standardization (CEN) and the European Telecommunications Standards Institute (ETSI) in Europe, the Japanese Association of Radio Industries and Businesses (ARIB) in Japan and ISO, have continued the standardization of what they called Dedicated Short Range Communications (DSRC). Based on DSRC, the concept of VANET has made multi-hop networking possible.

The major limitation of VANETs is that they cannot adequately cope with the rapid changes in the network topology. The self-organized communication mode makes it impossible to manage the network based on the global information. Hence, along with the involvement of MEC, new communication architectures or schemes are continually developed to solve this problem. For instance, information-driven Software-Defined Networks (SDNs) architecture is designed to make content available to VNs. The architecture integrates the Information-Centric Networking (ICN) scheme to the SDN paradigm [3]. In [4], hierarchical SDVN-based architecture is used to construct stable clusters for improving the overall performance of vehicle communication. In [5], the authors in-

roduce an efficient routing scheme based on the collaboration between vehicles and Unmanned Aerial Vehicles (UAVs). The UAVs-assisted routing model provides an alternative scheme in the case of path failures. Besides, Named Data Network (NDN) is applied in VNs to overcome the path failure problem (caused by rerouting of link interruption) of existing TCP/IP due to the highly dynamic nature of VNs [6].

Based on the adaptive and flexible nature of SDVNs, network can be easily managed for V2V and V2I. SDVNs can be classified into centralized SDVNs and hybrid SDVNs. In centralized SDVNs, the controller calculates the optimal path as a response to the routing request of a node, in which the data forwarding plane is only responsible for data transmission. The networking operation of centralized SDVNs highly relies on the controller, which can cause an excessive network overhead from request and data collection, and computation overload from the calculation of routes. In contrast, the controller in hybrid SDVNs gathers global network information to assist routing decisions without handling specific path calculations, thus reducing the burden on the controller. Hybrid SDVNs enable the effective guidance for routing with avoiding exorbitant delays caused by handling routing requests in controllers. However, most adaptive routing protocols are designed for traditional VANETs with limited applications of artificial intelligence techniques for optimization, such as swarm intelligence and machine learning. Existing works are generally suitable for certain scenarios, whereas road traffic is dynamically changing due to terrain, topography, population density, and other factors which requires one routing scheme to cope with all situations.

Aiming at proposing a novel solution, this paper integrates the advantages of SDNs and VANETs in hybrid SDVNs to realize global management and combines an artificial neural network based on swarm intelligence to deal with global information and assist the routing process. The proposed routing scheme can construct a decision-making model to realize the selection of the optimal routing strategy for specific traffic conditions. The key contributions of this paper are summarized as follows.

1) A framework of hybrid SDVN is proposed for collecting the global information to assist vehicular networking. With applying Geohash [7], the local controller acts as an Mobile Edge Computing (MEC) server for regional management. In the control plane, an online sequence learning algorithm is proposed and embedded for learning the decision-making model based on traffic information to select the optimal routing strategy.

2) To achieve efficient online decision-making of routing policy, a swarm intelligence algorithm, namely, Penicillium Reproduction Algorithm (PRA), is proposed. With its potential capability in solving complex optimization problems, PRA is applied to improve the fitting and generalization ability of Online Sequential Extreme Learning Machine (OS-ELM) [8]. In order to reduce the probability of over-fitting and under-fitting, verification set has been added during the training process. PRA-optimized OS-ELM (i.e., PRO-ELM) improves testing accuracy based on feedback from the validation set under the restriction of the proposed constraint function.

3) We also introduce a data processing method for SDVNs to describe the real-time traffic conditions by extracting geographical features and traffic characteristics. These data are labeled as data sets for learning decision-making models. Through PRO-ELM, they can be easily trained chunk-by-chunk or one-by-one in real-time.

The rest of this paper is organized as follows. We present the related work on the development status of swarm intelligence and VNs in Section II. Section III introduces the efficient online sequential learning-based adaptive routing scheme, including the system model and specific implementation methods. In Section IV, simulation and results analysis are described in detail. Finally, we conclude our work in Section V.

## II. RELATED WORK

Artificial Intelligence (AI) brings new opportunities for the development of network technology. As a part of AI, swarm intelligence algorithm plays a crucial role in solving complex optimization problems. In routing technology, the application of swarm intelligence has enhanced its adaptive capabilities. In Subsection II-A, we introduce the development of swarm intelligence algorithm and its application in networking. Then, in Subsection II-B, the adaptive routing algorithms in VNs are elaborated in detail.

### A. Swarm Intelligence

Nature-Inspired Computing (NIC) is a type of meta-heuristic algorithms inspired by some natural phenomena, including biological, physical, or other phenomena. They have been widely used in science and engineering to solve some complex optimization problems. In NIC, swarm intelligence is a significant topic that refers to the behavior of simple individuals in a group through cooperation, competition, interaction, and learning. Without centralized control and global model, swarm intelligence provides a new idea for finding solutions to complex problems. As one of the most classic algorithms, the basic concept of *Particle Swarm Optimization (PSO)* [9] algorithm originates from the study of foraging behavior of birds as they often change direction, disperse and gather suddenly during flight, but the overall consistency is maintained. *PSO* simulates this behavior, that is, using the information-sharing mechanism to enable each individual to learn from each other's experience, to promote overall development. Besides, the recent study on swarm intelligence provides an effective way to solve complex optimization problems which can be applied to various fields and domains. In particular, in the field of networking, swarm intelligence is used for load balancing [10], computing offloading [11], and routing [12]. Inspired by the fireworks explosion, Y. Tan *et al.* [13] propose the *FireWorks Algorithm (FWA)* which forms a parallel explosive search method by introducing random factors and selection strategies. It is a global probability search method capable of solving complex optimization problems. As an application of networking, the study uses *FWA* to optimize energy consumption in a cluster-assisted Internet-of-Things (IoT) [14]. In [15], the design of *Grey Wolf Optimizer (GWO)* is inspired by the leadership level and hunting mechanism of the grey wolves

in nature. In [16], the authors propose a swarm intelligence algorithm based on *GWO*, which is applied to two different antenna design cases of 5G mobile communication. Moreover, in [17], *Whale Optimization Algorithm (WOA)* originated from the observation of humpback whales hunting behavior. In [18], the authors adopt *WOA* to solve multi-objective optimization to present a traffic-aware routing protocol. In light of the aforementioned work, swarm intelligence can effectively solve complex optimization problems in networks, which is a means to realize intelligent network management.

### B. Adaptive Routing in Vehicular Networks

The applications of AI technologies represented by swarm intelligence in Subsection II-A and other intelligent mechanisms can enhance the adaptability of routing. In VNs, some adaptive routing protocols work based in a distributed manner, while others are in centralized or semi-centralized architectures.

In VANETs, distributed adaptive routing protocols are self-organizing without the centralized management mechanism. In [19], an adaptive Quality-of-Service (QoS)-based routing for VANETs, namely, AQRV is introduced where AQRV transforms the routing selection process into a constrained optimization problem and solve it with *Ant Colony Optimization (ACO)* [20]. Moreover, other categories of intelligent mechanisms have been also applied in networking widely. A reinforcement learning-based hierarchical protocol is proposed in [21] which uses Q-learning to select grid-shaped regions or relay nodes hierarchically. In addition, inspired by the habits of spider, the authors [22] builds a spider web based on the network topology in which artificial spiders are used to find a feasible path to the destination node. However, due to the rapid changing of network topology, the routing information maintained by the source node to the destination node may expire due to the failure of consideration with a global view which is not enabled in VANETs.

As a solution to the above problem, SDVNs become an efficient communication paradigm which has attracted wide attention in recent years. Centralized SDVNs implement full centralized management that can handle specific routing requests. On the contrary, hybrid SDVNs are semi-centralized to assist vehicle communication. He *et al.* propose an SDN-based architecture to achieve rapid network innovation, which allows the combination of traditional distributed routing and softwarized management [23]. In [24], the authors propose a hierarchical geography routing scheme in SDVNs where it divides the area into grids according to the global view constructed by the controller. The process of routing selection is divided into three phases, the selection of regions, the selection of intersections in selected regions, and the selection of specific relay nodes. Besides, the proposed method DROM [25] realizes a customizable and universal routing optimization by adopting a deep reinforcement learning mechanism. DROM can intelligently adjust the reward function to optimize the customized parameters or strategies dynamically.

We can conclude that AI can assist the distributed routing schemes to improve their adaptability to various traffic

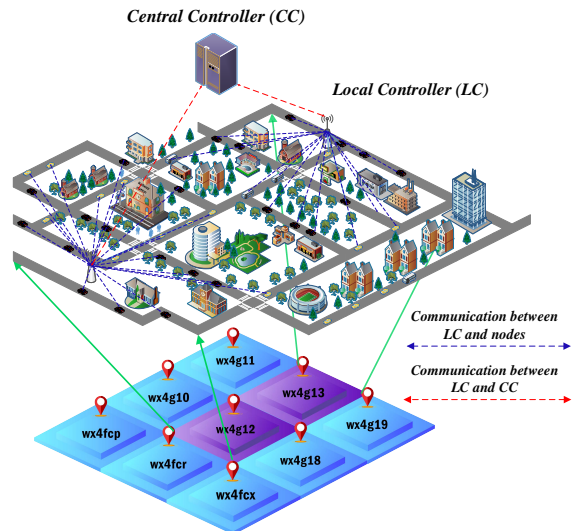


Fig. 1. Geohash coding of the area near Beijing Station

scenarios. However, the limitation of existing solutions is that the global information cannot be captured for routing in the conventional ad-hoc networking architecture. Besides, the distributed architecture is not conducive to the rapid innovation and flexibility of the network. On the contrary, SDN enables VNs to facilitate rapid network innovation and deployment, with its nature of programmability and flexibility. The nodes information collected by the controller can be employed to select routing paths or to assist routing decisions of VANETs in the data plane. In such a case, the application plane can deploy multiple modules to optimize for various scenarios of network traffic separately.

