Federated learning based Task Orchestration Scheme using Intelligent Vehicular Edge Networks

Nishu Bansal*, Shilpi Mittal†, Rasmeet S. Bali‡, Neeraj Kumar (SMIEEE)§,
Joel Rodrigues (FIEEE)¶, Liang Zhao (SMIEEE)‖
*‡ Computer Science & Engineering, Chandigarh University, India
† University Institute of Computing, Chandigarh University, India
§ Department of Computer Science Engineering, Thapar University, Patiala, Punjab
¶ China University of Petroleum (East China), Qingdao 266555, China,
and Instituto de Telecomunicações, 6201-001 Covilhã, Portugal
‖ Shenyang Aerospace University, China
(e-mail: nishubl@gmail.com, g.shilpi84@gmail.com, rasmeetsbali@gmail.com, neeraj.kumar@thapar.edu,
joeljr@ieee.org,lzhao@sau.edu.cn)

ICC 2023 - IEEE International Conference on Communications | 978-1-5386-7462-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICC45041.2023.10279392

***Abstract*—Vehicular Edge Computing (VEC) is gradually evolving into one of the most prevalent paradigms for vehicular computation. This is due to its ability for effectively handling the tasks of varied complexity. VEC based task orchestration has therefore emerged as an exciting research domain. A large number of task orchestration schemes have been proposed that exploit its technical capabilities. However, identifying the most appropriate vehicles for such edges still remain a challenge. In this work, we propose an intelligence based Task Orchestration Scheme integrated with Vehicular Cloud Edge Networks that uses Federated learning (FL) for forming vehicular edges. FL is a privacy preserving technique with no data being shared centrally. This scheme uses characteristics of vehicles such as computational capacity and their starting as well as ending point for creating the edges. Obtained results depict the improved performance of this scheme as compared to conventional schemes.**

Keywords*—*Intelligent Transportation System, Vehicular Edge Networks, Federated Learning, Task Orchestration.

I. Introduction

The rapid evolution of vehicular technology in the past decade is contributing to an increasing number of applications being deployed through moving vehicles. The advent of smart vehicles is expected to further accelerate this development due to its relatively higher computing requirements. An estimation of next five years states that for in-vehicles data generation capacity for single day could be about 30 terabytes [1]. The in-vehicle computational capacity can help to meet these future challenges. This can contribute more to various paradigms such as Fog Computing, Cloud Computing and Edge Computing being integrated with vehicular networks and gives the computation solutions as Vehicular Fog Computing (VFC) [2], Vehicular Cloud Computing (VCC) [3] and Vehicular Edge Computing (VEC) [4], [5].

In VEC, Vehicles communicating with its nearest located edge device and the computing resources are distributed at local edge networks [6], [7] therefore there is minimal deployment cost incurred. So, it is considered as the preferred technique to use in-vehicle computation capacity for task orchestration in vehicular networks [8]. Here, the task is divided into multiple connected vehicles inside vehicular edge network for task execution. In VEC, computation can be taken at local road side units (RSUs) or via dynamic moving vehicles. The task orchestrated to vehicular edge network in both the cases, faces big challenge due to increased number of nodes and lesser number of communication resources availability to make an efficient vehicular edge network [9].

To solve the above issue and to enhance the performance of VEC, Intelligence based edge creation can be helpful. Intelligence may be applied in terms of Artificial Intelligence (AI) [10], Machine Learning (ML) [11] or Deep Learning (DL) [12], [13]. AI uses human Intelligence to solve the task whereas ML is the subset of AI which leverages algorithm for data analysis, learning, forecasting and solving the tasks. DL is further subset of ML, which is applied on large data sets to filter the data through layers for classification and prediction to perform the task solving same as the human brains [14], [15]. Federated Learning (FL) as collaborative ML [16] and decentralized technique has been used by various service providers [17] and plays a major role in support of privacy-preserving applications which uses training of distributed data at edges [18]. FL has found limited usage in vehicular communication on account of centralized processing being preferred as compared to distributed. This also results in limited data analysis being done for vehicular infrastructure management. However, FL can provide an effective outcome when merged with VEC. In this paper we are extending our previous work by incorporating intelligence in the form of FL over VEC.

