View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Lancaster E-Prints

2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS)

Proactive Content Caching for Internet-of-Vehicles based on Peer-to-Peer Federated Learning

Zhengxin Yu, Jia Hu, Geyong Min, Han Xu, Jed Mills
Department of Computer Science, University of Exeter, UK
Email:{zy246, j.hu, g.min, hx230, jm729}@exeter.ac.uk

2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS) | 978-1-7281-9074-7/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICPADS51040.2020.00083

***Abstract*—To cope with the increasing content requests from emerging vehicular applications, caching contents at edge nodes is imperative to reduce service latency and network traffic on the Internet-of-Vehicles (IoV). However, the inherent characteristics of IoV, including the high mobility of vehicles and restricted storage capability of edge nodes, cause many difficulties in the design of caching schemes. Driven by the recent advancements in machine learning, learning-based proactive caching schemes are able to accurately predict content popularity and improve cache efficiency, but they need gather and analyse users' content retrieval history and personal data, leading to privacy concerns. To address the above challenge, we propose a new proactive caching scheme based on peer-to-peer federated deep learning, where the global prediction model is trained from data scattered at vehicles to mitigate the privacy risks. In our proposed scheme, a vehicle acts as a parameter server to aggregate the updated global model from peers, instead of an edge node. A dual-weighted aggregation scheme is designed to achieve high global model accuracy. Moreover, to enhance the caching performance, a Collaborative Filtering based Variational AutoEncoder model is developed to predict the content popularity. The experimental results demonstrate that our proposed caching scheme largely outperforms typical baselines, such as Greedy and Most Recently Used caching.**

***Keywords*-Federated Learning, Deep Learning, Internet of Vehicles, Edge Caching, AutoEncoder**

I. Introduction

To improve road safety and travel comfort, the Internet-of-Vehicles (IoV) has emerged as a new paradigm for intelligent transportation systems [1]. It supports a wide range of emerging vehicular applications, such as smart navigation and infotainment [2]. These applications require low network latency and substantial network resources (*e.g.*, caching, computation, and communication), which places huge challenges to the IoV. Shifting cloud computing and storage capabilities to the edge nodes of IoV has been considered as a promising approach to satisfy the diverse requirements of vehicular applications. Especially, caching popular contents at edge nodes (*e.g.*, Base Station (BS), Road side unit (RSU), vehicles) can alleviate the data traffic on backhaul links and reduce service latency.

Due to the limited caching storage at edge nodes, efficient caching schemes that manages the caching resources is necessary. Recent breakthroughs in Machine Learning (ML) facilitate many learning-based content caching schemes [3] [4] [5]. ML techniques can effectively extract hidden features and representations from users' data to accurately predict content popularity. However, conventional caching schemes cannot be directly applied in IoV, due to the inherent characteristics of IoV, *e.g.*, the high mobility of vehicles and dynamic network environment. Additionally, most of the existing learning-based caching schemes need to centrally analyse users' data to make caching decisions. This process may cause the disclosure of users' privacy. Thus, it is of paramount importance to design a learning-based caching scheme for IoV that can achieve high caching performance while protecting users' privacy.

Federated learning (FL) [6] provides a new framework for fitting ML techniques into the edge while mitigating user privacy risks. It allows a central server to cooperate with multiple vehicles to jointly train an ML model in the IoV. Vehicles upload parameters of the trained model to the central server and keep their training data locally. However, if an RSU is chosen as a central server, vehicles with high speed may pass several RSUs during the FL training process, since the coverage area of RSU is small. This may seriously affect the performance of the trained model in FL. To address this challenge, we propose a Peer-to-Peer Federated learning based proactive Caching scheme (PPFC) that is well suited to the highly dynamic IoV environments. In PPFC, a vehicle with enough computation, caching and communication resources can be selected as a central server to aggregate a global model from peers. Nearby vehicles with the same direction can then connect to this server vehicle to participate in the FL training. Compared with traditional FL, peer-to-peer FL can eliminate the issue of hand-over between RSUs, achieve lower latency and adapt to the mobility of vehicles. PPFC utilises a Collaborative Filtering based Variational AutoEncoder (CF-VAE) model to predict content popularity based on the contextual information of users for making smart caching decisions.

The main contributions of the paper are summarised as follows:

1) A peer-to-peer federated learning based proactive caching scheme is proposed to adapt to high mobility of vehicles in IoV. In the proposed scheme, a vehicle rather than a fixed edge node, acts as a central server

2690-5965/20/$31.00 ©2020 IEEE
DOI 10.1109/ICPADS51040.2020.00083

Authorized licensed use limited to: Lancaster University. Downloaded on May 06,2021 at 14:59:42 UTC from IEEE Xplore. Restrictions apply.

to aggregate ML models from nearby vehicles.
2) Due to the heterogeneous abilities of vehicles, a dual-weighted model aggregation scheme is designed to reduce the effect of straggler vehicles, in order to further improve the accuracy of the trained global model in the designed peer-to-peer FL.
3) A collaborative filtering based variational autoencoder model is proposed to predict the popularity of contents by using users' historical requests and contextual information, which can learn deep latent representations of users' characteristics, while preserving data privacy through the use of FL.