### III. HYBRID SDVN

This section firstly explains the assumptions of the environment and equipments required for the proposed scheme. Then, the framework of hybrid SDVNs and the functions of each modules in the framework are introduced.

#### A. Assumption

The fifth-generation (5G) cellular network provides high-capacity and low-latency communications to vehicles in highly dynamic environments, with the potential to meet the needs of applications in IoV. In this paper, we assume that each vehicle is equipped with two wireless interfaces to support 5G communication and DSRC, respectively. The speed and location information of a vehicle can be obtained through the Global Positioning System (GPS). In the hybrid SDVNs, the central controller and local controllers are connected by wired interface, while the communication between the control plane and data plane is carried out via 5G wireless interfaces. Also, DSRC is applied to enable the communications between vehicles and the Road Side Units (RSUs) in the data plane.

In a hybrid SDVN, the control plane consists of a central controller and local controllers. The data plane is composed of vehicle nodes, RSUs, and other infrastructures where all devices are equipped with DSRC units. In such an architecture,

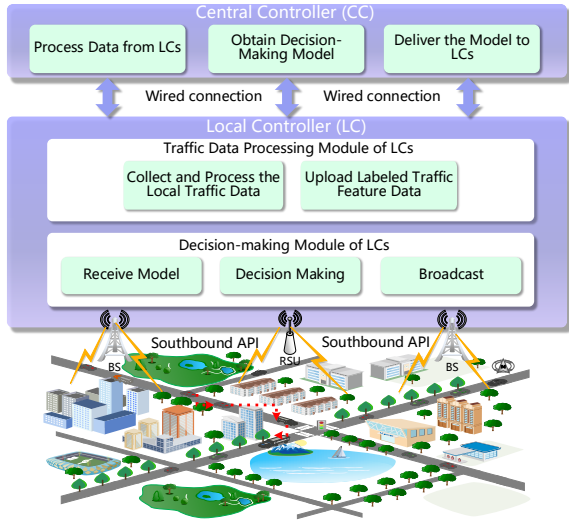


Fig. 2. The framework of hybrid SDVN

we combine centralized control with the distributed manner of VANET routing. The controller selects an optimal routing strategy based on layouts of the road network and the real-time traffic conditions. Compared with the centralized SDVN, the controller only assists in selecting a distributed routing strategy in accordance with the global information without calculating a specific routing path, which reduces the overhead caused by the fully centralized computing.

The scheme proposed in this paper is applied to the selection of the optimal routing strategy in a certain grid. For different grids, it will be given in the subsequent research. In order to achieve regionalized management, the control plane is divided into different levels. Local controllers are responsible for local management, while the central controller is used to consolidate the information of local controllers for the global allocation. In our architecture, a geographic coordinate system, Geohash is applied to encode urban areas and divides the given area into equal-sized grids. Each vehicle can transmit its speed, location, as well as other status data to the specific local controller by recognizing area codes. The bottom of Fig. 1 illustrates an example of the Geohash coded subarea, which is located near to the Beijing Station, China. Each area is represented by a unique string, where the length of the string describes the size of the partition area. The local controller is deployed in the center of each divided area.

### B. The Architecture of Hybrid SDVN

1) *Functional Modules*: Fig. 2 shows the functional modules of the hybrid SDVNs proposed in the paper. In this framework, the local controller sits in the middle where it can be divided into two functional modules in our proposed SDVNs. The first module (upper one) is responsible for collecting, processing, and uploading of the local real-time traffic data. The second module (lower one) implements routing policy selection and broadcasting. Algorithm 1 details the traffic data processing of local controllers. Vehicles periodically send speed and location information to the local controller during

#### Algorithm 1: Traffic Data Processing Module of LCs

---

```

1: Process one:
2:  $t = \text{Currenttime}$ 
3: repeat
4:    $V_{info} = \text{Collectvehicleinfo}(\text{Geocode})$ 
5:    $M_{info} = \text{Getmapinfo}(\text{Geocode})$ 
6: until  $t < t + \Delta t$ ;
7: Process two:
8:  $F_i = \text{Extract\_feature}(V_{info}, M_{info})$ 
9:  $\text{Trajectory} = \text{get\_movementfile}(V_{info}, M_{info})$ 
10:  $\text{Label} = \text{Evaluate\_routing\_schemes}(\text{Trajectory})$ 
11:  $\text{Labeled\_}F_i = \text{Insert}(F_i, \text{Label})$ 
12: Upload  $\text{Labeled\_}F_i$  to the central controller.

```

---

#### Algorithm 2: Central Controller Module

---

```

1:  $DS' = \text{Receive\_labeledfeature}(\text{LC\_list})$ 
2: if there is no decision-making model then
3:    $DS = \text{Extend}(DS')$ 
4:   if  $\text{Size}(DS) > N_{init}$  then
5:     Normalize the data set  $DS$ .
6:      $\text{Model} = \text{Initialized\_Learning}(DS)$ 
7:   end
8:   else
9:     Continue to collect feature data.
10:  end
11: end
12: else
13:   Normalize the data set  $DS'$ 
14:    $\text{Model} = \text{Online\_SequentialLearning}(DS')$ 
15: end
16: Send the current model to each local controller.

```

---

the time slice  $\Delta t$  (lines 2-6). In order to facilitate the usage of historical data, the local controller extracts the required feature data based on the digital map and traffic data (line 8). Then, a simulation environment is constructed on the basis of the information of vehicle movement. Through simulation, the local controller can obtain performance parameters of multiple routing strategies, and selects the optimal routing strategy by measuring these network performance parameters (lines 9-10). The selected routing strategy serves as the label for the extracted feature data (line 11). After the labeled data entry is formed, the local controller transmits the data to the central controller (line 12). In this algorithm,  $V_{info}$  and  $M_{info}$  represent the status information about vehicle movement and real-time map information, respectively.  $F_i$  denotes the feature data extracted in  $\Delta t$  that can describe the traffic condition.

Algorithm 2 shows the detail of central controller module. In this algorithm,  $DS$  refers to the data set used to initialize training while  $DS'$  denotes the recent data received from local controllers. In this module, the central controller receives labeled feature data describing the traffic conditions from each local controller, then forms the original training data set (lines 1-11). The original data set is used for initial learning to get

**Algorithm 3: Decision-making Module of LCs**


---

```

1:  $Model = Receivemodel(CC)$ 
2:  $RS = Select\_routing(Model, V_{info}, M_{info})$ 
3: if  $RS == Current\_RS$  then
4:   | Quit.
5: end
6: else
7:   |  $Current\_RS = RS$ 
8:   | Transfer  $Current\_RS$  to local nodes.
9: end

```

---

the decision-making model. When the initialization is completed, the central controller will execute the online sequential learning process (lines 13-14) to construct the decision-making model which then is distributed to local controllers (line 16). It should be noted that, before initial learning or updating the decision-making model, the central controller shall normalize the feature data set. Fig. 2 is a generic framework, where we deploy PRO-ELM as the online decision-maker in the central controller, as an efficient instance.

Algorithm 3 shows the second module of the local controller. In this module, it calculates the most appropriate routing strategy for the current traffic scenario. The calculation process is based on feature data representing current traffic conditions and a decision-making model derived from the central controller (line 1-2). If the current local routing strategy is the same as the new one, the controller will not perform any operations (lines 3-5). Otherwise, the latest routing strategy will be sent to local nodes through 5G network (lines 6-9).

#### IV. EFFICIENT ONLINE SEQUENTIAL LEARNING-BASED ADAPTIVE ROUTING SCHEME

In this section, we introduce the proposed routing scheme, namely, Penicillium reproduction-based Online Learning Adaptive Routing scheme (POLAR). Our proposed scheme can optimally tailor to the framework shown in Fig. 2 as POLAR can select the most appropriate routing strategy for a particular area according to the current traffic conditions based on online learning. For instance, Dynamic Source Routing (DSR) [26] outperforms Ad-hoc On-demand Distance Vector routing (AODV) [27] when nodes are dense and evenly distributed, while the packet loss rate and latency increase significantly in the sparse scenarios. In this context, POLAR then selects DSR as the optimal routing strategy regarding the learned model in a dense network. As observed from Fig. 3, POLAR is designed for hybrid SDVNs where its core components are PRO-ELM and the method of feature extraction. In this section, first, in order to improve the prediction accuracy of OS-ELM and better establish the mapping of traffic feature data to the optimal distributed routing strategy, we transform the optimization process of OS-ELM into a continuous variable optimization problem, and propose Penicillium Reproduction Algorithm (PRA) to solve this problem. Compared with classic algorithms and some intelligent optimization algorithms proposed in recent years, it has more excellent ability to solve complex optimization problems. Then, we apply PRA to

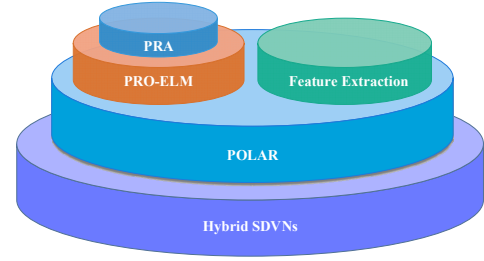


Fig. 3. Inclusion relations of some concepts

improve the learning efficiency of OS-ELM. Since PRO-ELM obtains decision-making models by training data set describing traffic conditions, we discuss the extraction of traffic feature data in the latter part of this section.

