Federated Learning with Variational Autoencoder for Popularity Profile Prediction

Minkyun Ahn[†] and Minseok Choi[†]
[†]Department of Electronic Engineering, Kyung Hee University, Yongin, South Korea
E-mails: amky94@khu.ac.kr, choims@khu.ac.kr

***Abstract*—Motivated by increasingly exploding data traffic of online video services, the prediction of the popularity profile of video contents becomes very important for network traffic prediction, recommendation systems, and wireless caching. This paper proposes a federated learning-based popularity prediction scheme using a variational autoencoder (VAE), which copes with the situation where users are moving and/or their data privacy should be protected. Users are participants of federated learning, and the VAE model is trained by user's own request history; afterwards, randomly generated samples from the pretrained decoder of VAE can mimic the original popularity profile. We adopt the MovieLens dataset to validate the proposed model, and experimental results show that our scheme predicts the popularity profile almost perfectly.**

***Index Terms*—Federated learning, Variational autoencoder, Popularity prediction**

I. Introduction

With multimedia services making up 75% of global data traffic [1], a multitude of techniques have been investigated for video traffic, which constitutes a significant portion of the total multimedia service traffic. Analyzing traffic patterns in online video services has revealed characteristics where a minority of content accounts for the majority of traffic [2]. The probability of users requesting specific videos, i.e., the content popularity profile, has been mathematically modeled using the Zipf distribution [3]. Furthermore, the authors of [4] have delved into fine-grained mathematical modeling of user-request probabilities for content based on the real-world dataset. However, content evolves over time, with new content emerging and old content vanishing. Moreover, popularity distributions can undergo significant shifts over time due to social trends. In addition, within limited local regions, user mobility can also influence the distribution of content popularity within that region. As a result, approximating popularity distribution with static mathematical distribution models can lead to substantial disparities when compared with actual datasets. To address this challenge, researches on deep learning techniques to predict network traffic patterns and time-varying content popularity profile have been actively conducted [5], [6].

In accordance with the stringent regulation law [7], users' content request information has been considered as privacy-sensitive data, and this law mandates that even when users request and receive content through base stations (BSs), the BSs are prevented from collecting such data. Beyond legal requirements, users could also choose not to share their personal information as well as their own request history. Additionally, many users come into and leave the coverage region of the BS in cellular systems, resulting in the lacking access of the BS to the users' content request history. In such scenarios, due to restrictions on data collection by base stations or parameter servers, it becomes necessary to leverage distributing machine learning (ML) over edge devices, where users participate in training. Federated learning, as one of the most popular distributed learning techniques, has demonstrated the capability to converge to the result of a centralized learning while ensuring privacy protection [8]. In federated learning, users conduct training processes separately and contribute to training a shared global model.

In this respect, there have been extensive studies on federated learning-based popularity prediction; however, most of them focus on federated learning over BSs or edge servers, not individual user devices. In [9], edge nodes gather the user request information and train a global model in a federated setting using convolutional recurrent neural network (CRNN) for predicting the future popularity. The context-aware popularity prediction policy is learned by federated learning over multiple fog access points (F-APs) based on the estimated user preference which is separately learned at the client side in [10]. In [13], variational autoencoder (VAE) is adopted to estimate the data distribution from users' training datasets. The request history of multiple users is an input of the VAE to predict the popularity of all contents including unseen ones, which means that users' private data needs to be collected to train the VAE. However, these works need to gather the raw data of users at the central node, which reveals the user privacy. On the other hand, the authors of [11] model the time-varying heterogeneous user preference using the Zipf distributions with different factors that change over time based on a Markov chain. Then, each user participates in federated learning by learning its own local popularity, but this method is not evaluated with the real-world dataset. Instead of predicting preference and popularity separately, the weighted summation of user preference and content popularity is predicted in [12] to secure users' preferred content lists, and Lastfm-1K datasetis adopted to validate their own model. However, Lastfm-1K dataset is the music request dataset which is not related to the video request, and they utilize the simple fully connected neural networks.

This paper proposes a federated learning approach with VAE for predicting the content popularity profile under the MovieLens 1M dataset which is relatively sparse than Lastfm-1K dataset. We consider the scenario in which users' raw

979-8-3503-1327-7/23/$31.00 ©2023 IEEE

data should not be shared with any other nodes to preserve their privacy; therefore, each user is a participant of federated learning and trains a local model based on its own request history only. Simulation results show that the predicted popularity profile is very close to the real data, and also the trained model anticipates the movie files even requested by users who did not participate in federated learning.