The rest of this paper is organised as follows. Section II reviews the related work. The system architecture of the proposed cache scheme is presented in Section III. Section IV describes the detailed implementation of PPFC. The performance evaluation and analysis of PPFC are provided in Section V. Section VI concludes this paper.

II. Related Work

Several caching schemes have been widely studied in IoV scenario. Zhang *et al.* [3] designed a heterogeneous information network-based content caching scheme to reduce network load and enhance the quality of experience by combining data mining techniques with the features of IoV. Ndikumana *et al.* [4] proposed a deep learning based caching scheme for IoV to reduce the delivery delay of content. To optimize caching decisions, the popularity of contents is estimated by Multi-Layer Perceptron. The age and gender of passengers are predicted by Convolutional Neural Network. Then, a k-means algorithm and binary classification are used to determine contents cached at vehicles. Park *et al.* [7] proposed a distributed proactive caching scheme in vehicular networks by taking the movement of vehicles into account. Chen *et al.* [5] developed a cooperative edge caching scheme for connected vehicles by considering the content popularity and location. Zhang *et al.* [8] introduced a mobility-aware cooperative caching framework. To better utilise the resources of vehicles, vehicles can share contents with BSs. Ainagar *et al.* [9] exploited a mobility-aware proactive caching scheme to minimise communication latency by considering the demands from users and their mobility. The effect of vehicle velocity has also been evaluated. Gad *et al.* [10] designed a hierarchical proactive caching by utilising the storages at vehicles and RSUs to minimise the vehicle communication latency. Zhang *et al.* [11] proposed a proactive caching scheme for autonomous vehicles by adopting a non-negative matrix factorization technique to estimate the preference of users. The contents at video level are stored at the core network nodes, whereas the chunk level contents are cached at edge nodes. Zhu *et al.* [12] investigated a deep reinforcement learning based approach to deal with the problem of automatic vehicle control and the selection of proactive caching action. Zhang *et al.* [13] presented a proactive caching scheme for vehicular multi-view 3D videos which utilises deep reinforcement learning to select views set and allocate cache memory.

However, these existing proactive caching schemes in IoV need to upload and centrally process users' data at a central server, which may cause the risk of sensitive data leakage and misuse, and also result in the large communication cost. To protect users' privacy and relieve the communication load, Federated learning (FL) has been regarded as a promising framework, which was firstly proposed by Google [6]. There are several works utilising FL in IoV. Samarakoon *et al.*[14] introduced a FL-based method to predict the tail distribution of the network-wide queue lengths, in order to realise the status of networks. Ye *et al.* [15] designed a FL selective model aggregation method to select participating vehicles by considering the computation capacity of vehicles and data quality at vehicles. Lu *et al.* [16] proposed a hybrid blockchain based asynchronous FL scheme to secure data sharing. Another asynchronous FL scheme was developed in [17] for resource sharing purpose, which combines differential techniques into FL to protect the privacy of local updates. Roy *et al.* [18] proposed a peer-to-peer decentralized federated learning, without a central server. However, the above works lack consideration inherent limitations of edge caching in IoV, such as mobility of vehicles, time-varying and location-dependent content popularity. The FL needs to be incorporated with edge caching in IoV to facilitate cache efficiency.

III. System Architecture

The system architecture of the proposed PPFC is shown in Fig. 1. A vehicular network is considered, which consists of a BS, RSUs and vehicles. It is a hierarchical structure. The top layer is a BS, which links to the Internet through a reliable backhaul link. In the middle layer, several RSUs are placed equidistantly at the coverage area of the BS. Each RSU connects to several vehicles that are distributed at the bottom layer. The communication between BS, RSUs and vehicles are via wireless links. In our design, both RSUs and vehicles have cache capability, because vehicles are equipped with OBUs and RSUs have cache-able servers. Users can fetch their requested contents from RSUs and vehicles, instead of the internet only. When a vehicular user requests a content, it will firstly check its own cache. If the requested content is stored locally, the vehicle can directly obtain it without any transmission. If not, this request will broadcast to neighbour vehicles. If the broadcast is responded, nearby vehicles will send the requested content to the vehicle which requested the content. Otherwise, the request will forward to the current connected RSU. If the requested content is available in the RSU, the RSU can delivery this content to the requested vehicle. If the requested content is still missing, the vehicle has to request this content from the Internet.

Authorized licensed use limited to: Lancaster University. Downloaded on May 06,2021 at 14:59:42 UTC from IEEE Xplore. Restrictions apply.

