VeSoNet: Traffic-Aware Content Caching for Vehicular Social Networks Using Deep Reinforcement Learning

Nyothiri Aung, Sahraoui Dhelim, Liming Chen, *Senior Member, IEEE*, Abderrahmane Lakas, *Senior Member, IEEE*, Wenyin Zhang, Huansheng Ning, *Senior Member, IEEE*, Souleyman Chaib, and Mohand Tahar Kechadi

***Abstract*— Vehicular social networking is an emerging application of the Internet of Vehicles (IoV) which aims to achieve seamless integration of vehicular networks and social networks. However, the unique characteristics of vehicular networks, such as high mobility and frequent communication interruptions, make content delivery to end-users under strict delay constraints extremely challenging. In this paper, we propose a social-aware vehicular edge computing architecture that solves the content delivery problem by using some vehicles in the network as edge servers that can store and stream popular content to close-by end-users. The proposed architecture includes three main components: 1) the proposed social-aware graph pruning search algorithm computes and assigns the vehicles to the shortest path with the most relevant vehicular content providers. 2) the proposed traffic-aware content recommendation scheme recommends relevant content according to its social context. This scheme uses graph embeddings in which the vehicles are represented by a set of low-dimension vectors (vehicle2vec) to store information about previously consumed content. Finally, we propose a deep reinforcement learning (DRL) method to optimise the content provider vehicle distribution across the network. The results obtained from a real-world traffic simulation show the effectiveness and robustness of the proposed system when compared to the state-of-the-art baselines.**

***Index Terms*— IoV, vehicular social networks, path planning, social computing, vehicular edge computing, content caching, social-aware.**

Manuscript received 28 December 2021; revised 26 July 2022 and 20 January 2023; accepted 24 February 2023. Date of publication 14 March 2023; date of current version 2 August 2023. This work was supported in part by the Insight Centre for Data Analytics funded by the Science Foundation Ireland under Grant 12/RC/2289 P2; in part by the CONSUS Project funded by the SFI Strategic Partnerships Programme under Grant 16/SPP/3296; and in part by Origin Enterprises Plc., National Natural Science Foundation of China, under Grant 61872038. The Associate Editor for this article was W. Wei. *(Nyothiri Aung and Sahraoui Dhelim are co-first authors.) (Corresponding author: Sahraoui Dhelim.)*

Nyothiri Aung, Sahraoui Dhelim, and Mohand Tahar Kechadi are with the School of Computer Science, University College Dublin, Dublin 4, D04 V1W8 Ireland (e-mail: sahraoui.dhelim@ucd.ie).

Liming Chen is with the School of Computing, Ulster University, BT37 0QB Newtownabbey, U.K.

Abderrahmane Lakas is with the College of Information Technology (CIT), United Arab Emirates University, Al Ain, United Arab Emirates.

Wenyin Zhang is with the School of Information Science and Engineering, Linyi University, Shandong 276000, China.

Huansheng Ning is with the School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China.

Souleyman Chaib is with LabRi Laboratory, Ecole Supèrieure en Informatique, Sidi Bel Abbès 22000, Algeria.

Digital Object Identifier 10.1109/TITS.2023.3250320

I. Introduction

With the emergence of the Internet of Vehicles (IoV) as a new networking paradigm that interconnects vehicles with the ubiquitous Internet of Things (IoT) network and the increasing adoption of the 5G network in many countries, the vision of the intelligent transportation system (ITS) is closer to realisation than ever. The IoV network is expected to enhance many applications and offers a wide range of services, ranging from essential emergency services to entertainment service applications. There are currently more than 1.4 billion vehicles worldwide, and it is expected to reach three billion in 2037 [1], which will worsen the existing traffic congestion problem. As more and more people spend hours in traffic congestion, they turn to social media and other entertainment services to spend the waiting time [2]. The IoV can offer an alternative to connect the users with the Internet and seamlessly interconnect their existing social networks to a vehicular social networking model that brings social content near to passengers and reduce the expensive access to the 4G/5G networks.

One of the most challenging problems in a vehicular social networking model is the difficulty of seamlessly accessing social network content without interruptions and delivery delays. In vehicular networks, the content can be delivered through Vehicle to Infrastructure (V2I) communication with the Roadside Units (RSU) connected to the Internet or through cellular base stations using 4G or 5G networks. The former is reasonably cheap and convenient communication but suffers from difficult access and sparse RSUs. Vehicles must rely on Vehicle-to-Vehicle (V2V) communications to overcome sparse RSUs. The latter has the advantage of wide coverage and instant access, but it has expensive communications [3]. The intuitive approach is to store the content of social networks on a cloud server, and the vehicles can access it through V2I communications or by downloading it using 4G or 5G cellular networks. Nonetheless, V2I communications are not appropriate for live streaming due to the high speed of vehicles and the frequent disconnections between vehicles and RSUs [4]. On the other hand, the vehicle-to-base station communications are more stable compared to V2I communications but not suitable for downloading large files due to the high costs of those network usage [5].

We propose a traffic-aware Vehicular Social Network (VeSoNet) content caching architecture that exploits the vehicular edge computing paradigm to deal with the above limitations. VeSoNet stores the most popular social content in vehicles and brings it near the end-users for future use. In the following, we summarise our contributions:

- We proposed a social-aware hybrid content distribution scheme, where only the popular data content is replicated and stored in vehicles to minimize the access time. The content provider vehicles architecture is based on the vehicular edge computing paradigm.
- We developed a social-aware graph pruning search algorithm that computes and assigns the content consumer vehicles to the shortest path with the most relevant content and uses a Deep Reinforcement Learning DRL model to optimize content distribution across the network.
- We developed a traffic-aware content recommendation approach based on graph embeddings, called vehicle2vec, where vehicles are represented by a set of low-dimensional vectors based on their previously consumed content.

The rest of the paper is organized as follows: In Section II, we reviewed the literature on content caching and delivery using vehicular edge computing paradigm and social vehicular networks. In Section III, we presented the main components of the proposed vehicular network architecture. Section IV details the system modelling of the proposed system. While in Section V, we presented the experimental evaluation and discussed the obtained results. Finally, we concluded and gave some future work in Section VI.

II. Related Work

Various studies have proposed different system architectures for content delivery in vehicular networks. Dzyiauddin et al. [11] surveyed computational offloading, content delivery and caching in vehicular edge computing, including the architecture, communication layers and applications of vehicular edge computing for content delivery and caching. Zhang et al. [10] introduced a social-aware mobile edge computing architecture for content caching. They employed the DRL model and proposed a method to take advantage of the relationships among vehicles and RSUs to perform content dissemination with diverse vehicular social characteristics for urban environments. Moreover, they extended their model by introducing digital twin technology to map the edge caching system into cyberspace [12]. They used the vehicular cloud to coordinate the correlations of the cached content among multiple vehicles, then employed the deep-learning approach for route selection, taking into account the social context, the vehicular cloud formation and cache resource allocation. Similarly, Qiao et al. [13] introduced a cooperative vehicular edge caching system to optimize the content delivery and placement within a vehicular edge computing environment, with the aid of flexible cooperation between cellular stations, RSUs, and vehicular nodes. The optimization problem is formulated as a double-time-scale Markov decision process (DTS-MDP). Zhou et al. [14] introduced a new content delivery architecture by utilizing the 5G edge networks, where the content caching and data pre-fetching methods are discussed. Furthermore, they studied the comprehensive dynamic link utilization problem in 5G edge networks from the perspectives of network operators and vehicle users. Luo et al. [15] introduced an intelligent algorithm (EdgeVCD) that is based on a content distribution mechanism. It uses a dual-importance (DI) evaluation method to reflect the relationship between the Priority of Vehicles (PoV) and the Priority of Contents (PoC) and formulate an optimization problem to maximize the system utility for content distribution.

De Souza et al. proposed a Safe and Sound (SNS) approach, which uses a hybrid architecture and cooperative re-routing method to enhance the system performance and scalability [6]. SNS utilizes a recurrent neural network (RRN) to predict future safety risk dynamics and to offer a customized re-routing in which every vehicle chooses the risks to avoid. When the traffic server detects a congested road, it notifies the incoming vehicles by sending the traffic report to all vehicles having their paths crossing this road segment. The SNS re-routing strategy objective is to balance the traffic flow over a set of alternative routes for each vehicle based on their current and final positions and preferences. Soua et al. proposed a vehicular social networking architecture that combines the content-centric networking (CCN) model, Floating Content (CF), and Software Defined Networking (SDN) to offer a multi-pronged approach for adaptive content delivery [8]. Each content includes the location and name of the requester. Intermediate nodes that receive the requester message check it in their local content store (CS). If the requested content is not available in CS, they forward the message and trigger a timer. The FC is used to support geographic content routing. An SDN controller operates a direct path between the content requester and the provider, similar to the dynamic unicast method. Alowish et al. proposed content delivery architecture for vehicular networks called Cuckoo, in which the RSU delivers the user content via controller nodes [7]. In other words, RSU selects the optimal routing path to the provider location with the help of controller nodes. Moreover, Zhao et al. proposed DP-IB, a vehicular content delivery system that uses a data pouring and buffering mechanism for content dissemination in VANET. The source node sent data contents, buffered along the selected path, and rebroadcasted in the road intersections [9]. The main idea behind Data Pouring (DP) is instead of broadcasting the content to the entire network. The system only sends the disseminated content to a few road segments known as *axis roads (A-roads)*. Usually, A-roads segments have denser traffic flow than other roads and are chosen as main roads. Intersection Buffering (IB) means that the scheme disseminates the content to nodes travelling along the crossing roads (C-roads) intersecting the A-roads.