Integrated Vehicular Edge Computing and Federated learning based task orchestration is proposed in this model. The utmost goal of FL is to train a shared global model to further use in distributed devices [19]. As the structure shown in figure 2, local training models are available with edge controllers (ECs). The local training updates computed at edge controllers will be transferred to federated server and global model weights will be communicated to local models of ECs of distributed devices. Firstly, the global training model is downloaded to local devices and then training of

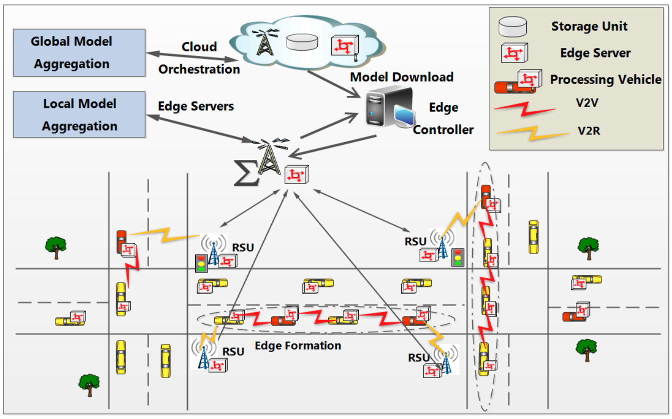

Fig. 1: Task Orchestration using Vehicular Edge Networks integrated with Federated Learning

local models takes place. Distributed edge devices train their model using the vehicular data available locally. The updates drawn from local model training will then aggregate the global server model. Based on the regular updations from the local models, the global model will get improved. The improved central server model is finally shared with various local edge devices to better train their models. This structure of FL brings the advantage that only the updates to the global model or downloading the updated model from the server is permitted and no data packet communication to server takes place [20].

Task orchestration for processing is one of the critical issues to enable computing in vehicular networks effectively. Efficiently executing task orchestration procedures reduce delay or latency and improves the efficiency of the networks for different domains. To make an efficient task orchestration, proposed framework (iVEC) suggests the traffic light points (junctions) where vehicles are approaching as the edge creation points. Edge controllers are installed at each junction which are further connected to Edge Server (ES) to make a centralized repository of vehicular data. This data is used to train the local models at EC while preserving the privacy of data. The sequence of paper explaining this model is as follows: Section 2 describes the proposed model for an intelligent task orchestration scheme. Section 3 explains the simulation environment and result analysis. The last section 4 concludes the work done.

II. System Model

A moving vehicle on road, when comes to stationary state at any traffic junction (red traffic signal) gets registered at EC installed at junctions. The architecture of the same is explained using figure 1. A vehicle identification V_{ID} will be assigned to each vehicle reaching the Edge Controller. The vehicles will communicate their start point and destination to the EC which is further stored at ES corresponding to different ECs. Path-IDs to be followed by vehicles will be assigned as per described in [8]. Local models present at EC are getting trained with the vehicular information. The updates after training are aggregated to global model. This centralized global model is communicated to local devices for better training of local models which helps in edge forming. Multiple edges at each junction in each EC will be formed. The final Global model in federated scenario is helping to decide the best vehicular edge after all the updates. The task orchestration on best selected edge takes place where it is distributed among different edge vehicles for computation. Thus the vehicles in the edges will actually perform the tasks processing using their computational capacity. The process of the same is divided into three phases.

A. Intelligent Edge Creation Phase

A stationary state vehicle at a junction gets a vehicular identification V_{ID}. This is checked for its already existence on ES before assigning V_{ID} to a new vehicle reaching at the junction. If V_{ID} is already present on ES then only the vehicle information will be transferred to current EC followed by storage at ES, otherwise the new V_{ID} will get generated as explained in algorithm 1. For registered vehicles at EC, based on local vehicular data, the model is trained locally and the intermediate updates are communicated to Global model. The Global model gets improved with every update. FL process is used to train the Shared Global model of distributed ECs. The shared model is downloaded from central server to local ECs. The recurring training model updates, after training are shared with the global server and multiple edges of vehicles may be created at each EC. The local data considered for edge creation includes number of junctions, number of lanes at a junction, average traffic density at lanes, Path-ID length etc. The goal in federated-edge creation process is to minimize the following function:

$$\min_{w} F(w), where\ F(w) := \sum_{m=1}^{v} p_m F_m(w) \tag{1}$$

Here, v is the total number of vehicles at a given road junction. p_m is the impact of m^{th} vehicle, $p_m \geq 0$ and $\sum m p_m = 1$. F_m is the local objective function for the m^{th} vehicle given as:

$$F_m(w) = 1/s_m \sum_{j_m=1}^{n_m} f_{j_m}(w; x_{j_m}, y_{j_m}) \tag{2}$$

Where s_m is the number of samples taken locally. p_m denotes the impact of each vehicle with $p_m = (1/s)$ or $p_m = (s_m/s)$ and $s = \sum m n_m$ is the total number of samples. Let

$$veh_v = veh_1,\ veh_2,\ veh_3, ...,\ veh_v \tag{3}$$

represents the v number of vehicles registered on EC. The FedAvg algorithm of FL is used to compute the average of model updates which are received from the EC in every round. The local training of model takes place with local vehicle data. v vehicles with local dataset d_v become the part of local training. Weight updates are communicated to edge server and Global server. FL training takes place to minimize the F_m function and developing the final Global model G_m given as:

$$G_m = \frac{\sum_{j=1}^{v} v k_j w_j}{J} \tag{4}$$

which is computed as sum of all the weights received from local EC divided by total number of considered road junctions J. Stochastic Gradient Decent (SGD) optimizer for federated

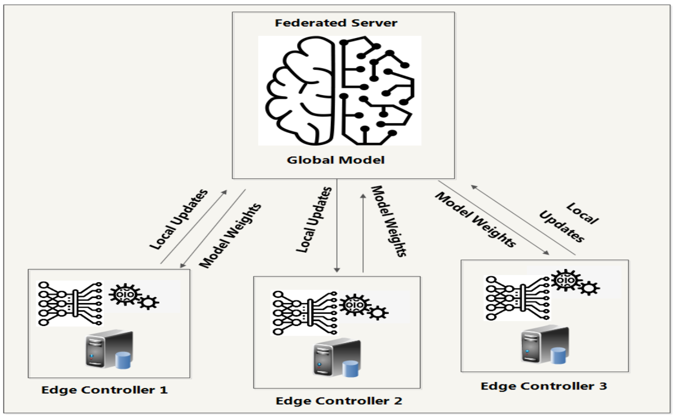

Fig. 2: Communication of Edge Controllers with Federated Server

learning is also used while training the local models and is computed as:

$$g_v = \triangle f_v(w) \tag{5}$$

After applying the SGD, the updated weight is computed as follows and communicated to Global server.

$$w^{t+1} \leftarrow w^t - \Lambda \sum_{j=1}^{v} \frac{k_j}{J} g_v \tag{6}$$

where Λ is learning rate of EC.

Based on the local vehicle information provided at EC and after going through the federated learning, an edge of vehicles is created for a road junction as discussed in algorithm 1. As per algorithm, firstly the vehicle is registered when in stationary state to corresponding EC placed at road junction. Information about each vehicle at respective road junction is passed to ES. At each EC, the weight w_0 is first initializes to 0. Local model at each EC is trained with vehicle information and the edge server is aggregated with the training updates received as weighted sum, using which Global server model is also aggregated. Vehicle with similar characteristics become the part of the edge. Multiple edges are created at respective junctions.

B. Best Edge Selection Phase

Each junction will have the corresponding multiple edges of vehicles as per the vehicle information collected. Multiple edges are due to different groups of vehicles containing similar characteristics. The edge information of all the edges created at different ECs, is communicated to Global server at cloud and the best edge as per task computation requirement is selected as per algorithm 2. The parameters considered for the same are edge stability and edge duration. Moreover the edge with maximum vehicles as well as on maximum busy path is preferred.

$$f \leftarrow Bfit(TaskID, EdgeID, Longest_{P}athID) \tag{7}$$

Multiple edges created in phase 1 are taken as input for phase 2 as shown in algorithm 2. A pathID with maximum length is computed. Vehicles going through the longest pathID are preferred to be part of considered EdgeID. A best fit match of task duration and edge duration gives the way to decide the best edge f for computation.

Algorithm 1 **Edge Creation Algorithm**

Inputs:
1. Vehicles V at road junction with vehicle detail, start point V_a, destination point V_b.
2. Edge Controllers E_x at each road junction.
3. Edge Server E_s equipped with vehicle detail.
Outputs:
1. Global Model G_m.
2. Edge of vehicles E_v for each E_x.