#### A. Penicillium Reproduction Algorithm

1) *Design of PRA:* We propose a novel swarm intelligence algorithm, namely, Penicillium Reproduction Algorithm (PRA) inspired by the behavior of penicillium reproduction. Penicillium is a kind of fungus in which the mycelium of penicillium contains a plurality of hyphae with a transverse septum. In general, they are propagated by producing conidia. In the production of spores, the apical part of the mycelium produces multicellular conidiophores. The apex of the stem branches two or three times, where the terminal cells of each branch split into clusters of conidia. Then, conidia disperse in the wind after maturation and germinate into hyphae when it meets a suitable environment.

Fig. 4 shows the initialization, the local and global exploration process of PRA. The largest circle of this figure represents a culture dish, which can be abstracted into the entire search space. In a culture dish suitable for the growth of penicillium, spores uniformly distributed in the air fall into the culture dish. A colony is formed in the most suitable area for the propagation. The circle on the left in Fig. 4 represents the above process, where the selected spores represent the optimal solution obtained by initialization of PRA. The reproduction process is shown in the circle on the right of the figure in which some of the spores produced by penicillium fall in the nearby area, while the other part flies to a farther area. In light of the above, we assume that there are two types of spores, *Global Exploration Spores (GES)* and *Local Exploration Spores (LES)*. *GES* and *LES* are responsible for searching the global suitable breeding area and the local suitable breeding area, respectively. The search radius of *GES* and *LES* varies depending on the number of iterations and the search results of the optimal solution.

We set the number of spores produced in each iteration to be a constant, in which the following equations give the number of *GES* and *LES*,

$$N_{LES_t} = N, \text{ if } \frac{(R_s(t))}{(UL - LL)} = 1 \quad (1)$$



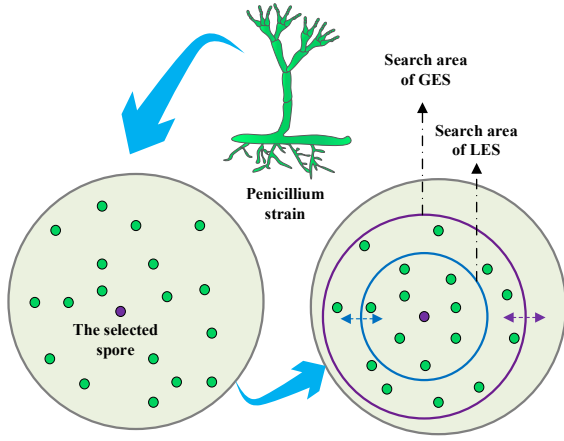


Fig. 4. Execution process of PRA

$$N_{LES\_t} = \left[ N \times \left( \left( \frac{Min_{LES\_r} \times t}{T_{iter}} \right)^2 - \frac{t}{T_{iter}} + Max_{LES\_r} \right), if FD \leq 5 \right] \quad (2)$$

$$N_{LES\_t} = \left[ N \times \frac{R_s(t)}{UL - LL} \right], if FD > 5 \quad (3)$$

$$N_{GES\_t} = N - N_{LES\_t} \quad (4)$$

where  $N_{GES\_t}$  and  $N_{LES\_t}$  represent the number of spores for global exploration and local exploration in  $t$ -th iteration, respectively.  $R_s(t)$  is the dispersal radius of locally explored spores in the  $t$ -th iteration.  $UL$  and  $LL$  refer to the upper and lower limits of the search space, respectively.  $Min_{LES\_r}$  denotes the minimum proportion of spores for local exploration to the total number of spores  $N$ , while  $Max_{LES\_r}$  is the opposite.  $T_{iter}$  is the total number of iterations of the algorithm.  $FD$  is a feedback factor that reflects the relationship between the current fitness value and the historical fitness value, which is used to adjust the number of  $LES$  and  $GES$ .

Here, we use Eq. (1) to indicate that the propagation radius of all spores is set to the size of search space at the beginning. In Eq. (2), as the number of iterations increases, the proportion of  $LES$  decreases; the rate of decline is faster than that of linear decline, which is set to reduce the probability of falling into local optimum.  $FD$  reflects changes in fitness values. We find that the fitness values of successive generations are the same at the later stage of the iteration. In order to avoid local optimum, we reduce the number of  $LES$  according to Eq. (3). Hence, most spores are used for global exploration.

In the local exploration process, the search radius of  $LES$  is computed by Eq. (5).

$$R_s(t) = \begin{cases} UL - LL & t = 1 \\ R_s(t-1) \times \delta & f_{spore}(t) < f_{spore}(t-1) \\ R_s(t-1) \times \gamma & f_{spore}(t) \geq f_{spore}(t-1) \end{cases} \quad (5)$$

where  $f_{spore}(t)$  represents the fitness value of the most suitable position for penicillium reproduction in the  $t$ -th iteration. In the initial phase of the algorithm, the spore propagation radius is set to the size of the search space. Generally,

---

**Algorithm 4:** Local exploration of spores
 

---

```

1: Calculate the number of LES by Eq. (1), (2), (3).
2: Calculate the search radius of LES by Eq. (5).
3: for  $n = 1$  to  $N_{LES\_t}$  do
4:   for  $m = 1$  to  $len(S(n))$  do
5:      $S(n)_m = Pos_m + rand(-R_s(t), R_s(t))$ 
6:     if  $S(n)_m$  out of limits then
7:        $S(n)_m = Rand(LL_m, UL_m)$ 
8:     end
9:   end
10: end
  
```

---



---

**Algorithm 5:** Global exploration of spores
 

---

```

1: Calculate the number of GES by Eq. (4).
2: Calculate SD by Eq. (7).
3: for  $n = N - N_{LES\_t} + 1$  to  $N$  do
4:   Randomly select a dimension  $\rho$ .
5:    $PT = \frac{Pos_\rho + R_s(t) - LL}{UL - LL - 2 \times R_s(t)}$ 
6:   if  $Rand(0, 1)$  then
7:      $S(n)_\rho = Pos_\rho - R_s(t) - |Nrand(0, SD^2)|$ 
8:   end
9:   else
10:     $S(n)_\rho = Pos_\rho + R_s(t) + |Nrand(0, SD^2)|$ 
11:  end
12:  if  $S(n)_\rho$  out of limits then
13:     $S(n)_\rho = Rand(LL_\rho, UL_\rho)$ 
14:  end
15: end
  
```

---

when penicillium encounters a more suitable environment, the change of colonies follows the principle of greed, which is manifested in its larger scale. In the absence of a better environment, the depletion of nutrients would make the colonies unable to maintain their original size. Given the above natural phenomena, we set the greed factor  $\delta$  and contraction factor  $\gamma$  to represent the adaptation of penicillium to the environment. If the fitness value of the current iteration is better than previous iteration, the search radius of the spore should be increased; otherwise, it should be reduced.

The local exploration process of PRA is shown in Algorithm 4.  $S(n)$  represents the location of a spore, which can be multidimensional.  $S(n)_m$  represents a certain dimension of the  $S(n)$ .  $Pos$  is the location of a spore with the best fitness value in the previous generation, while  $Pos_m$  is a certain dimension of  $Pos$ . The idea that PRA selects the best location of the previous generation as the center for the next-round spore propagation is in line with natural phenomena.

Local exploration of spores can find locations suitable for reproduction near the current area of the colony. In order to avoid falling into local optimum,  $GES$  is also required for global exploration. In this case,  $GES$  can explore the entire area of the search space. In the process of global exploration, PRA randomly selects a dimension of  $Pos$ , and then adjusts the value of the dimension by a random number generated by the Gaussian distribution. This random number is used

---

**Algorithm 6:** Penicillium reproduction algorithm
 

---

```

1:  $FD=0$ 
2:  $Fitnesslist [T_{iter}] = Null$ 
3: repeat
4:   Local exploration of spores in Algorithm 1.
5:   Global exploration of spores in Algorithm 2.
6:   Measure the fitness value of each spore.
7:   Select the spore ( $Pos$ ) with the best fitness ( $BF_{Pos}$ ).
8:    $Fitnesslist [t] = BF_{Pos}$  if  $t < 1$  then
9:     if  $Fitnesslist [t] == Fitnesslist [t - 1]$  then
10:       $FD = FD + 1$ 
11:    end
12:    else
13:       $FD = 1$ 
14:    end
15:  end
16: until the termination condition is met;
17: return  $Pos$  and the fitness of  $Pos$ 
    
```

---

as a *Global Exploratory Variable (GEV)*, and the Gaussian distribution it obeys is as follows,

$$\begin{cases} GEV \sim N(S(n) - R_s(t), SD^2) & GEV \leq S(n) - R_s(t) \\ GEV \sim N(S(n) + R_s(t), SD^2) & GEV \geq S(n) + R_s(t) \end{cases} \quad (6)$$

where  $S(n) \pm R_s(t)$  is the mean of the Gaussian distribution. The change of operational symbol depends on the direction in which the spore flies before landing. In the local exploration process of *LES*, since spores fall within  $(-R_s(t), R_s(t))$ , this area should be avoided during the global exploration of *GES*. As shown in Eq. (7),  $SD$  is the standard deviation where it dynamically changes according to the number of iterations and variations in fitness values.

$$SD = \begin{cases} UL - LL & t = 1 \\ SD(t-1) \times 0.8 & f_{spore}(t) \geq f_{spore}(t-1) \\ UL - LL \times \gamma & f_{spore}(t) < f_{spore}(t-1) \end{cases} \quad (7)$$

The global exploration process of PRA is shown in Algorithm 5, where  $PT$  is the ratio of two-interval lengths outside the local search area. It is set to make the probability of *GES* scattering in each interval proportional to the interval length.