To deal with the privacy leakage problem in social vehicular networks, Zhang et al. introduced a distributed location privacy-preserving spatial crowdsourcing method for IoV [12]. It enables vehicular nodes to be involved in spatial crowdsourcing and guarantees the privacy of location information. They employed blockchain to record the user data without requiring a centralized spatial crowdsourcing server. In [13],

TABLE I
COMPARISON BETWEEN VESONET AND SIMILAR SCHEMES

System	Vehicle rerouting	Content caching	Content recommendation	Content delivery	Incentive method
SNS[6]	Risk-avoidance rerouting	No	No	No	Safe routes
Cuckoo [7]	SDN-based route selection	No	Policy-based bifold classifier	Best forwarder selection	No
CCN-CF [8]	No	SDN controlled on-demand caching	No	SDN-based path between a requester and a content source	No
DP-IB [9]	No	Intersection Buffering	No	Data pouring in main roads	No
Zhang et al [10]	No	Vehicular edge caching	No	Content dispatch using vehicular edge mechanism.	No
VeSoNet	Social-aware rerouting	popular content replication	Traffic-aware content recommendation	Content provider vehicles path planning	Monetized content

the authors investigated the use of blockchain and smart contracts to improve data storage security during content sharing among vehicles in vehicular edge networks. They concluded that blockchain technology could achieve content sharing. Furthermore, they introduced a reputation-based content-sharing scheme to guarantee the quality of the shared data among vehicular nodes.

VeSoNet has many novel characteristics that existing systems do not have. Table I compares VeSoNet with similar vehicular network systems on five main criteria: content rerouting, caching, recommendation, delivery, and security. VeSoNet has an efficient rerouting strategy and is the only system that considers the social interest of the users. It brings relevant content to users without congesting the network. Others focus mainly on traffic congestion and travel time without considering social interests. VeSoNet distributes the content across the road network to optimise delivery. Unlike other systems that pour the content into main roads, such as DP-IB and CCN-CF, without considering the distribution of content providers' locations. VeSoNet replicates only popular content to reduce the download time, while other schemes do not handle it. We can make the same observations for the criteria content-recommendation and security. VeSoNet is built around traffic-aware recommendations and monetised content, which others do not consider.

III. VEHICULAR EDGE ARCHITECTURE

The two common ways of delivering content in vehicular networks are V2I connection or cellular base stations using a 4G/5G interface [16]. The former is cheap and has a simple communication model, but it is not easy to directly access the content. The vehicles rely on V2V communications to reach the sparse RSUs [17]. The latter has the advantage of better coverage and instance access, but at the expense of expensive communications. To deal with these issues, VeSoNet implements a hybrid data distribution approach, where only popular data content is replicated and stored in vehicles to avoid excessive simultaneous downloads from 4G or 5G networks. In this regard, we distinguish three types of vehicular nodes: 1) Consumers form most vehicles in the network and represent the system end-users. 2) Provides store social content. The objective is to maximize the delivery to consumers as they travel through the city. 3) Meta-data vehicles, situated in busy locations of a city, provide information about content locations and perform various tasks, such as shortest social path calculation, content similarity, etc.

Fig. 1 illustrates an example of such a system. Consumer, provider, and meta-data server vehicles are represented in yellow, blue, and green, respectively. Meta-data servers are in busy locations, such as parking lots, where they are always present and not moving frequently, which ensures the quality of content lookup service. Meta-data servers maintain a table that contains a list of available contents in the network and a list of providers. The providers send frequent location and expected path updates to meta-data servers. The VeSoNet system follows Information-Centric Networking (ICN) model. When a consumer requests a given content, it creates a packet regarding the desired information that contains the content identifier and traffic information of the requesting vehicle, such as the expected travel path. The message is sent to all neighbouring nodes and forwarded to other nodes until it reaches the provider. When an intermediate node receives that packet and does not store the requested content, it forwards it to the nearest meta-data servers. If the providers do not have the requested content, RSU downloads it from an external network and forwards it to the requester. The providers back it up for future use.

IV. SYSTEM MODELING

A. Consumers Path Planning

The proposed framework leverages traffic information and dynamic changes in vehicles' travelling paths to bring the consumers close to providers, enhancing the content delivery experience. As a provider takes the same path as consumers, the delivery delay is significantly reduced. For instance, in Fig. 2, a consumer is travelling from the source location (S) to the destination location (D). Although path P_1 is the shortest path, the system recommends P_2 since it contains more providers and does not exceed the rerouting threshold as P_3 does.

Let $P_{sh} = \{I_s \rightarrow I_{x_1}, \ldots, I_{x_n} \rightarrow I_d\}$ be the shortest path based only on traffic information, without considering the availability of the providers. I_s and I_d are the starting intersection (source intersection) and destination intersection,

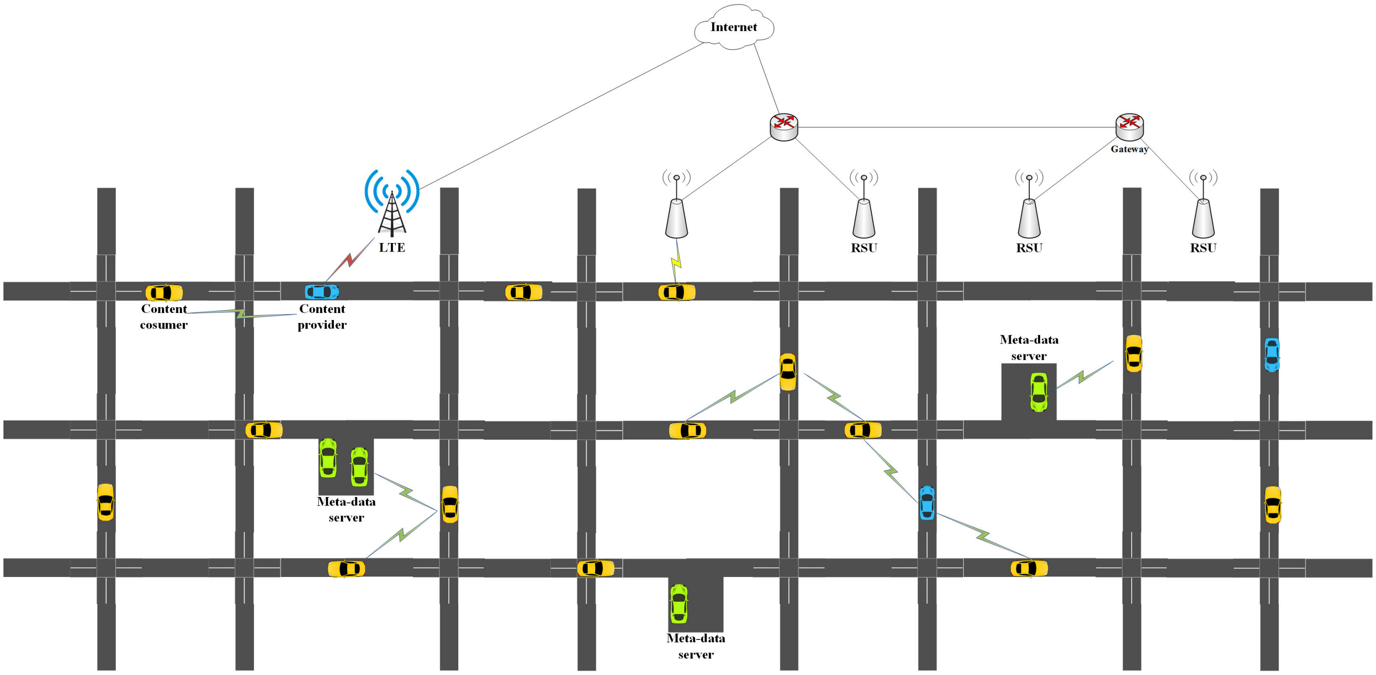


Fig. 1. Content dissemination architecture.