II. System Model

A. Problem Setup: Popularity Prediction

This paper considers the scenario in which there are M edge servers (ESs) having their own coverage regions and these regions are not overlapped. In the region of ES m, there are K_m users of the online video service. Users individually request contents to their ESs, and the ES should respond to these requests. We assume that the ES proactively caches at most L popular contents with the limited storage size in advance of user requests. Therefore, when a user requests a video file, the ES can directly provide the desired content to the user if it is cached, and this event is called as *cache hit*. Otherwise, the request is forwarded to the central server which has an access to the file library; therefore, the server delivers the content through the ES. This caching technique significantly reduces the content delivery latency as well as backhaul costs for communicating with the central server. Caching the most popular contents is an intuitively reasonable method; however, this requires the file popularity distribution that is now known in advance in practical scenarios. To address this challenge, this paper proposes the deep learning-based popularity prediction approach.

Motivated by the increasing concerns of data privacy and stringent regulation law about this concern [7], this paper considers the video users who are not willing to let their previous video viewing records open to any other users and edge servers. Accordingly, our objective is to predict the popularity distribution of content across all users in the regions of a single edge server without directly obtaining individual user content request history. Federated learning is the key distributed learning technology to train a global model without gathering local training data separately stored in user devices. Additionally, since not all users agree to participate in federated learning, we predict the videos likely to be requested by other users based on the estimated popularity profile using the deep neural network (DNN). In cellular networks, users with mobility can come into or leave the given region of the edge server. The popularity prediction method could be also applied to estimate the new coming users' future requests.

B. Federated Learning

This subsection explains the federated learning process of a single edge server m and $K_m = K$ users in its coverage region. Let $\mathbf{x}_k$ denote the dataset of user k for all $k \in \{1, 2, \cdots, K\}$, which represents the user k's content request history in our problem setup. The union of all users' datasets is denoted by $\mathbf{X} = \sum_{k=1}^{K} \mathbf{x}_k$. The goal of federated learning is to train a global model $\mathbf{w}^*$ which minimizes the following global loss function:

$$\min_{\mathbf{w}} F(\mathbf{w}) = \min_{\mathbf{w}} \sum_{k=1}^{K} \frac{|\mathbf{x}_k|}{|\mathbf{X}|} F_k(\mathbf{w}), \tag{1}$$

where $F_k(\mathbf{w})$ is the local loss function of data samples in client k which is defined as $F_k(\mathbf{w}) = \frac{1}{|\mathbf{x}_k|} L(\mathbf{x}_k, \mathbf{w})$. According to FedAvg algorithm [8], at the beginning of each communication round, the subset of users, denoted by $\mathcal{S}$, is randomly selected, and they receive the model $\mathbf{w}(t)$ from the ES. Then, each user k performs E local updates with the initial model $\mathbf{w}_k(t) = \mathbf{w}(t)$ as expressed by

$$\mathbf{w}_k(t+i+1) = \mathbf{w}_k(t+i) - \eta \nabla F_k(\mathbf{w}_k(t+i)), \tag{2}$$

for all $i \in \{0, 1, \cdots, E\}$, where η is the learning rate and ∇F_k represents the gradient of loss. After K local updates, users upload the learned weights to the ES, and the ES aggregates them in accordance with $\mathbf{w}(t+E) = \sum_{k \in S} \frac{|\mathbf{x}_k|}{\sum_{k \in S} |\mathbf{x}_k|} \mathbf{w}_k(t+E)$ which is a global model. This process iterated to make the global model converge to the optimal model as shown in [14].

III. Popularity Prediction by Variational Autoencoder

A. Variational Autoencoder

We adopt VAE as a deep learning model to train prediction of the content popularity distribution over the target region because VAE is powerful to extract the latent features from input data and generate samples belonging to the probability distribution of input data. VAE is a type of autoencoder (AE) that fundamentally learns to compress input data into a lower-dimensional space and then reconstructs it to its original dimensions, specializing in the extraction and compression of data features. However, different from conventional AE, VAE is capable of generative modeling by learning the inherent probabilistic distribution within the data. Accordingly, VAE can produce outputs that possess similar characteristics to the input data. For this purpose, the VAE structure fundamentally consists of an encoder and a decoder. It utilizes Bayesian probabilistic inference to learn the probabilistic distribution of the desired population [18].