Algorithm 6 describes the whole process of PRA, including local exploration and global exploration of spores, selection strategy, and feedback regulation. We assume that the number of spores produced in each generation is a fixed value  $N$ , then the complexity of the algorithm can be expressed as  $O(N \times t)$ . In order to demonstrate the optimization efficiency of PRA, we adopt benchmark functions to evaluate its performance by comparing it with other classical swarm intelligence algorithms. The detailed comparison is shown in the supplementary document.

### B. PRA Optimized Online Sequential Extreme Learning Machine

The core of our proposed framework is the online learning algorithm deployed in the controller for learning the adaptive

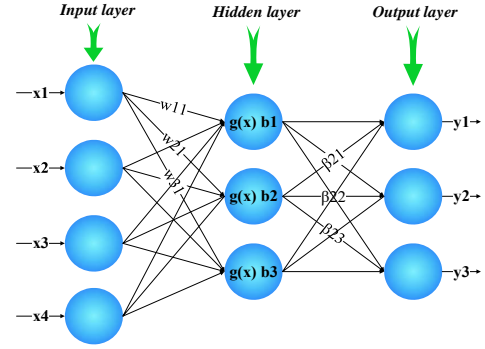


Fig. 5. Structural sketch of ELM

routing scheme. To improve the efficiency, we apply our newly proposed PRA to optimize the performance of the online sequential extreme learning machine, and then utilize this algorithm (PRO-ELM) as the online learning method for decision-making.

1) *Original ELM and OS-ELM*: ELM is a type of Single-hidden Layer Feedforward Neural network (SLFN) [28]. This artificial neural network can effectively solve the classification problem. As shown in Fig. 5, it consists of an input layer, a hidden layer, and an output layer. The connection weight between the input layer and the hidden layer and the biases in the hidden layer are randomly generated. The number of hidden layer neurons is the only parameter that needs to be manually set in ELM. The remaining parameters do not need to be adjusted. According to the principle of ELM, we can get Eq. (8).

$$H\eta = T \quad (8)$$

where  $H$  represents the hidden layer output matrix.  $\eta$  denotes the weight matrix of the hidden layer to the output layer.  $T$  is the output matrix obtained from the labels of the feature data entries. Through matrix operations, we can get Eq. (9).

$$\hat{\eta} = H^\dagger T \quad (9)$$

Then, the derived weight matrix  $\hat{\eta}$  between the hidden layer and the output layer will be used to predict the label of the new data entry. ELM is an algorithm designed for offline learning. As the data scale increases, it cannot update the existing models in real-time. As a result, for conventional ELM, we can only have a new decision-making model through retraining. Therefore, OS-ELM is proposed to implement online learning which can learn data one-by-one or chunk-by-chunk (a block of data) with a fixed or varying chunk size. When the new data set arrives, OS-ELM does not need to repeat training by merging the new data with the original data, but it only adjusts the original model according to the characteristics of the new data set.

$\varphi_0$  and  $\varphi_1$  are the original training data set and the data set used to update the model, respectively.  $N_i$  represents the size of the data set.  $x_i$  is the feature data entry, while  $t_i$  is the label corresponding to  $x_i$  shown as follows.

$$\varphi_0 = \{(x_i, t_i)\}_{i=1}^{N_0} \quad \varphi_1 = \{(x_i, t_i)\}_{i=N_0+1}^{N_0+N_1} \quad (10)$$

TABLE I  
VARIABLES DEFINITION

Variables	Description
$DS_{init}$	Normalized training data set in offline learning
$DS_v$	Normalized validation data set in offline learning
$N_{hd}$	Number of hidden neurons
$Actfunc$	The activation function of the hidden layer
$T$	A matrix transformed from the labels of $DS_{init}$
$P$	A matrix transformed from feature data of $DS_{init}$
$T'$	A matrix transformed from the labels of $DS_{init}$
$T'$	Zero matrix of $N_{tn}$ rows and $N_{ip}$ columns
$N_{tn}$	The number of data entries
$N_{ip}$	The number of input neurons
$IW$	The weight between the input layer and the hidden layer
$Bias$	The bias of hidden layer
$IB$	$IW$ and $Bias$ converted to $IB$
$T_v$	A matrix transformed from the labels of $DS_v$
$P_v$	A matrix transformed from feature data of $DS_v$
$DS_{block}$	Normalized data block in online learning
$T_b$	A matrix transformed from the labels of $DS_{block}$
$P_b$	A matrix transformed from feature data of $DS_{block}$

Then, the problem is transformed to minimize Eq. (11), in which  $H_0$  and  $T_0$  are derived from the original training data while  $H_1$  and  $T_1$  are obtained from the new data set.

$$\left\| \begin{bmatrix} H_0 \\ H_1 \end{bmatrix} \beta - \begin{bmatrix} T_0 \\ T_1 \end{bmatrix} \right\| \quad (11)$$

Through a series of derivations, the update equation of the weight  $\eta$  in the online sequential learning phase can be obtained.

$$P_{m+1} = P_m - P_m H_{m+1}^T (I + H_{m+1} P_m H_{m+1}^T)^{-1} H_{m+1} P_m \quad (12)$$

$$\eta^{(m+1)} = \eta^m + P_{m+1} H_{m+1}^T (T_{m+1} - H_{m+1} \eta^{(m)}) \quad (13)$$

where  $P_{m+1} = R_{m+1}^{-1}$ . The derivation process of  $R_i$  is shown in Eq. (14) and Eq. (15). More details of the calculation can refer to [8].

$$R_0 = H_0^T H_0, \quad R_1 = R_0 + H_1^T H_1 \quad (14)$$

$$R_{m+1} = R_m + H_{m+1}^T H_{m+1} \quad (15)$$

2) *PRO-ELM*: From above, we know that the input weights and biases of the hidden layer in OS-ELM are randomly generated. However, the drawback is random parameters can make the prediction accuracy fluctuating greatly. In order to select the optimal hidden layer input weights and biases, we use PRA to improve the performance of OS-ELM. In this case, the optimization process of OS-ELM is transformed into a continuous variable optimization problem, which can be effectively solved by PRA. In addition, we add a validation set to the training process to reduce the probability of over-fitting and under-fitting. Eq. (16) is the fitness function, where  $ACC_{tn}$  and  $ACC_v$  represent training accuracy and verification accuracy, respectively. The range of these two variables is  $[0, 1]$ .

$$Fitness = 1 - ACC_{tn} \times ACC_v + |ACC_{tn} - ACC_v| \quad (16)$$

The goal of the training process is to find the smallest fitness value. Within the domain of  $ACC_{tn}$  and  $ACC_v$ , the minimum fitness value is 0. We can find that, as the two variables become

---

#### Algorithm 7: Initialized Learning Function

---

**Input:**  $DS_{init}$ ,  $DS_v$ ,  $N_{hd}$ ,  $Actfunc$ ,  $IB$

**Output:**  $\eta$ ,  $P_0$ ,  $Fitness$

- 1:  $T$ ,  $P = Split(DS_{init})$
  - 2: Get  $N_{tn}$  and  $N_{ip}$  according to the dimension of  $P$ .
  - 3:  $T' = Zeromatrix(N_{tn}, N_{ip})$
  - 4: **for**  $n = 1$  **to**  $N_{tn}$  **do**
  - 5:    $T' [n, T [n, 0]] = 1$
  - 6: **end**
  - 7:  $T = 2 \times T' - 1$
  - 8:  $IB' = reshape(IB, N_{hd}, N_{ip} + 1)$
  - 9:  $IW = IB'[:, 1 : N_{ip}]$
  - 10:  $Bias = IB'[:, N_{ip} + 1]$
  - 11:  $H_0 = Actfunc(P, IW, Bias)$
  - 12: Calculate  $R_0$  according to Eq. (14).
  - 13:  $P_0 = R_0^{-1}$
  - 14: Calculate  $\eta$  according to Eq. (9).
  - 15: Get  $T_v, P_v$  by steps similar to 1-7.
  - 16:  $H_v = Actfunc(p_v, IW, Bias)$
  - 17:  $Y = H_0 \eta$ ,  $Y_v = H_v \eta$
  - 18: Calculate the  $ACC_{tn}$  and  $ACC_v$ .
  - 19: Calculate  $Fitness$  according to Eq. (16).
  - 20: **return**  $Fitness$ ,  $\eta$ ,  $P_0$
- 

larger and the absolute values of the differences between them become smaller, the fitness value will become smaller. In the training process, we only use the training set as the input of the neural network. More,  $ACC_{tn}$  reflects the fitting degree between the model and training set, while  $ACC_v$  is the embodiment of generalization ability of the model. The constraint function (16) constrains the trend of  $ACC_{tn}$  and  $ACC_v$ .