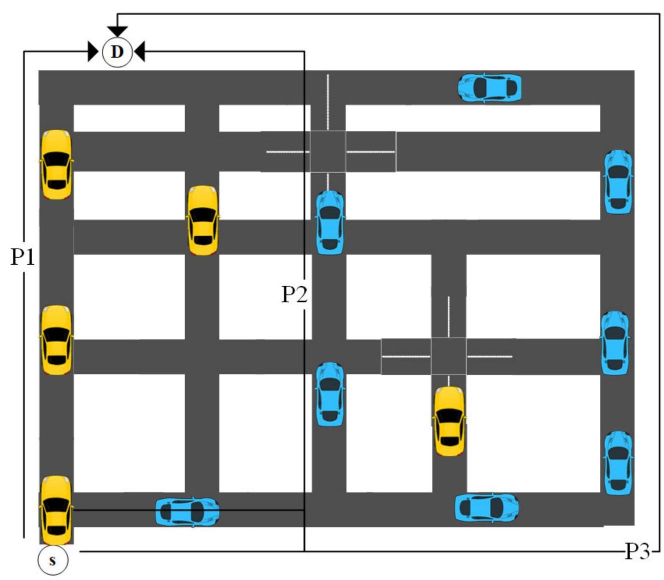


Fig. 2. Social path selection.

Algorithm 1 Shortest_Social_Path

Input $P_{sh} = \{I_s \rightarrow I_{x1}, \ldots, I_{xn} \rightarrow I_d\}$

Output $P_{so} = \{I_s \rightarrow I_{y1}, \ldots, I_{yn} \rightarrow I_d\}$

$P_{so} \leftarrow \{I_s \rightarrow I_{x1}\}$

$n_{cur} \leftarrow I_s$

while $n_{cur} \neq I_d$ **do**

  $n_{next} \leftarrow P_{sh}[next(n_{cur})]$

  $P_{temp} \leftarrow AlternitiveSocialPath(n_{cur}, n_{next})$

  $P_{par} \leftarrow AlternitiveSocialPath(I_s, n_{next})$

  $P_{can} \leftarrow P_{so}[I_s \rightsquigarrow n_{cur}] \cup P_{temp}$

  if $\tau(P_{par}) - \tau(P_{can}) \leq \varepsilon$ **then**

    if $|Providers(P_{can})| < |Providers(P_{par})|$ **then**

      $P_{so} \leftarrow P_{par}$

    else

      $P_{so} \leftarrow P_{so} \cup P_{temp}$

    end if

  end if

  $n_{cur} \leftarrow P_{so}[Last]$

end while

respectively. The objective is to find an alternative social-aware shortest path P_{so} that maximizes the number of providers for a vehicle during its trip, subject to the difference between shortest travel time $\tau(P_{sh})$ and social path $\tau(P_{so})$ is less than a threshold ε. A naive approach is to find the shortest paths between I_s and I_d, then consider the social path that maximizes the number of providers and satisfies the threshold ε. However, this approach is computationally expensive. So we proposed a graph pruning search algorithm that computes the social path as presented in Algorithm 1.

The proposed method is computationally efficient, as it uses a path pruning technique to eliminate the paths that exceed the path optimization threshold ε, hence reducing the search space. The Algorithm takes as input the traffic-only paths and goes through each road segment by keeping track of the current intersection n_{cur} (the current node) and the next intersection n_{next} (next node). P_{temp} (temporary social path) is defined as the optimal social path between n_{cur} and n_{next} that satisfies the threshold. P_{par} (partial social path) is defined as the optimal social path between the starting intersection I_s (source node) and the next intersection n_{next}. P_{can} (candidate social path) is defined as the optimal social path from the starting intersection I_s to the next intersection n_{next} and the temporary social path

P_{temp}. Note that this path is dynamically changing, as the alternative social-aware shortest path P_{so} is constantly updated over time. If the newly found partial social path P_{par} satisfies the path optimization threshold ($\tau\left(P_{par}\right) - \tau\left(P_{can}\right) \leq \varepsilon$), and the number of providers in P_{par} is higher than P_{can}, then P_{so} is set as P_{par}. Otherwise, P_{so} is updated to include the newly found P_{temp}. During a travel journey, VeSoNet keeps track of the current optimal path P_{opt} and its travel time t_{opt}, providers so_{opt}, and a set of the previously visited nodes $VIST$. This dynamic tracking and efficient way of finding and accessing social content make VeSoNet more reliable, scalable, and efficient. To do so, VeSoNet implements a set of incremental, efficient, and complementary algorithms. For instance, the Algorithm 2 finds an alternative social path between two intersections. The social graph pruning Algorithm 3 reduces the search space. The algorithm starts from the last node of the input path P and considers it as the current node n_{cur}. It checks whether the current node has not been visited before, or if it has been visited but there is a shorter travel time with more content providers (Lines 2-15). In this case, It evaluates all neighbours of the current node $\eta(n_{cur})$ (Lines 4-13). Firstly it computes the new travel time t_{new} and checks whether t_{new} is still less than the path optimization threshold ε. In that case, it computes the new count of content providers so_{new} and recursively runs a graph pruning procedure on the new path ($P \cup \{n_{cur} \rightsquigarrow n_{next}\}$). Otherwise, if the current road segment exceeds the path optimization threshold ε, the node is considered as visited (Lines 10-12) without going through its branches, hence the search space is considerably minimized. This reduces the computational cost of finding an alternative social-aware path. Finally, if the current node is the destination intersection ($n_{cur} = d$), the algorithm checks the travel time. If it is shorter than that of shortest the path (P_{sh}), then P is considered as an optimal social path.

Algorithm 2 Alternative_Social_Path

Input s, d
Output P_{opt}

$P_{opt} \leftarrow P_{sh}[s \rightsquigarrow d]$
$t_{opt} \leftarrow \tau(P_{sh}[s \rightsquigarrow d])$
$so_{opt} \leftarrow |Providers\,(P_{sh}[s \rightsquigarrow d])|$
$VIST \leftarrow \{(s, 0)\}$
SOCIAL_GRAPH_PRUNING($\{s\}, d, 0, 0$)
RETURN P_{opt}

Algorithm 3 Social_Graph_Pruning

Input P, d, t, so
Output P_{opt}

1: $n_{cur} \leftarrow P[Last]$
2: **if** ($n_{cur} \notin VIST$) OR (($n_{cur} \in VIST$) $AND\,(VIST[n_{cur}.t] \leq t)$ AND ($VIST[n_{cur}.so] \geq so$)) **then**
3: **if** $n_{cur} \neq d$ **then**
4: **for all** $n_{next} \in \eta(n_{cur})$ **do**
5: $t_{new} \leftarrow t + \tau(n_{cur} \rightsquigarrow n_{next})$
6: **if** $t_{new} \leq \tau(P_{sh}[s \rightsquigarrow d])$ **then**
7: $so_{new} \leftarrow so + Providers(n_{cur} \rightsquigarrow n_{next})$
8: $VIST \leftarrow VIST \cup \{n_{cur}, t_{new}, so_{new}\}$
9: SOCIAL_GRAPH_PRUNING ($P \cup \{n_{cur} \rightsquigarrow n_{next}\}, d, t_{new}, so_{new}$)
10: **else**
11: $VIST \leftarrow VIST \cup \{n_{cur}, t_{new}, so_{new}\}$
12: break
13: **end if**
14: **end for**
15: **else**
16: **if** $t \leq \tau(P_{sh}[s \rightsquigarrow d])$) and ($so \leq so_{opt}$) **then**
17: $so_{opt} \leftarrow so$
18: $P_{opt} \leftarrow P$
19: RETURN P_{opt}
20: **end if**
21: **end if**
22: **end if**

B. Traffic-Aware Content Recommendation

Given a provider vehicular node v_x that stores a set of social content items $c_x = \left\{c_{x_1}, c_{x_2}, \ldots, c_{x_j}\right\}$ and takes the path $p_x = \left\{i_{x_1}, i_{x_2}, \ldots, i_{x_k}\right\}$. Consider consumer vehicles $\Lambda(v_x)$ when v_x crosses p_x. At each traffic light of the road intersections in p_x, v_x requests the recommended content items from nearby providers (neighbourhood). Let v_y be a nearby provider that stores a set of social content items $c_y = \left\{c_{y_1}, c_{y_2}, \ldots, c_{y_j}\right\}$. The question is which items in c_y are more likely to be viewed/liked by the consumer vehicles $\Lambda(v_x)$.

We formulate this question as a link prediction problem to predict the potential links between the previously consumed content and the available content in nearby provider vehicles. Since the waiting time in intersections is relatively short, the classical filtering recommendation methods, such as matrix factorisation, are not suitable because these methods have high computational complexity and require knowledge of all vehicles' consumption history in the system.