$State(V_i) \leftarrow OFF$;
$w \leftarrow 0$;
for each E_x **do**
  for each V_i **do**
    $E_s \leftarrow V_{ID}$;
    if (V_{ID}) **then**
      Transfer(V_{ID}, V_a, V_b);
    else
      get V_{ID};
    end if
  end for
end for
for each E_x **do**
  $w_v^{t+1} \leftarrow w_v^t - \Lambda \triangle F_m(w_v^t) \leftarrow Train(V_a, V_b)$;
  $G_m \leftarrow Update(w_v^{t+1})$;
  Insert(E_v, G_m);
end for
End

C. Task Orchestration

The task processing is assigned to the best edge selected as per its interval and stability. The selected edge has the computational capabilities required for task completion. After the edge initialization, the vehicles will start moving and processing of the task allocated will take place. The edge can add more vehicles at next junction or any current vehicle may leave the edge there. Whenever a vehicle will leave the edge, it will communicate the task processing status to the edge server. The same will be updated to cloud server. A leave message from vehicle to server will be provided so that the edge reorganization should take place at next road junction before next task orchestration.

III. Simulation Environment and Result Analysis

To analyse the performance of proposed scheme, a real time network scenario of Chandigarh City has been considered for implementation. This road network scenario has been driven from OpenStreetMap(OSM). The OSM map has given route information between various end points in city along with the number of junctions lying on these routes. Further we use SUMO to generate data about the moving vehicles on

Algorithm 2 **Best Edge Selection Algorithm**

Inputs:

1. Vehicular edges E_v created at different ECs as per Algorithm 1.
2. PathID P of vehicular edges.
3. TaskID to be computed.

Outputs:

1. Longest PathID P_{ID}.
2. Best edge f for task processing out of available set of edges.

Begin
i=0, $P_{ID} = P_i$;
for each i in P **do**
    if ($P_{i+1} > P_{ID}$) **then**
        P_{ID}=P_{i+1};
    end if
    i++;
end for
j=0;
for each j in E_v **do**
    $f \leftarrow Bfit(TaskID, E_v, P_{ID})$;
    j++;
end for
End

these routes. The framework is implemented on open source federated Tensor Flow which is divided into Federated core and Federated Learning. FL APIs ($tff.learning$) are used for this purpose. The whole input about road scenarios and vehicle information further tested with proposed algorithms using NS3 Simulator. In our experiments, we have assumed some of the simulation parameters which are described in Table I.

Table I shows that vehicles arrive at random interval

TABLE I: Evaluation Parameters

Evaluation Parameters	
Parameters	**Values**
Vehicle Density at Junction	[10,500]
DSRC Data Rate	27Mbps
Arrival rate	Uniform:[0,1]/vehicles/s
Average number of edge Vehicles	100
Interval between BSM	1 second
Bandwidth(IEEE802.11P)	10MHz
Data Set task logs	1000
Simulation Time	360 seconds
Number of Routes	4
Average Number of Junctions	[2,15]
Route Distance	[6,23]KM

of [0,1] vehicles per second at each junction and the data rate capacity of each vehicle is 27Mbps under the channel capacity of 10MHz. Data set of 1000 task logs is taken. We also considered that about 10 - 500 number of vehicles are present at each junction at every second. The evaluation of federated model is performed using vehicular data training of 500 vehicles. We assume that there will be 100 average number of vehicles available for being a part of edge. These edge vehicles communicate with each other through V2V communication. The information has been calculated for at least 4 routes based on the distance between source and destination, route length and number of junctions between the route as described in [8] scheme.

TABLE II: Route Description

Route Description				
Route ID	**Starting Point**	**End Point**	**Distance(Km)**	**No. of Junctions**
Route 1	Sec 8	Sec 21	3.9	2
Route 2	Sec 34	Sec 9	4.8	3
Route 3	Sec 4	Sec 45	8	6
Route 4	Sec 5	Sec 87	15	11

To evaluate the performance of proposed scheme, we have considered different routes varying from one point to another in the area of Chandigarh City. The route description has been shown in the table II. It shows that the routes cover about to $3.9km$ distance to $15km$ distance along with the varying number of junctions in between the routes. The number of junctions lie between 2 to 11 depending upon the distance covered corresponding to the starting and ending point of routes. Based on these four routes, we have evaluated different parameters such as throughput, latency, edge density, edge duration and edge stability in the proposed networking scheme. Also based on these parameters, the Success Rate has been calculated to analyse the proposed scheme performance.