In PRO-ELM, PRA acts on the objective function to get the optimal solution. Algorithm 7 and Algorithm 8 are the objective functions of offline learning phase and the online sequential learning phase, respectively. Table I shows the definitions of the variables mentioned in Algorithm 7 and Algorithm 8. In the basic OS-ELM,  $IW$  and  $Bias$  are matrices composed of random numbers in  $[-1, 1]$ , which leads to instability of the decision-making model. We use PRA to determine  $IW$  and  $Bias$  to increase the stability of the model. In addition, the probability of over-fitting and under-fitting is reduced by the limits of the constraint function.

In Algorithm 7 and Algorithm 8,  $IB$  is made up of  $IW$  and  $Bias$  through specific transformations, retaining all information of them. Here,  $IB$  and  $(IW, Bias)$  can be converted to each other in which  $IB$  is equivalent to spores in the PRA algorithm.  $UL$  and  $LL$  are 1 and -1, respectively. PRA generates  $N$  spores with dimension  $N_{hd} \times (N_{ip} + 1)$  in each iteration and evaluates each spore according to the fitness values calculated in Algorithm 4 or Algorithm 5. By performing multiple iterations, PRO-ELM then can select the optimal spore corresponding to the minimum fitness value. At this point, we can obtain the weight of the hidden layer to the output layer and convert the optimal spore into  $IW$  and  $Bias$ .



**Algorithm 8:** Online Sequential Learning Function**Input:**  $DS_{block}$ ,  $DS_v$ ,  $IB$ ,  $\eta$ ,  $P_i$ **Output:**  $IW$ ,  $Bias$ ,  $\eta$ ,  $P_{i+1}$ ,  $Fitness$ 

- 1:  $IB' = reshape(IB, N_{hd}, N_{ip} + 1)$
- 2:  $IW = IB'[:, 1 : N_{ip}]$
- 3:  $Bias = IB'[:, N_{ip} + 1]$
- 4:  $T_b, P_b = Split(DS_{block})$
- 5: Get the block size  $N_b$  of  $DS_{block}$ .
- 6:  $T_b' = Zeromatrix(N_b, N_{ip})$
- 7: **for**  $n = 1$  **to**  $N_{tn}$  **do**
- 8:    $T_b'[n, T[n, 0]] = 1$
- 9: **end**
- 10:  $T_b = 2 \times T_b' - 1$
- 11:  $H_i = Actfunc(P_b, IW, Bias)$
- 12: Calculate  $R_{i+1}$  according to Eq. (14) or (15).
- 13:  $P_{i+1} = R_{i+1}^{-1}$
- 14: Calculate  $\eta$  according to Eq. (13).
- 15: Get  $T_v, P_v$  by steps similar to 1-7.
- 16:  $H_v = Actfunc(p_v, IW, Bias)$
- 17:  $Y = H_0\eta$ ,  $Y_v = H_v\eta$
- 18: Calculate the  $ACC_{tn}$  and  $ACC_v$ .
- 19: Calculate  $Fitness$  according to Eq. (16).
- 20: **return**  $Fitness$ ,  $\eta$ ,  $P_{i+1}$

TABLE II  
FEATURE DATA DEFINITION

Definition	Feature
$D_i$	The density of intersections
$D_{tl}$	The density of traffic lights
$D_{vr}$	The density of vehicles at roads
$D_r$	The density of roads
$N_{rl}$	The number of road lanes
$SR_{max}$	Maximum speed of roads
$P1_{vi}$	Proportion one of vehicles at intersections
$P2_{vi}$	Proportion two of vehicles at intersections
$P3_{vi}$	Proportion three of vehicles at intersections
$P4_{vi}$	Proportion four of vehicles at intersections

Finally, PRO-ELM can combine  $(IW, Bias, \eta)$  with the real-time traffic feature data to realize the online decision-making.

For verifying the learning ability of PRO-ELM, we detect the efficiency of PRO-ELM for solving online classification problems based on some public data sets. The detailed comparison is shown in the supplementary document.

### C. Collect Road Network Feature Data

We focus on extracting real-time traffic feature data in this subsection. The dynamic movement of vehicles leads to changes in the network topology. Since the form of the network topology affects the network performance obviously, the extraction of traffic feature data describing the real-time traffic scenarios can provide auxiliary information for determining the routing strategy. Routing protocols exhibit different performances in various traffic scenarios. In order to form a mapping of traffic data to an optimal routing strategy, the controller needs to extract feature data describing the traffic conditions.

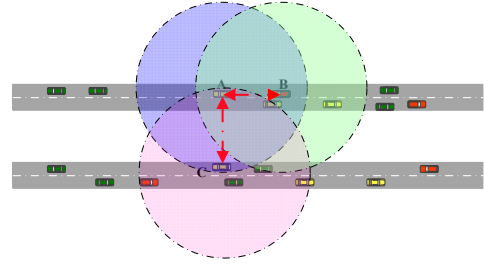


Fig. 6. Selection of relay node

In POLAR, local controllers are responsible for collecting local vehicle movement data and real-time map information. Table II shows the definition of feature data.  $D_i$  represents the density of the intersection, and  $D_{tl}$  denotes the density of traffic lights. If a vehicle encounters a traffic light, the average speed of the vehicle will decrease, which has an impact on the change of network topology.  $D_{vr}$  is the density of vehicles at roads. Regarding the mechanisms of multi-hop routing schemes, we can find that most of them transmit data along the road by observing the communication links during simulations. Hence, we have  $D_{vr}$  reflects the average distance of vehicles on the road, which affects the efficiency of data transmission. We apply  $N_v$  to indicate the number of vehicles in the local area, and  $S_{loc}$  to be the area of the region. Then  $D_{vr}$  can be expressed by Eq. (17).

$$D_{vr} = N_v / S_{loc} \quad (17)$$

Fig. 6 shows the selection of relay nodes. High vehicle-density roads increase the possibility of communication between node A on a lane/road and node C on another lane/road, which indicates that the vehicle-density of roads can affect the number of potential data forwarding paths. Hence, we set  $D_r$  as an extracted feature.  $R_{tl}$  represents the total length of the road in the region. Then,  $D_r$  can be expressed as Eq. (18).

$$D_r = R_{tl} / S_{loc} \quad (18)$$

$N_{rl}$  is the number of road lanes because multi-lane enables more options for relaying data messages.  $SR_{max}$  is the maximum speed of roads. Here, we set the maximum speed permit of the road when generating the movement file, and then get the speed of vehicles according to the normal distribution. As a result, we use  $SR_{max}$  as a feature to describe the overall condition of the traffic. Of course, we can also use the average speed instead, which can also reflect the traffic conditions of the road. The local controller collects and processes traffic data periodically. We divide period  $T$  into 4 time slots.  $P1_{vi}$ ,  $P2_{vi}$ ,  $P3_{vi}$  and  $P4_{vi}$  represent the proportion of vehicles at intersections of four moments, respectively. When there is a traffic congestion, the proportion of vehicles around the intersection will increase significantly, which affects the connectivity of the network. In the following, we give the details about the method of calculating the proportion of vehicles at intersections.

The local controller retains the location information of intersections. Besides, it stores the information of road connectivity in the form of the adjacency matrix. If two intersections are

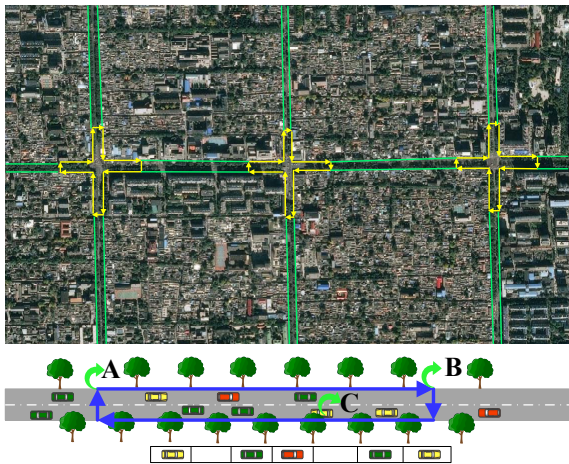


Fig. 7. Statistics on the number of intersection nodes

connected directly, the corresponding element is set to one. Otherwise, the value of the element is zero. For obtaining the proportion of vehicles around intersections, we use the judgment method of the position relationship between the point and the straight line to realize it. As shown in Fig. 7, we assume that the coordinates of node  $A$ , node  $B$ , and node  $C$  are  $(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $(x_3, y_3)$ , where  $A$  and  $B$  are the start-point and end-point of a vector, respectively; and  $C$  is an arbitrary point that needs to determine the position relationship with  $\vec{AB}$ . The function for determining the positional relationship between a point and a straight line is shown in Eq. (19).

$$\xi = (x_1 - x_3) \times (y_2 - y_3) - (x_2 - x_3) \times (y_1 - y_3) \quad (19)$$

If  $\xi$  is a positive number, then  $C$  is on the left side of  $\vec{AB}$ ; if  $\xi$  is a negative number, then  $C$  is on the right side of  $\vec{AB}$ ; if  $\xi$  is 0, then  $C$  is on line  $\vec{AB}$ . As shown in Fig. 7, we consider a road segment as a graph surrounded by vectors in a clockwise direction. The position relationship between point  $C$  and vectors can be obtained by Eq. (19). If each  $\xi$  is less than 0, point  $C$  is in this road segment. As a primary condition, local controllers can obtain the position information of vehicles. We apply the intersection position and the adjacency matrix representing the connection information, to divide the area near intersections. The number of vehicles around intersections can be counted through the discrimination of Eq. (19). Then, we can calculate the proportion of vehicles at intersections.