Therefore, we propose a graph embedding-based content recommendation approach called Vehicle2vec, where the vehicles are represented by a set of low dimensional vectors of the previously consumed content. Vehicle2vec starts by learning the feature representations of each content available in the system. The content network is represented as a graph data structure, where the nodes represent the data content, and the edges represent the content similarity between these nodes. Vehicle2vec learns the content node low dimensional vector that preserves the neighbourhood of nodes in the original graph [18]. To build such content node embeddings, vehicle2vec uses stochastic gradient descent (SGD) to optimize the objective function, hence learning the low dimensional representation.

Let the content network be represented as a connected graph $G_c = (V_c, E_c)$, where V_c is a set of nodes and $E_c \subseteq (V_c \times V_c)$ a set of edges between nodes. The objective of network embedding learning is to represent each node $c \in V_c$ as a vector of low dimensional space R^d. It involves evaluating the mapping function $f_c : V_c \rightarrow R^d$, where $d \ll |V_c|$ is the number of features in low-dimensional space, and the original network structure is preserved. For each node $c \in V_c$, we compute the content network neighbourhood $N(c) \subset V_c$ that represents semantically similar data content. We use the Skip-gram model of networks. Hence, the mapping function

f_c is evaluated by optimizing the objective function given in the equation (1).

$$\max_{f} \sum_{c \in V_c} \log Pr\left[N(c) \mid f_c(c)\right] \quad (1)$$

According to the symmetry property in the feature space, the proximity between every pair of nodes is symmetric. Therefore, the conditional likelihood between every content node and its neighbours can be modelled using a softmax function, as given in the equation (2).

$$Pr\left(n_i \mid f_c(c)\right) = \frac{e^{(f_c(n_i) f_c(c)}}{\sum_{m \in V_c} e^{(f_c(m) f_c(c))}} \quad (2)$$

A consumer vehicle is associated with its Vehicle2vec matrix representing all embedding features of the previously consumed content. The recommendation method is given in Algorithm 4. v_x is the provider that takes the path p_x. For each intersection in p_x, v_x checks the content available in every neighbour v_y, and computes its similarity with its expected consumers $\Lambda(v_x)$. if the similarity is above the content similarity threshold α, then that content item is downloaded during the intersection waiting time.

Algorithm 4 INTERSECTION_RECOMMENDATION

Input v_x, P_x
Output R_x
1: **for all** v_y *in* i_x **do**
2: **for all** v_y *in* i_x **do**
3: **for all** c_j *in* v_y **do**
4: **for all** m $\in vehicle2vec(\Lambda(v_x))$ **do**
5: **if** $\left(\frac{\sum_{k \in \mathrm{m}} sim(vec(k), vec(c_j))}{|\mathrm{m}|} > \alpha\right)$ **then**
6: $R_x \leftarrow R_x \cup \{c_j\}$
7: **end if**
8: **end for**
9: **end for**
10: **end for**
11: **end for**

C. Content Provider Distribution

Finding optimal routing from content providers to consumers is known as the vehicle routing problem (VRP). VRP is NP-hard [19]. Various meta-heuristic algorithms have been proposed to find sub-optimal solutions, such as the firefly algorithm [20], genetic algorithm [21] or hybrid meta-heuristic algorithms [22]. In addition, these meta-heuristic algorithms assume stable traffic conditions. Unlike traditional VRP, in the problem at hand, the traffic flow is constantly changing over time. In other words, one needs to find near-optimal solutions to a VRP with dynamic traffic conditions.

Given a content provider vehicle v_x travelling from starting position PoS_s to PoS_D taking the path p_x, the objective is to optimize the revenue generated from the advertisements delivered to consumers. An intuitive approach would be to choose a road that maximizes the number of consumer vehicles, but when the same road contains many provider vehicles, it cannot optimize its revenues. We define content deliverability, $CD(r_x, t_k)$, in road segment r_x at time slot t_k, as the difference between the number of consumers (CC) and number of providers (CP), (see equation (3)).

$$CD(r_x, t_k) = CC(r_x, t_k) - CP(r_x, t_k) \quad (3)$$

To deal with the dynamic traffic conditions and time constraints, we propose a distributed method that exploits Deep Reinforcement Learning (DRL) to find sub-optimal solutions to the problem of delivering the content to consumers. The reinforcement learning model evaluates the actions that change the system state at each step through reward and punishment.

In the proposed social-aware DRL model, the system states represent the current road segments and content deliverability. An action represents a decision taken by a provider at a road intersection during its travel path. The provider's objective is to optimize content deliverability. The reward is the difference between the old and new content deliverability (after executing an action) (Equation (4)).

$$r = O_{cd} - N_{cd} \quad (4)$$

The proposed model is trained using the Deep Q-Network (DQN) approach. We use a Q-learning system to choose an optimal option for all (action, state) pairs by evaluating the Q value of that action, $Q(s_n, a_s)$ (see equation (5)), where s_o represents the previous state, a_o the previous action, r the corresponding reward of that action, and γ the discount factor.

$$Q(s_n, a_s) = Q(s_o, a_o) + \beta(r + \gamma \ Q(s_n, a_s)) - Q(s_0, a_0) \quad (5)$$

The input is fed into a multilayer neural network that is used as a function that maps the system state to the corresponding Q-value. After the training, the network is used to predict the optimal action to take in a given situation in order to maximize the content deliverability reward r. In every intersection, a provider is aiming to take the optimal action in current state s as shown in (6), where $Q(s, a)$ is the list Q-values yielded from the neural network output; Therefore, VeSoNet chooses the action that yielded the maximum rewards, as shown in (7) where s, a are the current action and state.

$$Q(s_n, a_s) \leftarrow r(s_n, a_s) + \gamma \max_{\grave{a}} Q(s, \grave{a}) \quad (6)$$

$$a = argmax_{\grave{a}} \ Q(s, \grave{a}) \quad (7)$$

The content providers execute an epsilon-greedy policy, where they either take random actions or actions suggested by Q-network. It uses the loss function to compute the squared difference between the predicted and target values, hence the loss is minimized by updating the network weights [23], as shown in (8), where $\{\theta_i\}$ and θ_i^- are the weights of the Q-network at iteration (i).

$$L_i(\theta_i) = \left(r + \gamma \max_{\grave{a}} Q(\grave{s}, \grave{a}; \theta_i^-) - Q(s, a; \theta_i)\right)^2 \quad (8)$$

V. Experimental Evaluation

System evaluation is challenging, mainly when it involves ML/AI components. Many techniques have been developed

TABLE II
SIMULATION PARAMETERS

Parameter	Value
PHY model	IEEE 801.11p
MAC model	EDCA
Propagation model	Two rays
Fading model	Rayleigh fading
Antenna model	Omnidirectional
Shadowing model	LogNormal
Channel frequency	5.890e9 Hz
Propagation distance	450m
Transmission power	20 mW
Maximum hop count	15
Scenario map	London
Scenario area	$5km^2$

TABLE III
DRL PARAMETERS

Parameter	Value
Episodes	10,000
Greedy exploration rate	$1 \rightarrow 0.05$
Discount factor	0.99
Network update / learning step	3000
Number of actions	4
States	Dynamic
Learning rate	0.0001
Replay memory size	10,000
Batch size	32

depending on the type of evaluation and system at hand. User testing and simulations are among the most common evaluation techniques. User testing is valuable, but one cannot rely solely on it to get a good evaluation of large systems with complex behaviours. Simulation approaches are well-suited to evaluate large and well-defined systems that use detailed modelling. Implementing a dedicated simulator for a given system is not straightforward and time-consuming. Fortunately, many freely available and open-source simulators can be used to evaluate the various parts of the system and study its performance. We used two different simulators to evaluate three different parts of the system: social content network, vehicular network, and urban mobility traffic components. More precisely, we use the urban mobility (SUMO) simulator to generate the traffic [24], the OMNeT++ network simulator [25], the vehicular network framework Veins on top of OMNeT++ [26]. We give more details about each of them in the following sections.

A. Experiments and Dataset

As a case study, we use a real-world map of Greater London to simulate the traffic. OpenStreetMap allows us to extract the map raw data [27]. We process further the map (using the Netconvert tool) to generate the road network. SUMO takes as input the road network to generate the traffic data. Table II shows a detailed description of the network simulation parameters. We use the dataset "Last. FM" [28] to simulate the social network content. The dataset contains the social networking and music-listening information of more than 2000 users. Vehicles are randomly placed on the map, and they travel to randomly chosen destinations. Each vehicle is randomly linked with two users from the "Last. FM" dataset, and we consider them as passengers within that vehicle. Once a vehicle is added to the simulation, we measure the shortest path to its destination point and estimate the required travel time, and then it is routed according to the shortest social path as per the proposed algorithm.