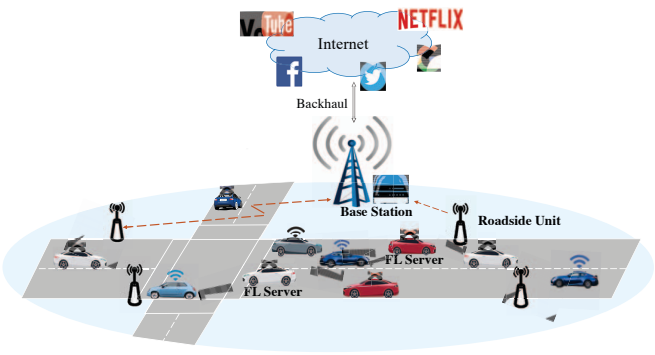

Figure 1. System architecture

Thus, vehicles fetch contents mainly in the following three ways: Vehicle-to-Vehicle (V2V) content delivery, Vehicle-to-RSU (V2R) content delivery and Vehicle-to-Internet (V2I) content delivery. V2V content delivery links two vehicles within distance γ. If vehicle k_i stores the requested content from nearby vehicle k_j, vehicle k_i can directly deliver the requested content to vehicle k_j through V2V link. RSU is another cache place in our designed PPFC. Vehicles who are located at the same coverage area of the RSU use the same frequency band. Vehicles fetch the requested contents from RSU via V2R links. V2I content delivery is the transmission between the Internet and vehicles. If the requested contents are missing in both vehicles and RSUs, it will get from the Internet with V2I transmission. Obtaining contents from vehicles and RSUs, without asking for the Internet, can significantly ease the network load. Moreover, users fetch their requested contents from near vehicles which can largely reduce latency.

Due to the limited storage resource of RSUs and vehicles, we assume that a vehicle can only store up to m contents and an RSU can cache n contents at most. To make full use of caching storage at RSUs and vehicles, designing a smart caching scheme is essential. The gain from the caching scheme highly depends on the accuracy of content popularity. However, content popularity is dynamic and hard to predict. Different vehicular users may prefer different contents and their preferences may change frequently which are influenced by location and time. The spatio-temporal variability on the popularity of contents adds substantial complexity in content caching of IoV. Moreover, the lifetime of contents in IoV is short. As a result, the cached contents are easy to be out-of-date. Thus, according to the estimated content popularity and the lifetime of contents, updating cached contents regularly is necessary to the edge caching in IoV.

We design a proactive caching scheme to make caching decision by predicting content popularity, based on the peer-to-peer federated learning. In typical FL, multiple vehicles collaboratively train a global model from their site-specific datasets under the instruction of a central server in the RSU. Instead of sending raw data to the central server in RSU for model training, vehicles only send parameters of the model to the central server. With the help of distributed training at vehicles, user privacy can be largely protected. The global model that trained in FL is a content popularity prediction model, which is utilised to make the smart caching decision. However, due to the high mobility of the vehicle and short coverage area of an RSU, vehicles with high-speed cannot complete FL training process within one RSU's coverage area. Switching between RSUs happens frequently. If the fixed RSU is chosen as the central server in FL, computation and communication costs are increased, and the accuracy of prediction is degraded. To adapt the FL framework to the IoV scenario and address the limitations and conundrums caused by the high mobility of vehicles, the peer-to-peer FL is proposed, without depending on a central server in an RSU. A vehicle can be selected as a moving central server of FL. The same direction of vehicles within one transmission hop are clustered into one group and then execute FL training within this group.

IV. Peer-to-Peer Federated Deep Learning for IoV Edge Caching

This section describes the details of our proposed proactive content caching scheme. In light of growing privacy concerns, FL is designed to collaboratively train a global ML model by using the local data at distributed vehicles. However, in the complex and dynamic IoV environments, the typical FL faces the challenge of frequently switching connected central servers and heterogeneous abilities of vehicles. To better fit FL to IoV, we proposed a peer-to-peer FL, as shown in Fig 2. Training a model in peer-to-peer FL is performed by multiple communication rounds and each communication round r consists of the following six steps:

1) **Location based vehicle selection**: To avoid frequently switching connected central servers for vehicles during the FL training process, the vehicle with sufficient computation and caching capacity can be selected as a central server. Unlike the server in the fixed RSU for typical FL, the vehicle server is a moving central server to aggregate models as well as providing caching contents to other vehicles. The same direction of vehicles with one transmission hop neighbours are chosen as participating vehicles to be involved in the FL model training.

2) **Model dissemination**: Once the server vehicle and participating vehicles K are selected, the server vehicle initialises the global ML model w_r and sends it to the participating vehicles with the aim of distributed model training at these vehicles.

3) **Learning of distributed prediction model**: Each participating vehicle utilises its own data to train the ML model, which is a Collaborative Filtering based Variational

Authorized licensed use limited to: Lancaster University. Downloaded on May 06,2021 at 14:59:42 UTC from IEEE Xplore. Restrictions apply.