## V. SIMULATION RESULTS

This section demonstrates the experimental deployment and analysis of the simulation results. We use NS-3 platform [29] to implement the simulation. In order to ensure the diversity of traffic feature data, we need to simulate under different road networks. However, the objective fact is the real road networks are difficult to obtain. Hence, our maps for simulation are from mixed sources where part of the maps used is automatically generated according to the features of road networks. Besides, we also use digital maps of some areas in Shenyang, China, downloaded from OpenStreetMap [30] for simulation. We

TABLE III  
PARAMETERS DEFINITION

<i>Description</i>	<i>Value</i>
Number of vehicles	50 ~ 250
Number of lanes	2, 4
Area of maps	0.5 ~ 1.5 $km^2$
Max speed in roads	30 ~ 90 $km/h$
Transmission range	250 $m$
Propagation Loss Model	Two Ray Ground [35]
Packets generation speed	2 <i>packets/s</i>
Max Queue Length	64
Max Queue Time	30 $s$
Packets size	512 <i>bytes</i>
Communication nodes rate	20%
MAC protocol	802.11p
Data rate	6 $Mbps$

adopt SUMO [31] to generate trajectories of the vehicle nodes with consideration of traffic lights based on the real scenarios. In this context, vehicles should adjust the speed according to the state of the traffic lights, which makes the trajectories of the vehicles more realistic. The strength of our proposed POLAR is that it can switch among different routing strategies according to the result of decision-making process in various traffic scenarios, adaptively. Hence, we evaluate the performance of POLAR by involving the classical routing strategies, including Ad-hoc On-demand Distance Vector routing (AODV), Optimized Link State Routing (OLSR) [32], Greedy Perimeter Stateless Routing (GPSR) [33], Dynamic Source Routing (DSR) and Destination-Sequenced Distance-Vector routing (DSDV) [34], where these routing strategies are also the fundamental of POLAR.

### A. Parameter Setup

Table III shows the parameter settings for the simulation. In order to ensure the validity of the simulation results, we use various structures of road network in the simulation process. Considering the signal coverage of the 5G base station, we set the map covers an area of 0.5 ~ 1.5 square kilometers. In general, the distribution of roads in different maps can significantly affect the communication between vehicles. If the density of road segments in a given area is high, a vehicle can communicate with vehicles on other roads instead of transmitting information along this road. It is known that traffic conditions change during the day. For example, in urban environments, traffic congestion may occur in rush hours, otherwise, the density of vehicles in other periods is relatively low. Thus, in this paper, we vary the vehicle density to simulate all kinds of possible road scenarios which also maintains the diversity of different levels of data generated on the road. According to Table V, the number of vehicles is generated randomly where the change of vehicle density affects the average distance between vehicles and their distribution, as well as data transmissions. We also set the maximum speed allowed by the road segment to limit the speed of vehicles where the speed of vehicles is consistent with the normal distribution. In addition, for the same map, we also adjust the number of lanes, the number of traffic lights and other parameters to obtain traffic status under different traffic conditions.

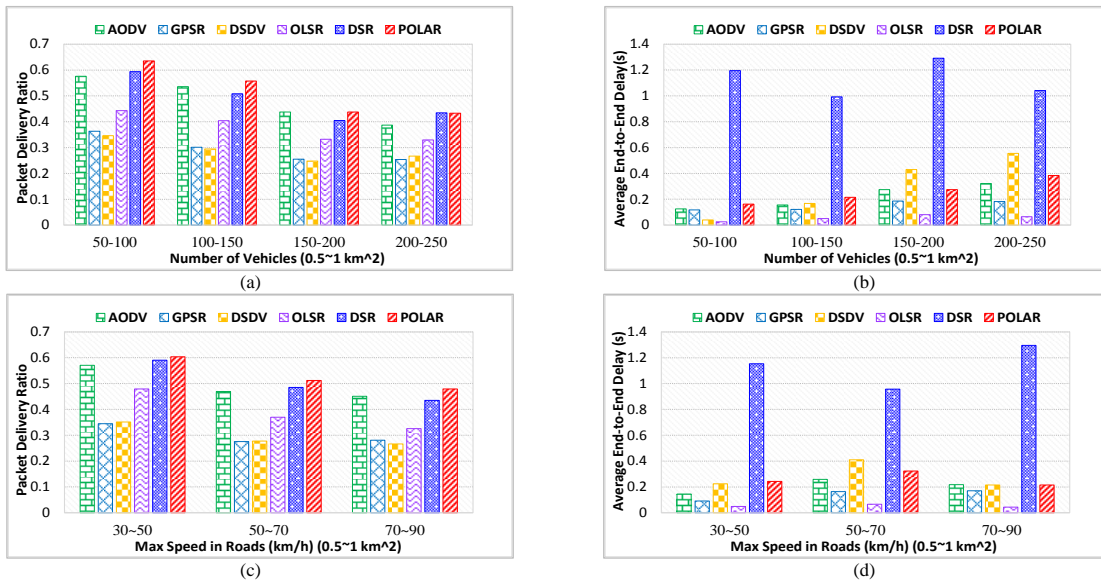


Fig. 8. Simulation results of PDR and AEED in  $0.5 \sim 1 \text{ Km}^2$  region

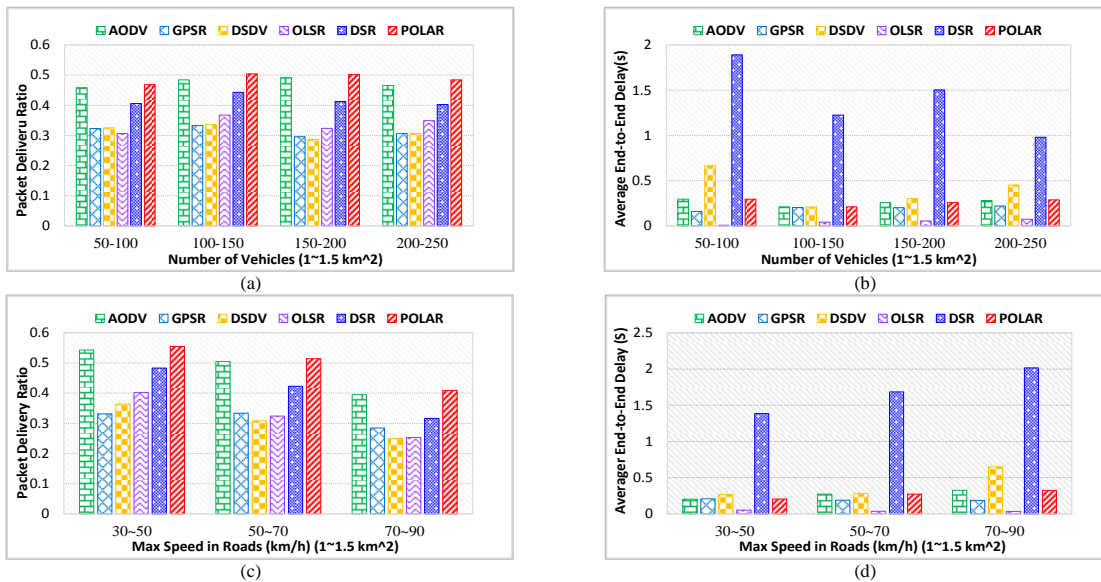


Fig. 9. Simulation results of PDR and AEED in  $1 \sim 1.5 \text{ Km}^2$  region

Besides considering the construction of the network topology shown above, we also set some network parameters of NS-3. We use Two Ray Ground as the propagation loss model. Each source node sends two packets per second and the packet size is 512 bytes. In the simulation, each routing protocol has the same sized buffer. The max queue length is the maximum number of packets that we allow a routing protocol to buffer. The max queue time is the maximum time packets can be queued. We randomly select 20% of the nodes as communication nodes, where all equipped with 802.11p units.