VeSoNet uses a neural network composed of four layers, the input and output layers, in addition to two fully connected hidden layers, the hidden layers have 150 neurons and 100 neurons respectively, and both use RELU as an activation function. We used the experience replay strategy to update weights during the updating process. As presented in Table III, the number of performed episodes in SUMO is 10,000 with 0.0001 set as the learning rate in order to train a vehicle agent. The discount factor for the reward is 0.99, and the greedy exploration rate is decreased from 1 to 0.05 to balance the trade-off between exploration and exploitation [29]. VeSoNet creates a stacked memory that consists of the old state, old action, reward, and current state, and to store transactions, the reply memory size is set as 10,000 and the batches for training as 32 [19]. The priority exponent is set as 0.6, whereas the sampling for prioritization importance is increased from 0.4 to reach 1. We apply Xavier initialization [30] to initialize all the trainable parameters. For the training and testing, we used 10-fold cross-validation, where the training testing split ratio is 80:20. The neural network learns the parameters by training, and the final network is ready after the training process. When VeSoNet's DRL module is well-trained by experience, a vehicular agent processes the current state of the intersection and computes the content deliverability of each road segment and the number of available data providers and data consumers, and checks if the appropriate action is present in replay memory, and predict the best action in the current situation.

B. Baselines and Metrics

As mentioned in the literature review, some existing systems are close to VeSoNet. These include Cuckoo [7], Safe and Sound (SNS) [6], CCN-CF [8], and DP-IB [9]. As VeSoNet has different architecture and operates in a scalable and dynamic environment, we conducted a comparative study to evaluate the effectiveness of its innovative features compared to these systems. We use the following metrics to evaluate the system performance:

Delivery Delay (DD): is the time required to deliver the requested content. Formally, DD is the time elapsed from the time the content was requested T_{req} to the time when it was received T_{rec} (Equation (9).

$$DD = T_{rec} - T_{req} \tag{9}$$

Delivery Rate (DR): is the ratio of successfully delivered content C_{del} by the total requested content C_{req}, (Equation (10)). This measure is considered the system precision performance metric. It assumes that the system has a maximum threshold above which one can declare that the content is not

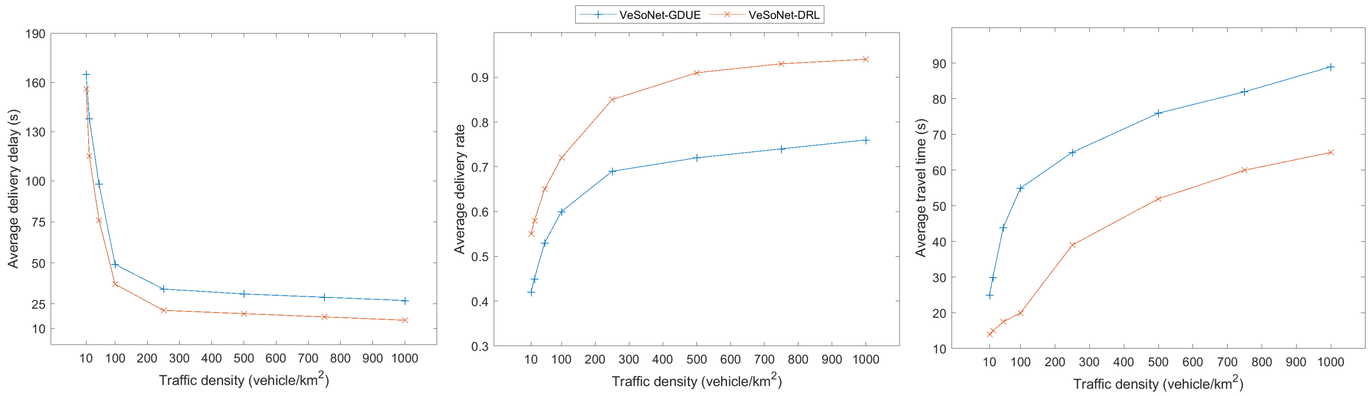


Fig. 3. Comparing VeSoNet-DRL and VeSoNet-GDUE in terms of (a) delivery delay (b) delivery rate (c) travel time.

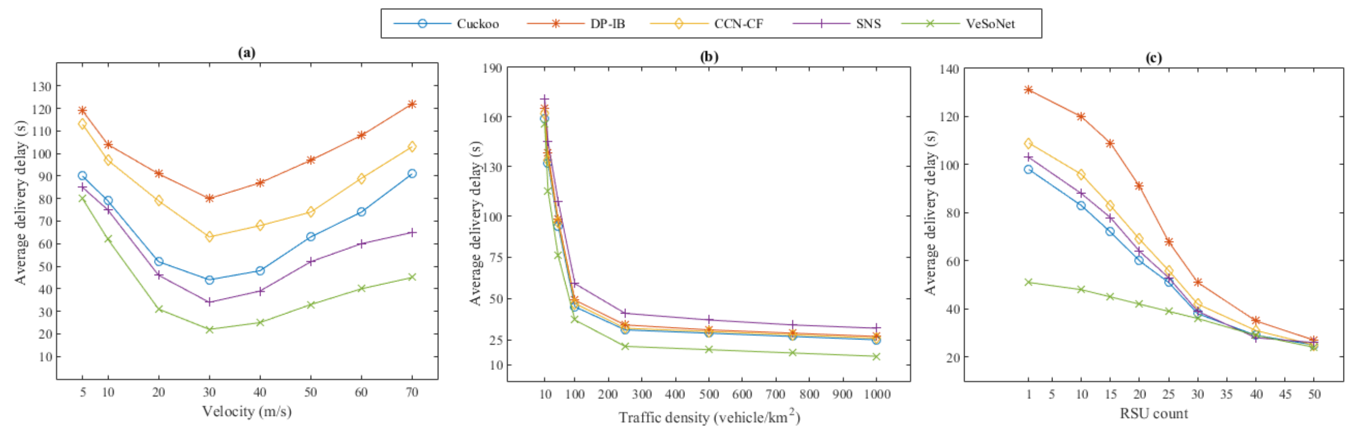


Fig. 4. Comparing average delivery delay under (a) increasing velocity (b) increasing traffic density (c) different RSUs count.

delivered and the request is either reissued or abandoned.

$$DR = \frac{|C_{del}|}{|C_{req}|} \tag{10}$$

Travel time (TT): is the time required to travel from the starting location to the destination location. This metric measures the additional travel time eventuated by taking an alternative social path rather than the shortest travel path. It is the difference between the arrival timestamp T_{arr}, and departure timestamp T_{dep} (Equation (11)).

$$TT = T_{arr} - T_{dep} \tag{11}$$

Computational Cost: is the number of operations required to perform the computational task of the system, such as path rerouting, content recommendation and traffic prediction.

C. Results Analysis

To evaluate the effectiveness of deep reinforcement learning for optimizing the content provider vehicles' path planning, we have compared VeSoNet when applying deep reinforcement learning for path planning with VeSoNet when applying Gawron's Dynamic User Equilibrium (GDUE) [31] for content provider vehicles path planning. GDUE is one of the default models in SUMO, it uses Dynamic Traffic Assignment (DTA) to model the traffics via a discrete time-dependent network, and it assigns content provider vehicles using the shortest path algorithm, regardless of the number of content consumer vehicles in the shortest path's roads. Figure 3 shows the average content delivery delay, average content delivery rate and average travel time of content provider vehicles when using VeSoNet-DRL and VeSoNet-GDUE. In Figure 3 (a), in low-density scenarios both systems have long delivery delays and lower delivery rates due to connectivity problems when few fast-moving vehicles are simulated, but with the increase of the number of vehicles the average delivery delay significantly decreases and delivery rate increases as the content can be delivered easily in dense environments. VeSoNet-DRL has a lower delivery delay and higher delivery rate across all settings because the trained agent chooses the most appropriate path that maximizes its content deliverability by choosing routes according to the availability of content the consumer and content providers vehicles in each road. On the other hand, in VeSoNet-GDUE content provider vehicles choose the shortest path, even if that path will diverge them away from their content consumers, which increases the delivery delay. Similarly, the average travel time is significantly higher in VeSoNet-GDUE compared to VeSoNet-DRL, that is because the roads get congested when all the content provider vehicles choose the shortest path to their destinations.