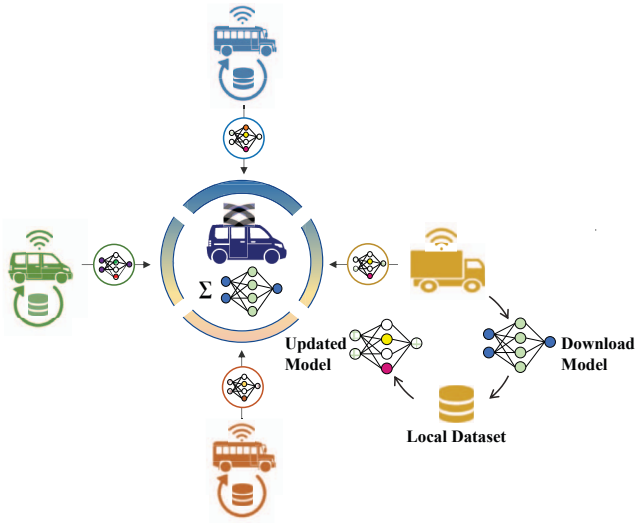

Figure 2. Peer-to-peer federated learning

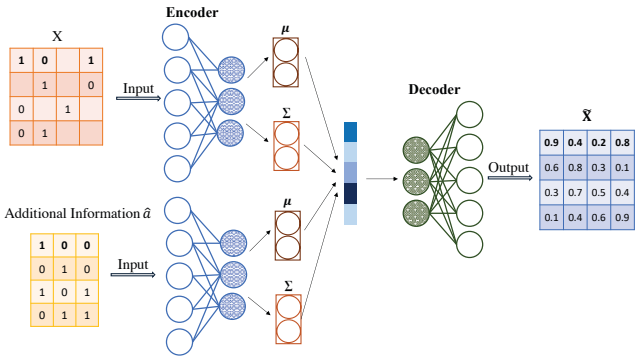

Figure 3. Collaborative filtering based variational autoencoder

Autoencoder (CF-VAE) model to predict the content popularity. It is an unsupervised learning algorithm to copy its input X to its output $\tilde{X}$, as shown in Fig. 3. The X is a user-content rating matrix, which consists of vehicular users' historical requests. The $\hat{a}$ is the matrix of the vehicular users' context information. We fed X and $\hat{a}$ into CF-VAE to learn the hidden representations Z, respectively. Then, these obtained representations are combined to reconstruct the input X. X samples variable x. The encoder $q(z \mid x)$ which is an inference neural network maps x to a Gaussian distribution and the latent variable z is estimated. The decoder, a generative neural network $p(z \mid x)$, decodes z back into x. In the generative process, our objective is to maximise the probability of each x. It can be defined as:

$$p(x) = \int p(x \mid z)\, p(z)\, dz. \tag{1}$$

$p(x \mid z)$ is parameterised with a function approximator. The likelihood $p(x \mid z)$ and the prior $p(z)$ can be formulated, while the posterior $p(z \mid x)$ requires an intractable integral over the latent space. The posterior $q(z \mid x)$ generates a distribution over the latent variables. Kullback-Leibler divergence can be used to minimise difference between $p(z \mid x)$ and $q(z \mid x)$.

$$\begin{aligned} &KL\left[q(z \mid x) \parallel p(z \mid x)\right] = \\ &\quad \mathbb{E}_{z\sim q(z|x)}\left[\log q(z \mid x) - \log p(z \mid x)\right]. \end{aligned} \tag{2}$$

Applying Bayesian inference we have

$$\begin{aligned} &\mathrm{KL}\left[q(z \mid x) \parallel p(z \mid x)\right] = \\ &\quad \mathbb{E}_{z\sim q(z|x)}\left[\log q(z \mid x) - \log p(z \mid x)\right] + \log p(x). \end{aligned} \tag{3}$$

Then, to minimise $\mathrm{KL}\left[q(z \mid x) \parallel p(z \mid x)\right]$, the Eq. (3) can be simplified as the following form:

$$\begin{aligned} \log p(x) \geq &\mathbb{E}_{z\sim q(z|x)}\left[\log p(x \mid z)\right] \\ &- \mathrm{KL}\left[q(z \mid x) \parallel p(z)\right]. \end{aligned} \tag{4}$$

where the right hand-side is the variational lower bound of VAE. The approximate posterior $q(z \mid x)$ follows a Gaussian distribution $N\left(\mu, \mathrm{diag}\left(\sigma^2\right)\right)$ where μ is the mean and σ^2 is variance. The generative network $p(x \mid z)$ and inference network $q(z \mid x)$ are trained by maximising the variational lower bound with respect to their parameters. The reparameterisation trick $z = \mu + \sigma \odot \epsilon$ can be implemented to get the unbiased estimate of low variance bound. We suppose the mean and covariance are $\mu(x)$ and $\sigma(x)$, respectively. ϵ follows $N(0, I)$, the equation can be rewritten as follow:

$$\begin{aligned} &\mathbb{E}_{q(z|x)}\left[\log p(x \mid z)\right] = \\ &\quad \mathbb{E}_{\epsilon\sim N(0,I)}\left[\log p(x \mid z = \mu + \sigma \odot \epsilon)\right], \end{aligned} \tag{5}$$

where ϵ is a vector sampled from standard Gaussian variables. With the help of the reparameterisation trick, the inference and generative networks can be trained through end-to-end backpropagation by SGD.