### B. Simulation Performance

We compare the performance parameters of POLAR and various distributed routing protocols through simulation, in-

cluding Packet Delivery Ratio (PDR) and Average End-to-End Delay (AEED). Over the simulation time, PDR refers to the proportion of packets received by destination nodes to the total number of packets sent by source nodes and AEED is the average time interval between the time when source nodes send the packets and the time when destination nodes receive the corresponding packets. In [36], it mentions that single-hop delay limited by DSRC technology is 50 millisecond. However, the scenario of our simulation is multi-hop communication. Its AEED is affected by many factors, including the number of relay nodes and queuing delay during the data transmissions. When labeling feature data, we give priority to routing protocols with high PDR, and AEED should be as small as possible. It should not exceed 1 second. In the simulation, we verify that POLAR can effectively select

TABLE IV  
COMPARISON OF POLAR AND OLAR

Variables	0.5 ~ 1.0 km <sup>2</sup>		1 ~ 1.5 km <sup>2</sup>	
	Packet Delivery Ratio		Packet Delivery Ratio	
Number of Vehicles	POLAR	OLAR	POLAR	OLAR
50 ~ 100	<b>0.635198121</b>	0.612080474	<b>0.468602676</b>	0.465602676
100 ~ 150	<b>0.557596253</b>	0.524542894	<b>0.504492654</b>	0.492015381
150 ~ 200	<b>0.437386588</b>	<b>0.437386588</b>	<b>0.501886737</b>	<b>0.501886737</b>
200 ~ 250	<b>0.431131158</b>	0.422933986	<b>0.484267445</b>	0.479267445
Max speed (km/h)	Packet Delivery Ratio		Packet Delivery Ratio	
	POLAR	OLAR	POLAR	OLAR
30 ~ 50	<b>0.603854951</b>	0.585553514	<b>0.555172339</b>	0.532814384
50 ~ 70	<b>0.511842574</b>	0.508035509	<b>0.514245744</b>	0.504842366
70 ~ 90	<b>0.479131893</b>	0.458210376	<b>0.409140145</b>	0.395460363
Number of Vehicles	Average End-to-End Delay (s)		Average End-to-End Delay (s)	
	POLAR	OLAR	POLAR	OLAR
50 ~ 100	<b>0.162313855</b>	0.640305745	<b>0.295891106</b>	0.315891106
100 ~ 150	<b>0.215635626</b>	0.628781367	<b>0.251267372</b>	0.389720799
150 ~ 200	<b>0.274962316</b>	<b>0.274962316</b>	<b>0.257984669</b>	<b>0.257984669</b>
200 ~ 250	<b>0.383866755</b>	0.421205007	<b>0.287068272</b>	0.327068272
Max speed (km/h)	Average End-to-End Delay (s)		Average End-to-End Delay (s)	
	POLAR	OLAR	POLAR	OLAR
30 ~ 50	<b>0.243065122</b>	0.609485271	<b>0.205066764</b>	0.338906835
50 ~ 70	<b>0.323891022</b>	0.331337569	<b>0.273605206</b>	0.295154214
70 ~ 90	<b>0.214934076</b>	0.631294291	<b>0.324106648</b>	0.342548774

the distributed routing strategy in the current traffic condition according to the traffic feature data. Also, we use POLAR and OS-ELM-based Adaptive Routing scheme (OLAR) [37] as learning algorithms, respectively, to compare their performance.

Fig. 8 shows the effect of vehicle number and speed on the PDR and the AEED for different routing protocols in the range of 0.5 to 1.0 square kilometers. In Fig. 8(a), we can find that with the increase of vehicles, the PDR of various routing protocols shows a downward trend. This is due to the traffic congestion. The increase in the proportion of vehicles at the intersection leads to disconnection of the network. However, the PDR of POLAR is the highest, which shows that it can effectively select the optimal routing protocol under current traffic conditions. Fig. 8(b) is the AEED corresponding to Fig. 8(a). As observed from Fig. 8, POLAR can improve PDR while avoiding excessive AEED, thus ensuring the Quality-of-Service (QoS). Fig. 8(c) is the effect of the maximum speed limited by the road on the PDR of multiple routing protocols. Fig. 8(d) is the AEED corresponding to Fig. 8(c). In general, the low speed of vehicles allows more stable wireless signal and less dynamic change of network topology which is considered as an advantage for data transmission. Hence, with the increase of vehicle speed, PDR shows a downward trend in Fig. 8(c), whereas the overall performance of POLAR is the best. POLAR can select the optimal routing strategy to improve PDR under different topology change speed. In Fig. 8(d), the AEED of POLAR is relatively low. Through a comprehensive analysis of Fig. 8(c) and Fig. 8(d), we can find POLAR guarantees the optimal PDR while reducing AEED as much as possible to ensure the QoS.

The area shown in Fig. 9 ranges from 1 to 1.5 square kilometers. As the number of vehicles increases, the trend of PDR shown in Fig. 9(a) is from rising to falling. Compared

with Fig. 8(a), the reason for this trend is that network connectivity is poor when the vehicles are too sparse or too dense. In all cases, POLAR exhibits excellent performance, which can effectively switch routing strategies according to traffic scenarios to improve PDR. At the same time, Fig. 9(b) illustrates that POLAR controls AEED at a lower level, which ensures the QoS in some delay-sensitive application scenarios. Fig. 9(c) presents that PDR of various routing protocols decreases as vehicle speed increases. The trend of PDR is similar to Fig. 8(c). However, compared with Fig. 8(c), the PDR of Fig. 9(c) decreases under various speed settings. This is due to the number of vehicles varies in the same range, and the larger area in Fig. 9 results in the decrease of vehicle density, which leads to the reduction of PDR. Under these circumstances, POLAR can also select the optimal routing strategy to maximize the average PDR. Fig. 9(d) shows the effect of different vehicle speeds on the AEED of various routing protocols in the simulation area of 1 to 1.5 square kilometers. We can find that the AEED of POLAR is also extremely low in different speed ranges.

Table IV shows the performance of POLAR and OLAR. The comparison between POLAR and OLAR is carried out in the same traffic scenarios, which means that the network topologies of nodes are the same. Therefore, if the two schemes choose the same distributed routing strategy, the network performance parameters will be the same due to the same simulation scenarios. Compare with OLAR, we can find that in various traffic scenarios, both PDR and AEED can be optimized through POLAR. The PDRs of both POLAR and OLAR are closed in different traffic scenarios, while in some cases, the AEED of POLAR is significantly better than OLAR. We can analyze Fig. 8 and Fig. 9 to get the reason for this result. As shown in Fig. 8 and Fig. 9, DSR and AODV have similar performance in PDR, whereas the AEED of DSR is far

inferior to AODV. OLAR cannot distinguish AODV from DSR in many cases. In addition, through simulations, we find that DSR is suitable for dense traffic scenarios. Hence, selecting DSR in relatively sparse traffic scenarios will increase AEED. The decision-making model of OLAR has not been optimized by PRA. On the contrary, POLAR can identify the right routing strategy under similar performance which eventually cause POLAR with better performance.

From the comparison between POLAR and traditional routing strategies, we find that POLAR can effectively select the optimal routing strategy according to the change of traffic conditions. As a scalable hybrid routing scheme, POLAR leverages the information of global topology provided by SDVNs to improve routing performance. In addition, the proposed solution enables better utilization of existing routing strategies with avoiding the high latency caused by centralized computing, where it can guarantee the QoS for delay-sensitive application scenarios.

## VI. CONCLUSION

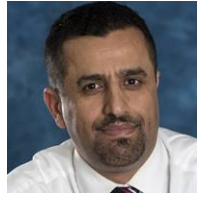
This paper proposes POLAR as an efficient online sequential learning-based adaptive routing scheme for SDVN. POLAR is a universal distributed routing scheme deployed on the control plane, which can select the optimal routing strategy according to the traffic conditions. We present a method for extracting traffic feature data to describe the current traffic condition and label the data. Based on POLAR, the central controller can implement the online learning algorithm which can learn the labeled traffic feature data one-by-one or chunk-by-chunk. Then, the central controller sends the decision-making model to local controllers in real-time for selecting the most appropriate routing strategy. The experimental results show that POLAR can effectively make decisions based on traffic conditions with improved PDR and AEED in various scenarios.