Fig. 4 shows the average delivery delay in different settings. In Fig. 4 (a), we increase the velocity of the vehicles and observe the delivery delay. We can see that all systems have relatively high delivery delays in low-velocity environments (less than 20m/s). Because in low-velocity environments, content deliveries are slowed down in store-and-carry situations when there are no nearby vehicles. Moderate velocity settings

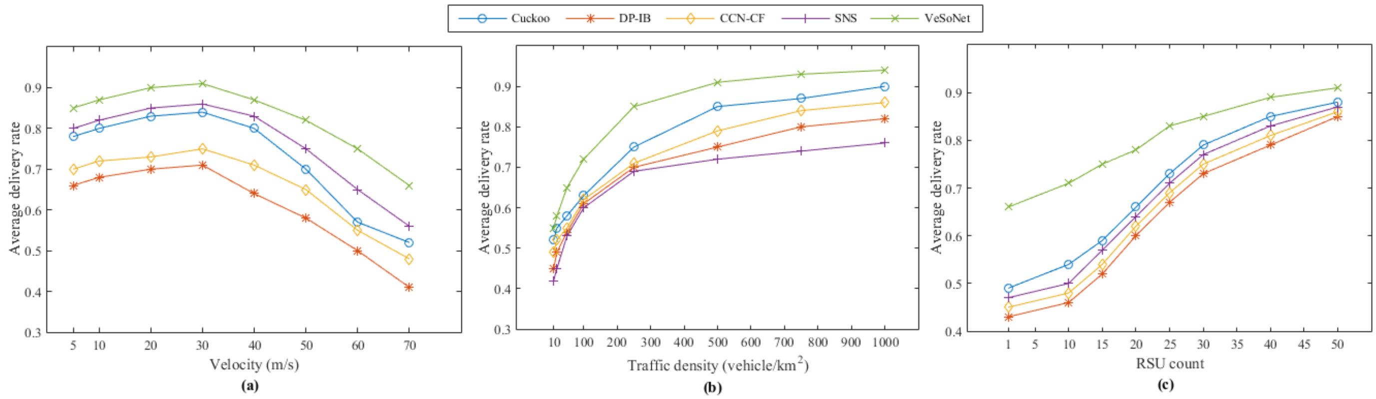


Fig. 5. Comparing average delivery rate under (a) increasing velocity (b) increasing traffic density (c) different RSUs count.

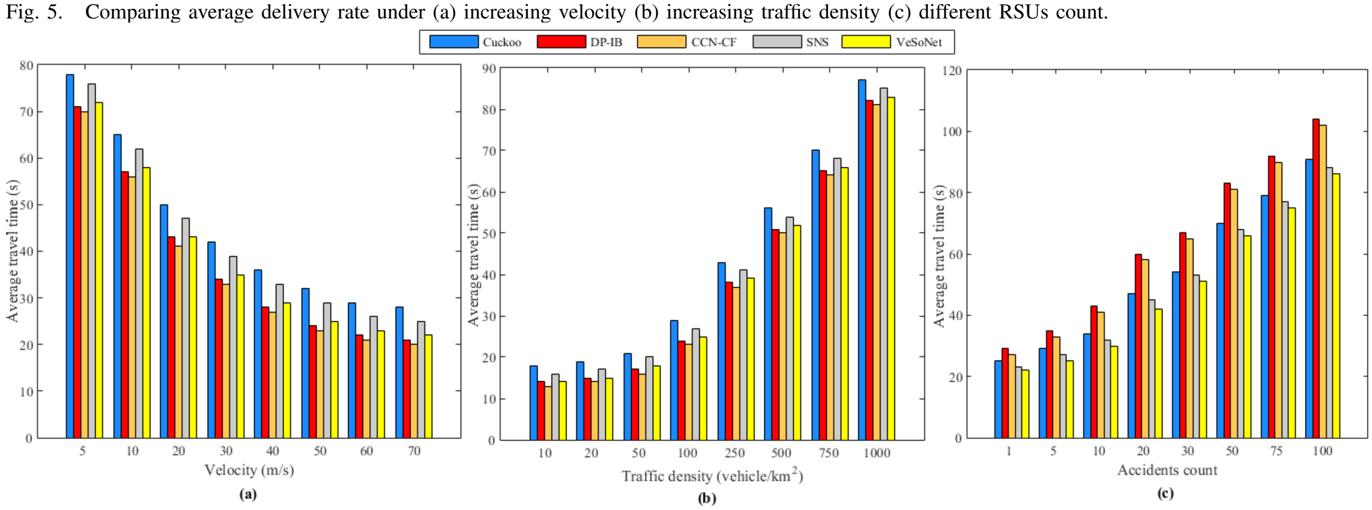


Fig. 6. Comparing average vehicle travel time under (a) increasing velocity (b) increasing traffic density (c) increasing traffic accidents count.

(25m/s to 45 m/s) yield the shortest delivery delay for all systems. While in high-velocity cases (more than 45 m/s), the delivery delay increases again. This is because of frequent disconnections due to high velocity, which makes it hard to establish V2V communications, hence increasing the delivery time of content requests and replies. Also, we can observe that VeSoNet, SNS and Cuckoo) outperform CCN-CF, as it does not leverage path planning or vehicle rerouting, and DP-IB, which relies on the A-roads and C-roads intersections to deliver the content. In Fig. 4 (b), we increase the traffic density and observe its impact on the delivery delay. We can see that all systems have longer delivery delays in low-density environments (10-50 vehicles). Because, in sparse environments, it becomes hard to route messages between vehicles. They need to travel long distances to find other content providers. However, the delivery delays are shortened considerably in high-density environments (more than 100 vehicles). VeSoNet still has the upper hand compared to other baselines due to the social-aware rerouting strategy and popular content replication method. As the density increases, the demand for such popular content increases accordingly, which contributes to the decrease in average delivery delays. Fig. 4 (c) shows the average delivery delays while varying RSUs. The objective is to study the system behaviour in case of network failures and its ability to work in infrastructure-less environments. With a high RSUs count, all the baselines have similar delivery delays. However, as the RSUs count decreases, we can see that VeSoNet outperforms other systems. Where most popular contents are stored in providers, distributed across the network, and indexed in meta-data servers, consumers request and get the content through V2V without V2I communications. DB-IP has the worst performance, as it mainly depends on RSU to broadcast the content in A-roads, and as the number of RSU decreases, so does the number of A-Roads, hence the number of intersections between A-roads and C-roads decreases as well.

In Fig. 5 (a), we vary the velocity of the vehicles and observe the delivery rates. We can see that all the systems have high delivery rates in low to medium-velocity environments and vice-versa. Because V2V communications frequently disconnect, hence reducing the delivery rate. VeSoNet, SNS, and Cukoo still perform better than the others for the same reasons mentioned above. In Fig. 5 (b), the density of the vehicles is varied to measure its impact on the delivery rate. We can see that all systems have a relatively low delivery rate in low-density environments (less than 100 vehicles). Because in sparse environments, it becomes challenging to forward messages to vehicles. These results are consistent with the first experiment. Fig. 5 (c) shows the average delivery rate with various RSUs counts. Similarly, we can see that the average delivery rate of VeSoNet is higher than other baselines in a few RSUs settings due to VeSoNet's distributed

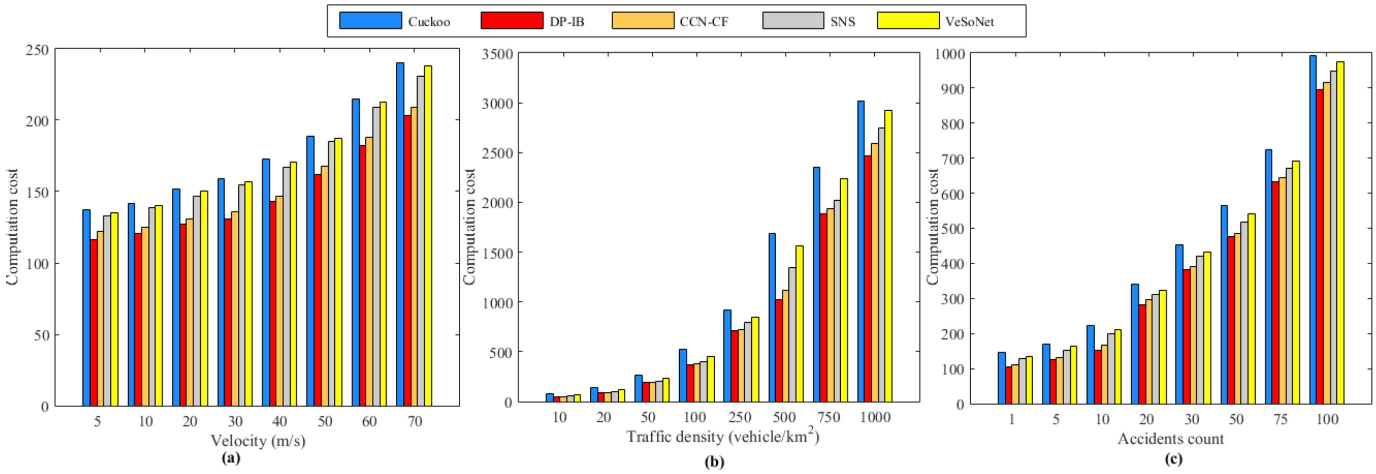


Fig. 7. Comparing average system computational cost under (a) increasing velocity (b) increasing traffic density (c) increasing traffic accidents count.