Figure 3 shows the variation in the density i.e the number of vehicles per Km in the edge network with respect to Arrival Rate of vehicles per second at the junction on the route depending upon the route ID as shown in table II. The graph defines that the density is increasing constantly as the number of vehicles are arriving in the vehicular network and accepting the request of being edge member due to presence on the same route. Route 4 has more number of junctions lying between the route therefore it has higher rate of vehicle density. Since route 1 has only 2 junctions where the possibility of edge creation is limited with less number of vehicles therefore edge density is lowest at this route. This impact is due to availability of junctions where vehicles become a part of edge network.

Figure 4 and Figure 5 illustrate the impact of throughput and latency with respect to increasing nodes in the network at different routes. It shows that the throughput is increasing constantly and latency is decreasing and remain stable at a threshold value. This is due to the increasing number of vehicles in the edge network and are increasing throughput but these vehicles are communicating with each other through out the travelling on route till they do not leave the edge network, hence latency is reduced.

Figure 6 and Figure 7 are showing the impact on edge stability and its time duration for executing task with respect to number of vehicles arriving in the network. It shows that edge will be stable for long route than the short route as it has more probability to execute task by adding more vehicles at junctions during the route. Also the edge duration for task execution will be higher at long route than the short route where number of junctions are limited. This is due the fact that more number of junctions will provide more availability of

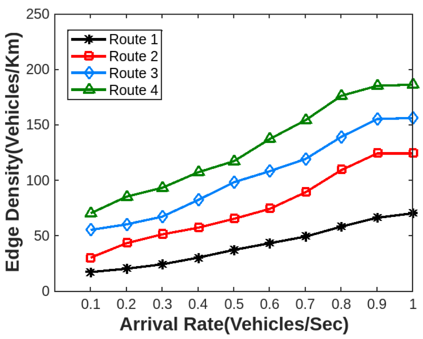

Fig. 3: Edge Density w.r.t Arrival Rate(Veh/Sec)

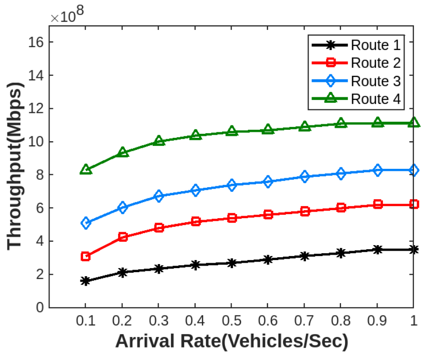

Fig. 4: Throughput w.r.t Arrival Rate(Veh/Sec)

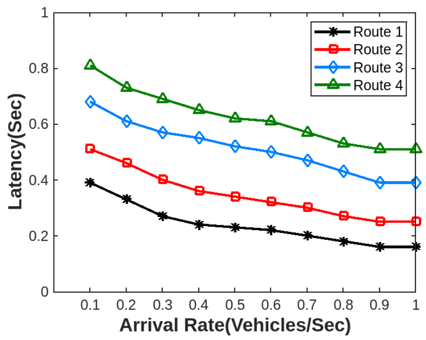

Fig. 5: Delay w.r.t Arrival Rate(Veh/Sec)

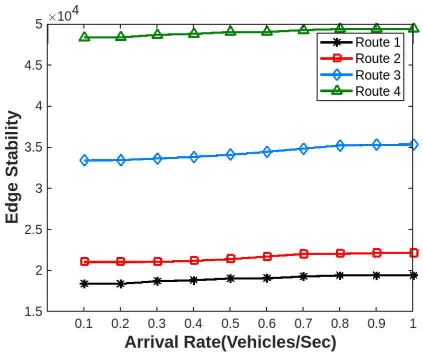

Fig. 6: Edge Stability w.r.t Arrival Rate(Veh/Sec)