1) Encoder: The encoder functions to extract features from input $\mathbf{x}$ by compresses this into a latent vector, denoted as $\mathbf{z}$. To accomplish this, the encoder approximates the distribution using the posterior probability $q(\mathbf{z}|\mathbf{x})$ of $\mathbf{z}$ given $\mathbf{x}$. Generally, the encoder output is assumed to follow a Normal distribution; therefore, the output consists of the mean $\boldsymbol{\mu}$, and the standard deviation $\boldsymbol{\sigma}$. Consequently, the latent vector is derived from $\boldsymbol{z} = \boldsymbol{\mu} + \boldsymbol{\sigma}^{\mathbf{2}} \cdot \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim N(0, 1)$ used for the reparameterization trick to ensure differentiability.

2) Decoder: The decoder serves to reconstruct the input (i.e., latent vector $\mathbf{z}$) into its original dimensions. To accomplish this, the distribution is approximated using the posterior probability $p(\mathbf{x}|\mathbf{z})$. In other words, given $\mathbf{z}$ by the encoder, the decoder generates the data, denoted as $\mathbf{x}'$, which has the same dimensions as $\mathbf{x}$ and exhibits similar features.

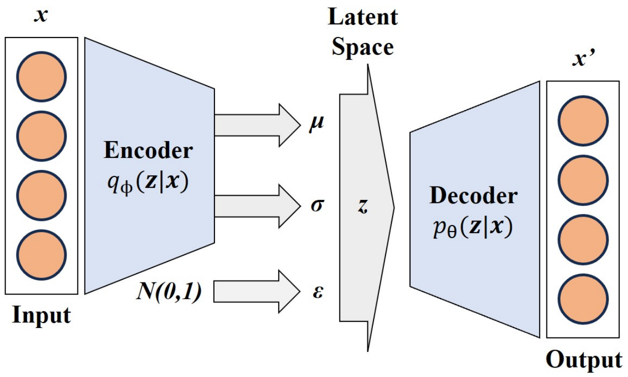

Fig. 1: Architecture of VAE

3) Loss Function: When training the decoder and encoder, the loss function is derived from the marginal log-likelihood using Bayes' theorem and the KL divergence, as given by

$$\log p_\theta(\mathbf{x}) = \int q_\phi(\mathbf{z}|\mathbf{x}) \log p_\theta(\mathbf{x}) d\mathbf{z} \\ - KL(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) + KL(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}|\mathbf{x})) \quad (3)$$

$$\geq \int q_\phi(\mathbf{z}|\mathbf{x}) \log p_\theta(\mathbf{x}) d\mathbf{z} - KL(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})). \quad (4)$$

Since the KL divergence is always non-negative, it is apparent that one can derive (4) as a lower bound on (3). This is referred to as the Evidence Lower Bound, and it is applied as an optimization criterion in VAE. In (4), $\int q_\phi(\mathbf{z}|\mathbf{x}) \log p_\theta(\mathbf{x}) d\mathbf{z}$ and $KL(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))$ are the reconstruction error and regularization error, respectively [18]. Here, the reconstruction error is an element exhibiting AE characteristics, reflecting how closely $\mathbf{x}'$ has been reconstructed to $\mathbf{x}$. The regularization error involves imposing a prior to ensure the assumption that $\mathbf{z}$ induced by the encoder for input $\mathbf{x}$ follows the Gaussian distribution.

B. Data Preprocessing

We utilized the MovieLens 1M dataset [15], which contains 1,000,209 ratings on 3952 movies made by 6040 users. Although this is the rating information, many studies considered this as the movie request data and predicted their popularity profile. Since the MovieLens dataset has individual user's data, it is appropriate for our federated learning scenario in which each user trains a local model based on its own request data only. In addition, zip-code data is included in the MovieLens dataset, and we can observe the regional preference and test the prediction model for various popularity profiles depending on the spatial characteristics. However, in a single zip-code, the number of data samples is not sufficient to train a deep learning model. Therefore, we cluster multiple zones based on their geographic closeness. Specifically, all the users having the same first two digits of the zip-code are considered to be located in the region of the same edge server. Among 3952 movies, a substantial number of movies have never been requested or watched only once. These the least popular movies do not affect the popularity profile very much, but they inject too excessive variance in the data statistics and make the training difficult. Accordingly, we excluded the corresponding movie IDs that are not requested by 10% of the total users from the dataset. For the same reason, users whose request numbers are smaller than 25 are neglected. Since the MovieLens dataset actually contains the rating information, it is essential to note that every user rates (i.e., requests) a single movie ID once at most. Consequently, the training data of user k becomes the following binary vector $\mathbf{x_k} = [b_0^k, b_1^k, ..., b_{N-1}^k, b_N^k]$, where $b_i^k \in \{0, 1\}$ for all $i \in \{0, 1, \cdots, N\}$ which represents the indicator of whether that movie ID i is requested by user k. Here, N is the number of users after filtering users whose request numbers are very small.