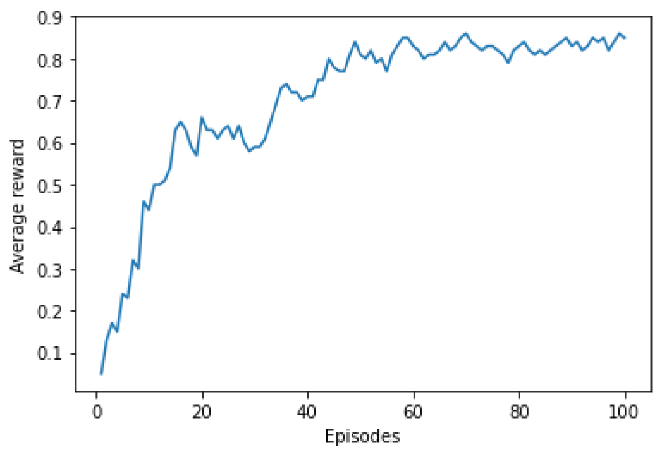


Fig. 8. The convergence of DRL in terms of average reward.

caching scheme. That is, most popular contents are stored in the providers, distributed over the network and indexed in meta-data servers. The consumers request and get the content through V2V without using V2I communications. DB-IP has the worst performance among the baselines, as it depends mainly on RSU to broadcast the content in A-Roads and C-Roads. Moreover, deploying many RSUs is very expensive [32].

However, the effectiveness of VeSoNet in delivery delays and rates comes with the cost of longer travel time and computational overhead, as shown in Fig. 6 and Fig. 7. In Fig. 6 (a), we increase the velocity of the vehicles and measure the travel time. We can see that all systems have longer travel times in low-velocity settings (less than 30 m/s), but it exponentially decreases as we increase the speed from 5 to 20. It stabilizes after 30 due to the stopping times at the intersections. Like in the previous experiments, VeSoNet, SNS, and Cuckoo present the same behaviours. They have longer travel times because the rerouted paths are usually longer than the shortest travel paths. For instance, VeSoNet's consumers choose routes having more relevant content providers as long as the chosen paths do not exceed the social threshold. In Fig. 6 (b), we vary the density of the vehicles to measure the impact on the travel time. Similarly, we observe that the same three systems still have longer travel times in all traffic density scenarios. However, the vehicle rerouting baselines have shorter travel times when the number of accidents increases due to the rerouting strategy that allows VeSoNet to avoid the accident roads (see Fig. 6 (c)).

Fig. 7 shows the baselines' average computational costs against content request count, density, and accident count settings. In Fig. 7 (a), we increase the average content requests and calculate the computational cost. VeSoNet has a relatively high computational cost compared to other systems. In fact, VeSoNet requires the cooperation of multiple vehicular servers to execute the requests. Thus, the updates and coordination of these vehicular servers are expensive. Moreover, unlike other baselines, where the computations are performed on a centralized server, VeSoNet distributes the computations among several servers, and the computational cost in each server is considerably low. In Fig. 7 (b), we evaluate the impact of the density on the computational cost. The computational cost exponentially increases with the increase of traffic density. VeSoNet has the second-highest computational cost, but this can be much lower as the computations are distributed among various servers. Finally, Fig. 7 (c), we study the impact of the accident count on computational cost. As the traffic accidents count increases, the baselines' computational costs increase. Furthermore, other systems have relatively higher computational costs due to the computational overheads of re-computing alternative travel routes. Fig.8 shows the convergence of DRL in terms of average reward (reward/episode length) when varying the episodes count, the episodes are sampled every 100 episodes. As we can observe, the average award exponentially increases in the early episodes, then stabilizes afterwards.

VI. Conclusion

In this paper, we presented a traffic-aware vehicular content caching architecture that optimizes content dissemination among vehicles using a social-aware graph pruning technique. This technique computes and assigns the shortest paths with the most relevant providers to the corresponding consumers.

To recommend relevant content according to their social context, we proposed a traffic-aware content recommendation approach based on graph embeddings. We described an efficient formal model, where vehicles are represented by a set of low dimensional vectors (vehicle2vec) of their previously consumed content. Experimental results show that the proposed architecture reduces content delivery delay and delivery ratio by more than 20% compared to the state-of-the-art baselines, at a slightly higher computational cost and average travel time. However, there are aspects for future improvements:

- Although all the communications between content consumers and content providers are encrypted, however, it is still possible to perform statistical attacks to infer the content consumers future paths, the privacy of the content consumers can be preserved by adding a pseudonyms identification scheme.
- The vehicular edge computing architecture can be further extended by adding computational task offloading, where all the computational tasks are performed in the vehicles.
- The social path selection process could be further extended to include driver preferences for individual road selection.
- We have used CNN as a training model for DRL. VeSoNet can be further developed by optimizing the training model.

References

[1] J. Voelcker. *1.2 Billion Vehicles on World's Roads Now, 2 Billion By 2035: Report*. Accessed: Jul. 5, 2022. [Online]. Available: https://greencarreports.com/news/1093560_1-2-billion-vehicles-on-worlds-roads-now-2-billion-by-2035-report

[2] N. Aung, W. Zhang, S. Dhelim, and Y. Ai, "T-Coin: Dynamic traffic congestion pricing system for the Internet of Vehicles in smart cities," *Information*, vol. 11, no. 3, p. 149, Mar. 2020. [Online]. Available: https://www.mdpi.com/2078-2489/11/3/149

[3] W. Zhang, N. Aung, S. Dhelim, and Y. Ai, "DIFTOS: A distributed infrastructure-free traffic optimization system based on vehicular ad hoc networks for urban environments," *Sensors*, vol. 18, no. 8, p. 2567, Aug. 2018. [Online]. Available: http://www.mdpi.com/1424-8220/18/8/2567

[4] S. Dhelim, H. Ning, F. Farha, L. Chen, L. Atzori, and M. Daneshmand, "IoT-enabled social relationships meet artificial social intelligence," *IEEE Internet Things J.*, vol. 8, no. 24, pp. 17817–17828, Dec. 2021.

[5] A. Naouri, H. Wu, N. A. Nouri, S. Dhelim, and H. Ning, "A novel framework for mobile-edge computing by optimizing task offloading," *IEEE Internet Things J.*, vol. 8, no. 16, pp. 13065–13076, Aug. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9372288/

[6] A. M. de Souza, T. Braun, L. C. Botega, L. A. Villas, and A. A. F. Loureiro, "Safe and sound: Driver safety-aware vehicle re-routing based on spatiotemporal information," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3973–3989, Sep. 2020.

[7] M. Alowish, Y. Shiraishi, Y. Takano, M. Mohri, and M. Morii, "A novel software-defined networking controlled vehicular named-data networking for trustworthy emergency data dissemination and content retrieval assisted by evolved interest packet," *Int. J. Distrib. Sensor Netw.*, vol. 16, no. 3, 2020, Art. no. 1550147720909280.

[8] R. Soua et al., "SDN coordination for CCN and FC content dissemination in VANETs," in *Ad Hoc Networks*. Cham, Switzerland: Springer, 2017, pp. 221–233.

[9] J. Zhao, Y. Zhang, and G. Cao, "Data pouring and buffering on the road: A new data dissemination paradigm for vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 56, no. 6, pp. 3266–3277, Nov. 2007.

[10] K. Zhang, J. Cao, H. Liu, S. Maharjan, and Y. Zhang, "Deep reinforcement learning for social-aware edge computing and caching in urban informatics," *IEEE Trans. Ind. Informat.*, vol. 16, no. 8, pp. 5467–5477, Aug. 2020.

[11] R. A. Dziyauddin et al., "Computation offloading and content caching and delivery in vehicular edge network: A survey," *Comput. Netw.*, vol. 197, Oct. 2021, Art. no. 108228.

[12] K. Zhang, J. Cao, S. Maharjan, and Y. Zhang, "Digital twin empowered content caching in social-aware vehicular edge networks," *IEEE Trans. Computat. Social Syst.*, vol. 9, no. 1, pp. 239–251, Feb. 2022.

[13] G. Qiao, S. Leng, S. Maharjan, Y. Zhang, and N. Ansari, "Deep reinforcement learning for cooperative content caching in vehicular edge computing and networks," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 247–257, Jan. 2020.

[14] H. Zhou, N. Cheng, J. Wang, J. Chen, Q. Yu, and X. Shen, "Toward dynamic link utilization for efficient vehicular edge content distribution," *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 8301–8313, Sep. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8732444/

[15] Q. Luo, C. Li, T. H. Luan, and W. Shi, "EdgeVCD: Intelligent algorithm-inspired content distribution in vehicular edge computing network," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5562–5579, Jun. 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9036956/

[16] N. Aung, W. Zhang, S. Dhelim, and Y. Ai, "Accident prediction system based on hidden Markov model for vehicular ad-hoc network in urban environments," *Information*, vol. 9, no. 12, p. 311, Dec. 2018. [Online]. Available: http://www.mdpi.com/2078-2489/9/12/311

[17] N. Aung, W. Zhang, K. Sultan, S. Dhelim, and Y. Ai, "Dynamic traffic congestion pricing and electric vehicle charging management system for the Internet of Vehicles in smart cities," *Digit. Commun. Netw.*, vol. 7, no. 4, pp. 492–504, Nov. 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2352864821000080

[18] S. Dhelim, N. Aung, M. T. Kechadi, H. Ning, L. Chen, and A. Lakas, "Trust2Vec: Large-scale IoT trust management system based on signed network embeddings," *IEEE Internet Things J.*, vol. 10, no. 1, pp. 553–562, Jan. 2023.