Once the local training of CF-VAE at vehicles is completed, the parameters of CF-VAE are sent back to the server vehicle for model aggregation.

4) **Dual-weighted model aggregation**: To improve the quality of the global model, the server vehicle constructs a new version of the global ML model by aggregating updated models from nearby vehicles with a dual-weighted method. Due to the heterogeneous abilities of vehicles, vehicles contain different amount of local data and have different learning status. Vehicles cannot equally contribute to the global model with such large differences. The effect of straggler vehicles needs to be reduced for the current FL communication round. Therefore, we introduce a dual-weighted aggregation scheme to solve this problem, which is divided into two parts: data weight and staleness weight [19]. The data weight λ_D is decided by the proportion of the local data size d_k at a vehicle k to the total data size D of all participating vehicles. The data weight of vehicle k is

Authorized licensed use limited to: Lancaster University. Downloaded on May 06,2021 at 14:59:42 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1 : The peer-to-peer federated learning based proactive caching scheme, K is the set of participating vehicles, where $k \in K$. η is the learning rate; E is the number of epoches. B is the local minibatch size;

Select A Server Vehicle and Participating Vehicles

Server Vehicle Execution:

initialise w_0
for each round r = 1,2,... **do**:
  K_r : a set of participating vehicles
  Get the parameters of the global model w_r
  for each vehicle $k \in K_r$ in parallel **do**:
    $w_r^k \leftarrow$ **VehicleUpdate**(w_r, k)
  end for
  $w_{r+1}^k \leftarrow w_r - \sum_{k=1}^{K} \lambda_D^k \lambda_S^k w_{r+1}^k$
for end
Return w_{r+1}

Participating Vehicle Execution:

Input: X, w_r
VehicleUpdate(w, k)**:**
  for each local epoch i from 1 to E **do**
    for batch $b \in B$ **do**
      Compute parameters with gradient descent:
      $w_{r+1} \leftarrow w_r - \eta \nabla l\,(w_r; b)$
    end for
  end for
Return w_{r+1}

$\lambda_D^k = \frac{d_k}{D}$, where $D = \sum_{k=0}^{K} d_k$, $d_k = |D_k|$ and D_k is the set of training samples on vehicle k. The staleness weight λ_S is influenced by the uploading time T_{up} and downloading time T_{down}, which reflects the staleness for the model. The staleness is calculated as $\varphi = T_{up} - T_{down}$. It also indicates the computing power of the vehicle. A smaller weight is given to the vehicle with a larger staleness. The staleness weight is calculated using the following exponential function [19][20]:

$$\lambda_S^k = (e/2)^{-\varphi}\,. \tag{6}$$

where e is Euler's number. Thus, the updated model is conducted with the weighted average sum:

$$w_{r+1} \leftarrow w_r - \sum_{k=1}^{K} \lambda_D^k \lambda_S^k w_{r+1}^k, \tag{7}$$

After one round of FL training is finished, vehicles overwrite their local parameters to the latest downloaded parameters and refresh their dual-weights to prepare the next round FL training.

5) **Model optimisation**: To improve the convergence of FL, an adam-based optimisation is exploited in the server vehicle. Based on the local data of server vehicle, the model will be validated. The aim of our proposed peer-to-peer FL is to minimise the loss function $\ell\,(w)$:

$$\begin{aligned} &\min_{w}\ \ell\,(w) = \sum_{k=1}^{K} L_k\,(w), \\ &\text{where}\ \ L_k\,(w) = \frac{1}{d_k} \sum_{i \in D_k} \ell_i\,(w)\,. \end{aligned} \tag{8}$$

6) **Caching decision and model update**: Based on the output of the updated model, the highest m predicted rating scores in $\tilde{X}$ are selected as the caching contents in the vehicle. The less n popular contents will cache in RSUs. Meanwhile, the server vehicle updates the global model and this model will be disseminated to all one-hop neighbour vehicles who will participate in the next FL communication round.

Above steps are repeated until an optimal model achieved at the server vehicle. The pseudo-code of PPFC is outlined in Algorithm 1.

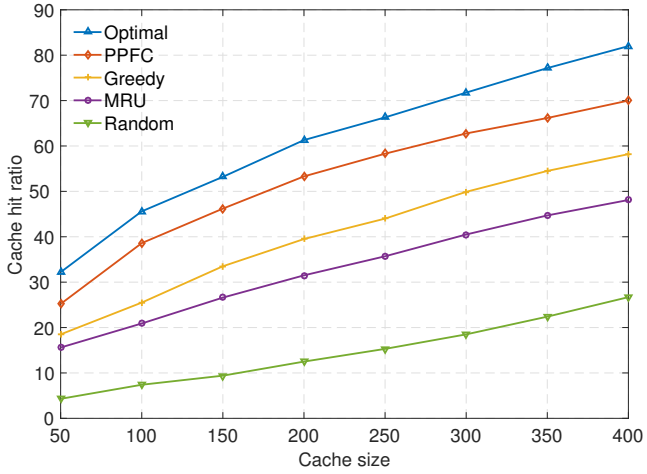

Figure 4. PPFC vs. Other reference schemes (10 vehicles)

V. Performance Evaluation

In this section, comprehensive experiments are conducted to evaluate the performance of PPFC under various IoV environments and compare the PPFC with four baseline caching schemes with respect to the cache hit ratio.