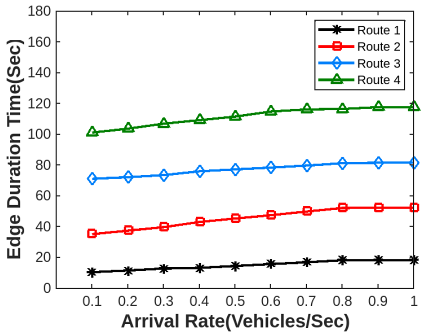

Fig. 7: Edge Duration(Sec) w.r.t Arrival Rate(Veh/Sec)

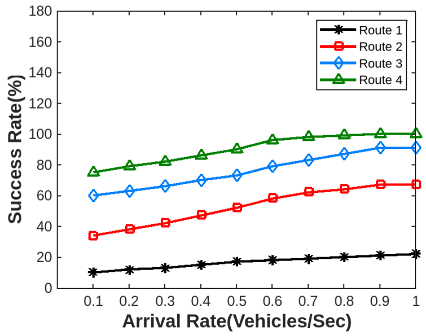

Fig. 8: Success Rate(%) w.r.t Arrival Rate(Veh/Sec)

vehicles and hence will improve edge density which is required for more distribution of task among vehicles.

Figure 8 illustrate the performance of proposed scheme in terms of success rate of the proposed algorithm. The graph depicts that the value of the success rate is increasing and remain constant at a level of number of vehicles in the edge network. This is because the selected edge network has higher probability to remain in the network based on intelligent prediction of edge stability in the proposed scheme. Thereby, higher edge network stability higher execution rate with minimum delay and maximum throughput. We have compared the proposed scheme iVEC with existing DTOF [8] scheme to evaluate the performance of edge stability and edge duration time because in DTOF there is no intelligence based algorithm being used for edge creation and maintenance. The proposed method iVEC is extending the DTOF functionality intelligently. Figure 9 and figure 10 shows the comparison about edge stability and edge duration time with respect to arrival rate of vehicles per second in the network. Figure 9 shows that the edge duration time will be increased as the number of vehicles are coming into network for the proposed scheme whereas in DTOF it is increasing with small value at constant rate. This is due to the fact that increasing number of vehicles after intelligent edge selection will remain there for task orchestration. Figure 10 depicts that the edge stability for the proposed scheme will be increased due to more probability

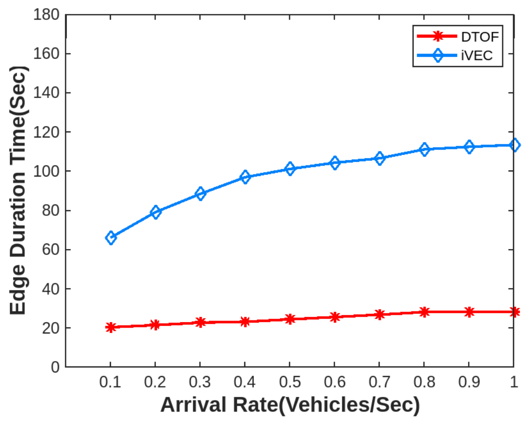

Fig. 9: Comparative Edge Duration Time(Sec) w.r.t Arrival Rate(Veh/Sec)

of vehicles availability to join in the edge network for the proposed scheme. However, in DTOF it is also increasing but iVEC shows better performance than it.

The evaluated results exhibit that intelligent network creation for task orchestration is performing well on the basis of efficient and reliable communication. Also, since the edge formation is efficient, the edge network duration remains stable

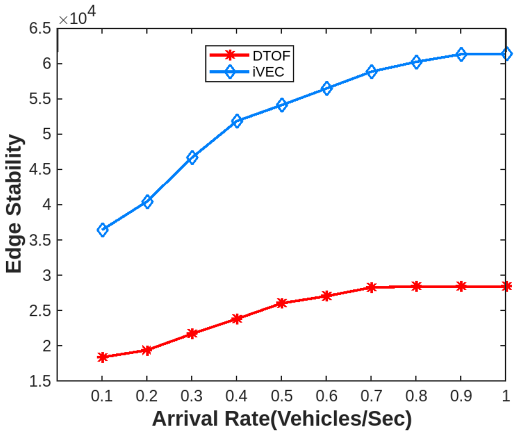

Fig. 10: Comparative Edge Stability w.r.t Arrival Rate(Veh/Sec)

resulting in higher throughput and minimum latency.