[19] S. Koh et al., "Real-time deep reinforcement learning based vehicle navigation," *Appl. Soft Comput.*, vol. 96, Nov. 2020, Art. no. 106694.

[20] A. M. Altabeeb, A. M. Mohsen, and A. Ghallab, "An improved hybrid firefly algorithm for capacitated vehicle routing problem," *Appl. Soft Comput.*, vol. 84, Nov. 2019, Art. no. 105728.

[21] M. A. Mohammed, M. K. A. Ghani, R. I. Hamed, S. A. Mostafa, M. S. Ahmad, and D. A. Ibrahim, "Solving vehicle routing problem by using improved genetic algorithm for optimal solution," *J. Comput. Sci.*, vol. 21, pp. 255–262, Jul. 2017.

[22] B. Pan, Z. Zhang, and A. Lim, "A hybrid algorithm for time-dependent vehicle routing problem with time windows," *Comput. Oper. Res.*, vol. 128, Apr. 2021, Art. no. 105193.

[23] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[24] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "Sumo—Simulation of urban mobility: An overview," in *Proc. 3rd Int. Conf. Adv. Syst. Simulation*, 2011, pp. 1–6.

[25] A. Varga, "OMNeT++," in *Modeling and Tools for Network Simulation*. Cham, Switzerland: Springer, 2010, pp. 35–59.

[26] C. Sommer et al., "Veins: The open source vehicular network simulation framework," in *Recent Advances in Network Simulation*. Cham, Switzerland: Springer, 2019, pp. 215–252.

[27] (May 2021). *OpenStreetMaps*. [Online]. Available: https://www.openstreetmap.org/

[28] GroupLens. *Last.FM*. Accessed: Dec. 30, 2021. [Online]. Available: https://grouplens.org/datasets/hetrec-2011/

[29] A. Mushtaq, I. U. Haq, M. U. Imtiaz, A. Khan, and O. Shafiq, "Traffic flow management of autonomous vehicles using deep reinforcement learning and smart rerouting," *IEEE Access*, vol. 9, pp. 51005–51019, 2021.

[30] J. Sirignano and K. Spiliopoulos, "Scaling limit of neural networks with the xavier initialization and convergence to a global minimum," 2019, *arXiv:1907.04108*.

[31] G. Peque, T. Miyagi, and F. Kurauchi, "Adaptive learning algorithms for simulation-based dynamic traffic user equilibrium," *Int. J. Intell. Transp. Syst. Res.*, vol. 16, no. 3, pp. 215–226, Sep. 2018.

[32] D. Kim, Y. Velasco, W. Wang, R. N. Uma, R. Hussain, and S. Lee, "A new comprehensive RSU installation strategy for cost-efficient VANET deployment," *IEEE Trans. Veh. Technol.*, vol. 66, no. 5, pp. 4200–4211, May 2017.

Nyothiri Aung received the master's degree in information technology from Mandalay Technological University, Myanmar, in 2012, and the Ph.D. degree in computer science and technology from the University of Science and Technology Beijing, China, in 2020. She was a Tutor with the Department of Information Technology, Technological University of Meiktila, Myanmar, from 2008 to 2010, and a System Analyst with ACE Data System, Myanmar, from 2012 to 2015. She is a Post-Doctoral Researcher with University College Dublin, Ireland. Her research interests include social computing, intelligent transportation systems, and medical image processing.

Sahraoui Dhelim received the master's degree in networking and distributed systems from the University of Laghouat, Algeria, in 2014, and the Ph.D. degree in computer science and technology from the University of Science and Technology Beijing, China, in 2020. He was a Visiting Researcher with Ulster University, U.K., from 2020 to 2021. He is a Post-Doctoral Researcher with University College Dublin, Ireland. His research interests include social computing, user modeling, deep-learning, recommendation systems, and intelligent transportation systems. He is the Workshop Chair of Cyberspace Congress (CyberCon).

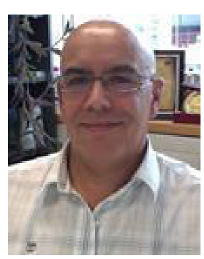
Liming Chen (Senior Member, IEEE) received the B.Eng. and M.Eng. degrees from the Beijing Institute of Technology (BIT), Beijing, China, and the Ph.D. degree in artificial intelligence from De Montfort University, U.K. He is a Professor with the School of Computer Science and Informatics, University of Ulster, Newtownabbey, U.K. His research interests include data analysis, ubiquitous computing, and human–computer interaction. He is a fellow of IET and a member of the IEEE Computational Intelligence Society (IEEE CIS) and the IEEE CIS Smart World Technical Committee (SWTC). He has served as an Expert Assessor, a Panel Member and an Evaluator for the Engineering and Physical Sciences Research Council (EPSRC)—a member of the Peer Review College, U.K., the Economic and Social Science Research Council (ESRC), the European Commission Horizon 2020 Research Program, the Danish Agency for Science and Higher Education, Denmark, the Canada Foundation for Innovation (CFI), Canada, the Chilean National Science and Technology Commission (CONICYT), Chile, and The Netherlands Organization for Scientific Research (NWO), The Netherlands. He is the Founding Chair of the IEEE CIS SWTC Task Force on User-Centered Smart Systems (TF-UCSS).

Abderrahmane Lakas (Senior Member, IEEE) received the M.S. and Ph.D. degrees in computer systems from the University of Pierre et Marie Curie (Paris VI, France). He has several years of experience in both academic and industrial. He spent two years as a Post-Doctoral Researcher with the School of Computing and Communication, University of Lancaster, U.K. He is a Professor with the Computer and Network Engineering Department, College of IT, United Arab Emirates University. He is the Head of Connected Intelligent Autonomous Systems (CAST) Research Group and the Connected Autonomous Intelligent Systems Laboratory (ASIL). He has published several research papers in scholarly journals. His current research interests include intelligent transportation systems, vehicular ad hoc networks, unmanned ground and aerial vehicles, autonomous systems, smart cities, the Internet of Things, and QoS. He is a member of TPC and a reviewer of several renowned conferences and serves in the editorial board for few journals.

Wenyin Zhang received the M.S. degree from the Shandong University of Science and Technology and the Ph.D. degree in computer science from the Chengdu Branch, Chinese Academy of Sciences (CAS). He is a Full Professor and the Dean of the School of Information Science and Engineering, Linyi University, China. He is the Vice Chairperson of the Shandong Computer Society and the Dean of the Linyi Smart Big Data Research Institute. He is Visiting Scholar with the University of Southern Mississippi and Portland State University. His main research interests include computer vision, digital watermarking, network information security, and blockchain technology application.

Huansheng Ning (Senior Member, IEEE) received the B.S. degree from Anhui University in 1996 and the Ph.D. degree from Beihang University in 2001. He is currently a Professor and the Vice Dean of the School of Computer and Communication Engineering, University of Science and Technology Beijing, China. He has published more than 150 journals/conference papers and authored five books. His current research interests include the Internet of Things and general cyberspace. He is the Founder and the Chair of the Cyberspace and Cybermatics International Science and Technology Cooperation Base. He has presided over many research projects, including the Natural Science Foundation of China and the National High Technology Research and Development Program of China (863 Project). He has been a Steering Committee Member of IEEE INTERNET OF THINGS JOURNAL, since 2016. He was an Associate Editor of IEEE INTERNET OF THINGS JOURNAL, from 2014 to 2018, and has been an Associate Editor of IEEE SYSTEMS JOURNAL, since 2013.

Souleyman Chaib received the Ph.D. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 2018. He is currently an Assistant Professor with the Ècole Supèrieure en Informatique (ESI) SBA, Sidi Bel Abbès, Algeria. His current research interests include remote sensing, deep learning, image scene classification, and geographic information systems.



Mohand Tahar Kechadi received the master's and Ph.D. degrees in computer science from the University of Lille 1, France. He is a Full Professor with the School of Computer Science, University College Dublin, Ireland. His research interests include data mining, distributed data mining heterogeneous distributed systems, grid and cloud computing, and digital forensics and cyber-crime investigations. He is a member of the *Communications of the ACM* journal and the IEEE Computer Society. He is an Editorial Board Member of the journal of *Future Generation Computer Systems*.