A. Experiment Settings and Dataset

We set up a networking testbed, consisting of 10 Raspberry Pi devices. Each Raspberry Pi represents a vehicle and has a local dataset to conduct learning-based prediction model training. The dataset is MovieLens 1M which contains about 1 million ratings from 6000 anonymized users on 3883 contents [21]. This dataset also involves the contextual information of users, such as, age, gender and address. Keras and TensorFlow are used to implement the CF-VAE and FL. The evaluation metric we used to measure the proposed

Authorized licensed use limited to: Lancaster University. Downloaded on May 06,2021 at 14:59:42 UTC from IEEE Xplore. Restrictions apply.

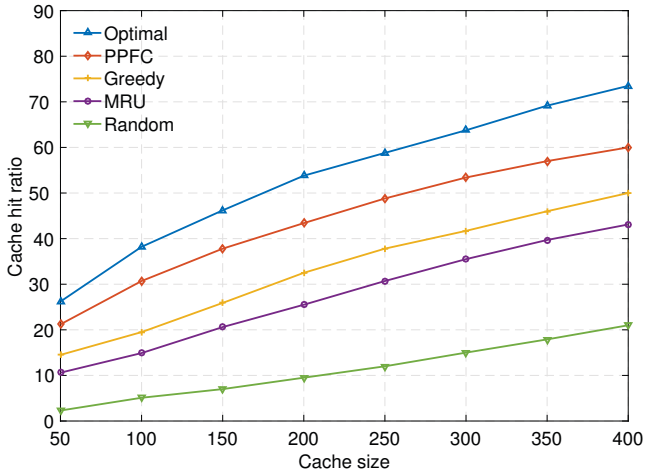

Figure 5. PPFC vs. Other reference schemes (5 vehicles)

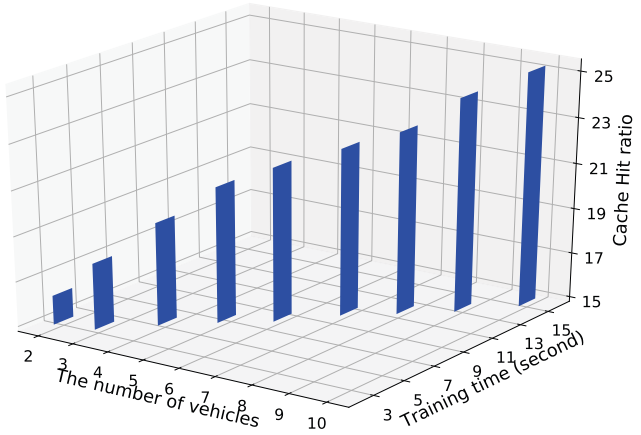

Figure 6. Vehicle density vs. Training time vs. Cache hit ratio

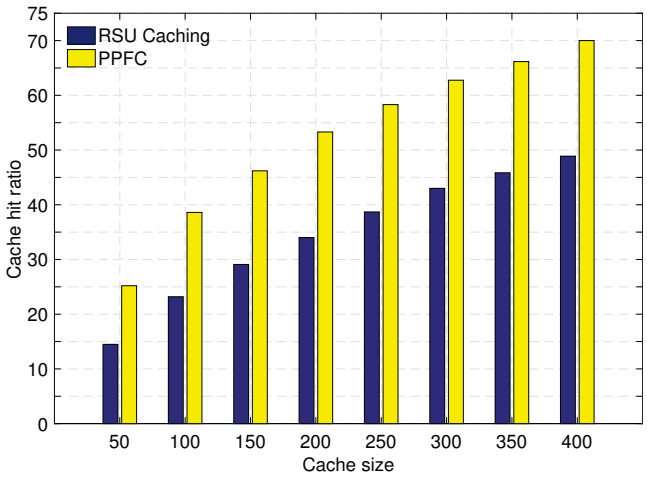

Figure 7. PPFC vs. RSU caching (10 vehicles)

PPFC is cache hit ratio, which represents the percentage of requested contents from users that enable to serve from vehicles and RSUs.