IV. Conclusion

The proposed task orchestration scheme iVEC has been designed to provide the resources for task orchestration with intelligent federated learning method introduced in vehicular edge computing paradigm. The proposed scheme identifies the most appropriate edge network of smart vehicles that provide efficient number of vehicles for task execution. The performance of proposed scheme is evaluated using various parameters and the result outperforms the superiority of the proposed iVEC over DTOF. In future, there will be more work extended to the development of applications based on iVEC in vehicular networks. We plan to empower the iVEC with an algorithms to enhace the performance in creation and managing the edge and task orchestration processes.

Acknowledgement

This work is partially funded by FCT/MCTES through national funds and when applicable co-funded EU funds under the Project UIDB/50008/2020; and by Brazilian National Council for Scientific and Technological Development - CNPq, via Grant No. 313036/2020-9.

References

[1] X. Huang, R. Yu, J. Kang, and Y. Zhang, "Distributed reputation management for secure and efficient vehicular edge computing and networks," *IEEE Access*, vol. 5, pp. 25 408–25 420, 2017.

[2] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 3860–3873, 2016.

[3] E. Lee, E.-K. Lee, M. Gerla, and S. Y. Oh, "Vehicular cloud networking: architecture and design principles," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 148–155, 2014.

[4] S. Raza, S. Wang, M. Ahmed, and M. R. Anwar, "A survey on vehicular edge computing: Architecture, applications, technical issues, and future directions," *Wireless Communications and Mobile Computing*, vol. 2019, 2019.

[5] L. Liu, C. Chen, Q. Pei, S. Maharjan, and Y. Zhang, "Vehicular edge computing and networking: A survey," *Mobile networks and applications*, vol. 26, no. 3, pp. 1145–1168, 2021.

[6] D. Garg, N. Garg, R. S. Bali, and S. Rawat, "Sdvn-based smart data dissemination model for high-speed road networks," in *Software Defined Internet of Everything*. Springer, 2022, pp. 259–270.

[7] S. Mittal, D. Garg, R. S. Bali, and G. S. Aujla, "Edge computing based resource supplementation for software defined vehicular networks," in *2022 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2022, pp. 1693–1698.

[8] S. Mittal, R. K. Dudeja, R. S. Bali, and G. S. Aujla, "A distributed task orchestration scheme in collaborative vehicular cloud edge networks," *Computing*, pp. 1–25, 2022.

[9] N. Cha, C. Wu, T. Yoshinaga, Y. Ji, and K.-L. A. Yau, "Virtual edge: Exploring computation offloading in collaborative vehicular edge computing," *IEEE Access*, vol. 9, pp. 37 739–37 751, 2021.

[10] W. Tong, A. Hussain, W. X. Bo, and S. Maharjan, "Artificial intelligence for vehicle-to-everything: A survey," *IEEE Access*, vol. 7, pp. 10 823–10 843, 2019.

[11] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[12] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[13] D. Jakhar and I. Kaur, "Artificial intelligence, machine learning and deep learning: definitions and differences," *Clinical and experimental dermatology*, vol. 45, no. 1, pp. 131–132, 2020.

[14] N. Bansal, R. S. Bali, K. Jakhar, M. S. Obaidat, N. Kumar, S. Tanwark, and J. J. Rodrigues, "Htfm: Hybrid traffic-flow forecasting model for intelligent vehicular ad hoc networks," in *ICC 2021-IEEE International Conference on Communications*. IEEE, 2021, pp. 1–6.

[15] N. Bansal and M. Dutta, "Article: Retina vessels detection algorithm for biomedical symptoms diagnosis," *International Journal of Computer Applications*, vol. 71, no. 20, pp. 41–46, June 2013, full text available.

[16] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowledge-Based Systems*, vol. 216, p. 106775, 2021.

[17] L. Li, Y. Fan, M. Tse, and K.-Y. Lin, "A review of applications in federated learning," *Computers & Industrial Engineering*, vol. 149, p. 106854, 2020.

[18] X. Yin, Y. Zhu, and J. Hu, "A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions," *ACM Computing Surveys (CSUR)*, vol. 54, no. 6, pp. 1–36, 2021.

[19] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[20] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečnỳ, S. Mazzocchi, B. McMahan *et al.*, "Towards federated learning at scale: System design," *Proceedings of Machine Learning and Systems*, vol. 1, pp. 374–388, 2019.