## REFERENCES

- [1] Y. Dai, D. Xu, S. Maharjan, G. Qiao, and Y. Zhang, "Artificial intelligence empowered edge computing and caching for internet of vehicles," *IEEE Wireless Communications*, vol. 26, no. 3, pp. 12–18, June 2019.
- [2] W. Zhuang, Q. Ye, F. Lyu, N. Cheng, and J. Ren, "SDN/NFV-empowered future IoV with enhanced communication, computing, and caching," *Proceedings of the IEEE*, vol. 108, no. 2, pp. 274–291, Feb 2020.
- [3] F. A. Silva, A. Boukerche, T. R. M. B. Silva, E. Cerqueira, L. B. Ruiz, and A. A. F. Loureiro, "Information-driven software-defined vehicular networks: Adapting flexible architecture to various scenarios," *IEEE Vehicular Technology Magazine*, vol. 14, no. 1, pp. 98–107, March 2019.
- [4] S. Correia, A. Boukerche, and R. I. Meneguette, "An architecture for hierarchical software-defined vehicular networks," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 80–86, July 2017.
- [5] O. S. Oubbati, N. Chaib, A. Lakas, P. Lorenz, and A. Rachedi, "UAV-assisted supporting services connectivity in urban VANETs," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3944–3951, April 2019.
- [6] S. Tiennoy and C. Saivichit, "Using a distributed roadside unit for the data dissemination protocol in VANET with the named data architecture," *IEEE Access*, vol. 6, pp. 32 612–32 623, 2018.
- [7] "Geohash homepage," <http://geohash.org/>, accessed July 25, 2019.
- [8] N. Liang, G. Huang, P. Saratchandran, and N. Sundararajan, "A fast and accurate online sequential learning algorithm for feedforward networks," *IEEE Transactions on Neural Networks*, vol. 17, no. 6, pp. 1411–1423, Nov 2006.
- [9] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, Nov 1995, pp. 1942–1948 vol.4.
- [10] A. A. Neghabi, N. Jafari Navimipour, M. Hosseinzadeh, and A. Rezaee, "Load balancing mechanisms in the software defined networks: A systematic and comprehensive review of the literature," *IEEE Access*, vol. 6, pp. 14 159–14 178, 2018.
- [11] F. Guo, H. Zhang, H. Ji, X. Li, and V. C. M. Leung, "An efficient computation offloading management scheme in the densely deployed small cell networks with mobile edge computing," *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, pp. 2651–2664, Dec 2018.
- [12] S. Bitam, A. Mellouk, and S. Zeadally, "Bio-inspired routing algorithms survey for vehicular ad hoc networks," *IEEE Communications Surveys Tutorials*, vol. 17, no. 2, pp. 843–867, Secondquarter 2015.
- [13] Y. Tan and Y. Zhu, "Fireworks algorithm for optimization," in *Advances in Swarm Intelligence*. Springer Berlin Heidelberg, 2010, pp. 355–364.
- [14] H. M. Ali, W. Ejaz, D. C. Lee, and I. M. Khater, "Optimising the power using firework-based evolutionary algorithms for emerging iot applications," *IET Networks*, vol. 8, no. 1, pp. 15–31, 2019.
- [15] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46 – 61, 2014.
- [16] S. K. Goudos, T. V. Yioultis, A. D. Boursianis, K. E. Psannis, and K. Siakavara, "Application of new hybrid jaya grey wolf optimizer to antenna design for 5g communications systems," *IEEE Access*, vol. 7, pp. 71 061–71 071, 2019.
- [17] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.
- [18] D. Rewadkar and D. Doye, "Multi-objective auto-regressive whale optimisation for traffic-aware routing in urban VANET," *IET Information Security*, vol. 12, no. 4, pp. 293–304, 2018.
- [19] G. Li, L. Boukhatem, and J. Wu, "Adaptive quality-of-service-based routing for vehicular ad hoc networks with ant colony optimization," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 4, pp. 3249–3264, April 2017.
- [20] M. Dorigo, G. D. Caro, and L. M. Gambardella, "Ant algorithms for discrete optimization," *Artificial Life*, vol. 5, no. 2, pp. 137–172, 1999.
- [21] F. Li, X. Song, H. Chen, X. Li, and Y. Wang, "Hierarchical routing for vehicular ad hoc networks via reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1852–1865, Feb 2019.
- [22] C. Chen, L. Liu, T. Qiu, K. Yang, F. Gong, and H. Song, "ASGR: An artificial spider-web-based geographic routing in heterogeneous vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1604–1620, May 2019.
- [23] Z. He, J. Cao, and X. Liu, "SDVN: enabling rapid network innovation for heterogeneous vehicular communication," *IEEE Network*, vol. 30, no. 4, pp. 10–15, July 2016.
- [24] Y. Gao, Z. Zhang, D. Zhao, Y. Zhang, and T. Luo, "A hierarchical routing scheme with load balancing in software defined vehicular ad hoc networks," *IEEE Access*, vol. 6, pp. 73 774–73 785, 2018.
- [25] C. Yu, J. Lan, Z. Guo, and Y. Hu, "DROM: Optimizing the routing in software-defined networks with deep reinforcement learning," *IEEE Access*, vol. 6, pp. 64 533–64 539, 2018.
- [26] D. B. Johnson and D. A. Maltz, *Dynamic Source Routing in Ad Hoc Wireless Networks*. Boston, MA: Springer US, 1996, pp. 153–181.
- [27] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proceedings WMCSA'99. Second IEEE Workshop on Mobile Computing Systems and Applications*, Feb 1999, pp. 90–100.
- [28] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*, vol. 2, July 2004, pp. 985–990 vol.2.
- [29] "ns-3 homepage," <https://www.nsnam.org>, accessed July 25, 2019.
- [30] "OpenStreetMap homepage," <https://www.openstreetmap.org>, accessed July 25, 2019.
- [31] "SUMO homepage," <http://sumo.sourceforge.net/>, accessed July 25, 2019.
- [32] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized link state routing protocol for ad hoc networks," in *Proceedings. IEEE International Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century.*, Dec 2001, pp. 62–68.
- [33] B. Karp and H. T. Kung, "Gpsr: Greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '00. ACM, 2000, pp. 243–254.



- [34] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *Proceedings of the Conference on Communications Architectures, Protocols and Applications*, ser. SIGCOMM '94. ACM, 1994, pp. 234–244.
- [35] A. Silva, N. Reza, and A. Oliveira, "Improvement and performance evaluation of GPSR-based routing techniques for vehicular ad hoc networks," *IEEE Access*, vol. 7, pp. 21 722–21 733, 2019.
- [36] L. Zhao, X. Li, B. Gu, Z. Zhou, S. Mumtaz, V. Frasca, H. Gacanin, M. I. Ashraf, J. Rodriguez, M. Yang, and S. Al-Rubaye, "Vehicular communications: Standardization and open issues," *IEEE Communications Standards Magazine*, vol. 2, no. 4, pp. 74–80, December 2018.
- [37] L. Zhao, W. Zhao, A. Al-Dubai, and G. Min, "A novel adaptive routing and switching scheme for software-defined vehicular networks," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, May 2019, pp. 1–6.



**Ahmed Y. Al-Dubai [SM]** received the Ph.D. degree in computing from the University of Glasgow, Glasgow, U.K., in 2004. In 2004, he joined the University of West London, London, U.K. In 2005, he joined Edinburgh Napier University, Edinburgh, U.K., where he became a Professor and the Programme Leader of the Post-Graduate Research degrees with the School of Computing. He is currently the Head of the Networks Research Group. He has been published in world leading journals and in prestigious international conferences. He has also been involved with research in the area of group communication algorithms, smart spaces, and high-performance networks. He is a Fellow of the Higher Academy, U.K. He was a recipient of the several academic awards and recognitions, and a member of several Editorial Boards of scholarly journals. He has served as a Guest Editor for more than 20 special issues in scholarly journals and chaired and co-chaired more than 30 international conferences/workshops.



**Liang Zhao [M]** is an associate professor at Shenyang Aerospace University, China. He received his Ph.D. degree from the School of Computing at Edinburgh Napier University in 2011. Before joining Shenyang Aerospace University, he worked as associate senior researcher in Hitachi (China) Research and Development Corporation from 2012 to 2014. His research interests include ITS, VANET, WMN and SDN. He has published more than 100 papers. He served as the Chair of several international conferences and workshops, including 2019 IEEE

IUCC (Program Co-Chair), 2019 IEEE Scalcom (Poster/Demo Co-Chair), IoT-Smart-2015 (Program Co-Chair), 2019 AI-driven Network Workshop (Program Co-Chair), and NGDN workshop (founder). He is/has been a guest editor of IEEE Transactions on Network Science and Engineering, Springer Journal of Computing, and Wiley Internet Technology Letters. He was the recipient of the Best Paper Awards at IEEE IUCC 2015 and ACM MOMM 2013. He is the corresponding author of this paper.



**Geyong Min** is a Professor of High Performance Computing and Networking in the Department of Computer Science within the College of Engineering, Mathematics and Physical Sciences at the University of Exeter, United Kingdom. He received the PhD degree in Computing Science from the University of Glasgow, United Kingdom, in 2003, and the B.Sc. degree in Computer Science from Huazhong University of Science and Technology, China, in 1995. His research interests include Computer Networks, Wireless Communications, Parallel and Distributed Computing, Ubiquitous Computing, Multimedia Systems, Modelling and Performance Engineering.



**Weiliang Zhao** received the M.S. degree in technology of computer application from Shenyang Aerospace University, China. He is currently pursuing the Ph.D. degree with Northeastern University. His research interests mainly include NoC, VANETS, SDN, and machine learning.



**Albert Y. Zomaya [F]** is a Chair Professor and director of the Centre for Distributed and High Performance Computing at the University of Sydney. He has published more than 600 scientific papers and is an author, co-author, or editor of more than 20 books. He is the Editor-in-Chief of IEEE Transactions on Sustainable Computing and ACM Computing Surveys and serves as an Associate Editor for several leading journals. He is a Fellow of IEEE, AAAS, IET, and member of Academia Europaea.



**Ammar Hawbani** received the B.S., M.S. and Ph.D. degrees in Computer Software and Theory from the University of Science and Technology of China (USTC), Hefei, China, in 2009, 2012 and 2016, respectively. Currently, he is a postdoctoral researcher in the School of Computer Science and Technology at USTC. His research interests mainly in WSN and WBAN. His postal address is 96 JinZhai Road, Baohe District, Hefei, China.



**Changqing Gong** is a Professor at Shenyang Aerospace University, China. He received his Ph.D. degree in Computer Application Technology from the School of Information Science and Engineering at Northeastern University, China, in 2006. His research interests include network security, quantum machine learning and quantum computing. His postal address is 37 Daoyi South Avenue, Shenbei New Area, Shenyang, China.