B. Experimental Results

Fig. 4 and Fig. 5 compare the performance of PPFC with other four reference caching schemes (Optimal, Greedy, MRU and Random) and show the cache hit ratio for varying cache sizes from 50 to 400 contents. They also demonstrate the impact of 10 vehicles and 5 vehicles participating in peer-to-peer FL training on cache hit ratio, respectively. Both figs exhibit the same trend. The cache hit ratios of all caching schemes increase, with the growth of cache size. The Optimal reference caching scheme presents the highest cache hit ratio, because it has a perfect knowledge of future vehicular user demands. Our proposed PPFC outperforms the other three reference caching schemes, since PPFC predicts the future popular contents for users by learning hidden features from the request of users and clustering these requests in the latent space. Greedy is a simple learning algorithm, caching the m highest previous demanded contents, but it does not consider future content popularity. MRU is the third reference caching scheme. It follows a static rule that firstly discards the most recently used contents. However, it lacks consideration of dynamically changing content popularity. The random algorithm shows the lowest cache hit ratio, which randomly selects the m contents to the cache. Compared Fig. 4 with Fig. 5, the cache hit ratio of 10 participating vehicles in the peer-to-peer FL training is higher than 5 participating vehicles. When the cache size is 50, the cache hit ratio of 10 participating vehicles is 25.2%, while 5 participating vehicles can only achieve 21%.

Fig. 6 investigates the relationship between vehicle density, training time and cache hit ratio. The vehicle density is from 2 to 10 vehicles per/km. The results demonstrate that when the cache size is 50, the cache hit ratio rises with the increase in vehicle density. When 2 vehicles participate in the peer-to-peer FL, the cache hit ratio is 16.2%. Whereas, the cache hit ratio will rise to 25.3%, if 10 vehicles attend in FL training. Meanwhile, when the number of participating vehicles changes from 2 to 10, the training time for per communication round increases from 3.98 seconds to 16 seconds. It indicates that more accurate prediction can be achieved if more vehicles participate in the FL training. It is due to that more training data and computation capacity are provided in more participating vehicles. However, it is a trade-off between the training time, vehicle density and cache hit ratio. As more vehicles attend to the FL training, the cache hit ratio improves, but the training time takes longer.

Fig. 7 and Fig. 8 depict the effectiveness of vehicle-to-vehicle caching. These experiments compare the caching performance of traditional RSU caching with the proposed

Authorized licensed use limited to: Lancaster University. Downloaded on May 06,2021 at 14:59:42 UTC from IEEE Xplore. Restrictions apply.

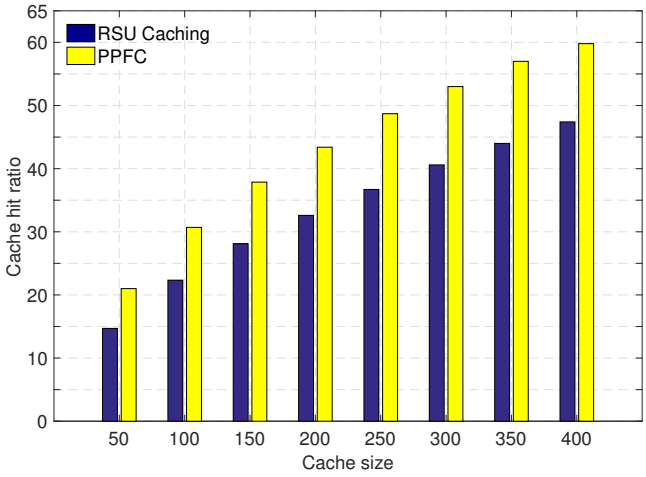

Figure 8. PPFC vs. RSU caching (5 vehicles)

PPFC that combines vehicle-to-vehicle caching and RSU caching. Fig. 7 shows the cache hit ratio for various cache sizes between 50 and 400 contents, when the number of participating vehicles in peer-to-peer FL training is set to 10. By contrast, Fig. 8 describes the results of 5 vehicles attending to the FL training. It can be seen from both Fig. 7 and Fig. 8, the PPFC demonstrates a better caching performance compared to RSU caching. When cache size is 50 and 10 vehicles participate in the FL training, the cache hit ratio of PPFC is 25.3%, while RSU caching can only reach 14.49%. Maintaining the same cache size, when the number of participating vehicles reduces to 5 vehicles, the cache hit ratio of PPFC can obtain 21%, but a similar cache hit ratio for RSU caching is achieved. For the other cache sizes, the same trend has been observed. It indicates vehicle-to-vehicle caching can improve the cache hit ratio. As the number of participating vehicles increases, more caching capacity from vehicles are brought, and therefore, the cache hit ratio rises. However, the caching performance of RSU caching is not affected by the number of participating vehicles.

VI. Conclusion

In this paper, we have proposed a new Proactive content Caching scheme on Peer-to-Peer Federated learning (PPFC) to protect vehicles' privacy, enhance caching performance and reduce latency. Due to the mobility of vehicles, a vehicle is selected as a moving central server, to ease reliance on the fixed central server in RSU and eliminate the issue of frequently hand-over between RSUs. PPFC utilises a collaborative filtering based variational autoencoder model to predict content popularity in the future and pre-fetch predicted popular contents at vehicles and RSUs to improve caching performance. Numerical results show that PPFC outperforms other reference caching schemes in terms of cache hit ratio. More vehicles participating in the peer-to-peer FL training can achieve better caching performance.

References

[1] C. Chen, C. Wang, T. Qiu, M. Atiquzzaman, and D. O. Wu, "Caching in vehicular named data networking: Architecture, schemes and future directions," *IEEE Communications Surveys & Tutorials*, 2020, DOI: 10.1109/COMST.2020.3005361.

[2] Y. Dai, D. Xu, Y. Lu, S. Maharjan, and Y. Zhang, "Deep reinforcement learning for edge caching and content delivery in internet of vehicles," in *Proc. of ICCC*. IEEE, 2019, pp. 134–139.

[3] Y. Zhang, R. Wang, M. S. Hossain, M. F. Alhamid, and M. Guizani, "Heterogeneous information network-based content caching in the internet of vehicles," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 10, pp. 10 216–10 226, 2019.

[4] A. Ndikumana, N. H. Tran, K. T. Kim, C. S. Hong *et al.*, "Deep learning based caching for self-driving cars in multi-access edge computing," *IEEE Transactions on Intelligent Transportation Systems*, 2020, DOI: 10.1109/TITS.2020.2976572.

[5] J. Chen, P. Yang, F. Lyu, X. Shen *et al.*, "Cooperative edge caching with location-based and popular contents for vehicular networks," *IEEE Transactions on Vehicular Technology*, 2020, DOI: 10.1109/TVT.2020.3004720.

[6] H. B. McMahan, E. Moore, D. Ramage, S. Hampson *et al.*, "Communication-efficient learning of deep networks from decentralized data," *in Proc. of AISTATS*, 2016.

[7] S. Park, S. Oh, Y. Nam, J. Bang, and E. Lee, "Mobility-aware distributed proactive caching in content-centric vehicular networks," in *Proc. of WMNC*. IEEE, 2019, pp. 175–180.

[8] K. Zhang, S. Leng, Y. He, S. Maharjan, and Y. Zhang, "Cooperative content caching in 5G networks with mobile edge computing," *IEEE Wireless Communications*, vol. 25, no. 3, pp. 80–87, 2018.

[9] Y. AlNagar, S. Hosny, and A. A. El-Sherif, "Towards mobility-aware proactive caching for vehicular ad hoc networks," in *Proc. of WCNCW*. IEEE, 2019, pp. 1–6.

[10] M. E. Gad, S. Hosny, B. Mokhtar, and A. A. El-Sherif, "Hierarchical proactive caching for vehicular ad hoc networks," in *Proc. of NILES*, vol. 1. IEEE, 2019, pp. 150–153.

[11] Z. Zhang, C.-H. Lung, M. St-Hilaire, and I. Lambadaris, "Smart proactive caching: Empower the video delivery for autonomous vehicles in icn-based networks," *IEEE Transactions on Vehicular Technology*, 2020, DOI: 10.1109/TVT.2020.2994181.

[12] Z. Zhu, Z. Zhang, W. Yan, Y. Huang, and L. Yang, "Proactive caching in auto driving scene via deep reinforcement learning," in *Proc. of WCSP*. IEEE, 2019, pp. 1–6.

Authorized licensed use limited to: Lancaster University. Downloaded on May 06,2021 at 14:59:42 UTC from IEEE Xplore. Restrictions apply.

[13] Z. Zhang, Y. Yang, M. Hua, C. Li, Y. Huang, and L. Yang, "Proactive caching for vehicular multi-view 3d video streaming via deep reinforcement learning," *IEEE Transactions on Wireless Communications*, vol. 18, no. 5, pp. 2693–2706, 2019.

[14] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah, "Distributed federated learning for ultra-reliable low-latency vehicular communications," *IEEE Transactions on Communications*, vol. 68, no. 2, pp. 1146–1159, 2019.

[15] D. Ye, R. Yu, M. Pan, and Z. Han, "Federated learning in vehicular edge computing: A selective model aggregation approach," *IEEE Access*, vol. 8, pp. 23 920–23 935, 2020.

[16] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Blockchain empowered asynchronous federated learning for secure data sharing in internet of vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4298–4311, 2020.

[17] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Differentially private asynchronous federated learning for mobile edge computing in urban informatics," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2134–2143, 2019.

[18] A. G. Roy, S. Siddiqui, S. Pölsterl, N. Navab, and C. Wachinger, "Braintorrent: A peer-to-peer environment for decentralized federated learning," *arXiv preprint arXiv:1905.06731*, 2019.

[19] X. Lu, Y. Liao, P. Lio, and P. Hui, "Privacy-preserving asynchronous federated learning mechanism for edge network computing," *IEEE Access*, vol. 8, pp. 48 970–48 981, 2020.

[20] Y. Chen, X. Sun, and Y. Jin, "Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation," *IEEE Transactions on Neural Networks and Learning Systems*, 2019, DOI: 10.1109/TNNLS.2019.2953131.

[21] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *ACM Transactions on Interactive Intelligent Systems*, vol. 5, no. 4, p. 19, 2016.

Authorized licensed use limited to: Lancaster University. Downloaded on May 06,2021 at 14:59:42 UTC from IEEE Xplore. Restrictions apply.