C. Training and Inference

In each zone, the designated edge server and users in this clustered region perform the federated learning algorithm with VAE. Each user k utilizes a full batch of its dataset $\mathbf{x_k}$ to train a local VAE model. The request history of a user can be divided into multiple mini-batches; however, since the input binary vector is sparse, each mini-batch vector could have very small number of ones, which spoils the training process. After E local updates, users upload their local VAE models to the ES, and the ES aggregates the uploaded model parameters to generate a global model.

After completing federated learning, Monte Carlo simulation is conducted with the global VAE decoder to predict the content popularity profile. As the VAE decoder is fundamentally a generator, the output of the decoder can inherently have a slight variance from our desired result. Since the decoder generates a sample, we first repeatedly obtaining output samples under the identical condition and derive the final popularity distribution by averaging output samples. To extract sufficiently many outputs from the decoder, random numbers sampled from a standard Normal distribution are fed into the decoder as the latent vector.

D. Privacy Issue of Federated Learning with Variational Autoencoder

While VAE excels at predicting the original probability distribution from a small number of samples, this inherently poses a fundamental issue that can make the user privacy revealed. Although federated learning prevents users' content request history from being directly shared with the edge server, the well-trained VAE decoder can almost perfectly reconstruct users' training data. This stems from a problem occurring during training VAE, where its input data is set as the target of VAE. This fundamentally challenges the very reason for employing federated learning.

To address this problem, two main approaches can be considered: 1) preventing the edge server from accessing the full decoder model of the trained VAE, and 2) ensuring user data cannot be predicted from the local VAE decoder model. In the first approach, the edge server can upload only

Algorithm 1: Federated Learning with Variational Autoencoder

```
ES executes:
    initialize model $\mathbf{w}(0)$
    Set $\mathbf{w}_k(0) = \mathbf{w}(0),\ k \in S$
for global epoch $t = 0, 1, \ldots$ do
    for client $k \in S$ parallel do
        $\mathbf{w}_k \leftarrow \mathbf{w}(t)$
        for local epoch $e = 0, 1, \ldots$ do
            $\mathbf{w}_k \leftarrow \mathbf{w}_k - \eta \nabla L(\mathbf{x}_k, \mathbf{w}_k)$
        end
        $\mathbf{w}(t+1) = \sum_{k \in S} \frac{|\mathbf{x}_k|}{\sum_{k \in S} |\mathbf{x}_k|} \mathbf{w}_k(t).$
    end
end
for $i = 0, 1, \ldots, N$ do // Monte Carlo simulation
    $p_i \leftarrow s_i$ // sampling from decoder
end
$p \leftarrow \frac{1}{N} \sum_{i=1}^{N} p_i$
```

a portion of the local VAE decoder model for generating a global model, while the remaining split model can be aggregated by the way of decentralized federated learning [16]. Users transmit the VAE decoder model fragments that are not uploaded to the edge server to all nearby users and receive their model fragments. This process is iterated multiple times, with users directly aggregating the received VAE decoder model fragments. However, when the number of users is large, decentralized federated learning could not aggregate all users' model fragments or causes long delays due to asynchronous aggregations of users. Nonetheless, by adopting an aggregation method of decentralized learning only for the output layer or a single hidden layer of the decoder and not disclosing information about that layer to the edge server, secure model aggregation can still be achieved.

The second approach involves applying differential privacy (DP) to uploading local VAE decoder models to the edge server and their global aggregation. DP is a technique that adds artificial noise to model parameters, preventing deep leakage of information about training data from the model or gradient [17]. However, in this scenario, the objective of using DP is slightly different that DP can also be employed to prevent the accurate parameter set of the local VAE decoder model from being shared with the edge server, thus ensuring user private data cannot be recovered. Detailed exploration of this aspect is left for our future work.

IV. Experimental Results

In our experimental setup, the VAE architecture consists of a hidden layer size of 256 and a latent space size of 32. Gumbel-softmax function with a temperature of 15 is employed for applying the canonical reparameterization trick and generating probabilistic outputs [19]. In federated learning, $E = 10$ local epochs in each communication round and total 100 rounds are conducted. Three different zones whose first two digits of their zip-codes are 55, 94, and 60, are considered to evaluate the proposed method. Specifically, 183 users, 149 users, and 128 users are assigned for zones 1, 2, and 3, respectively. We simulate the entire experiments with Windows11, Python v3.10.10, PyTorch v2.0.0, CUDA v11.8, CuDNN v8.8.1 for ML software, and NVIDIA 4070 Laptop, Intel i9-13900H, and RAM 32GB for hardware.

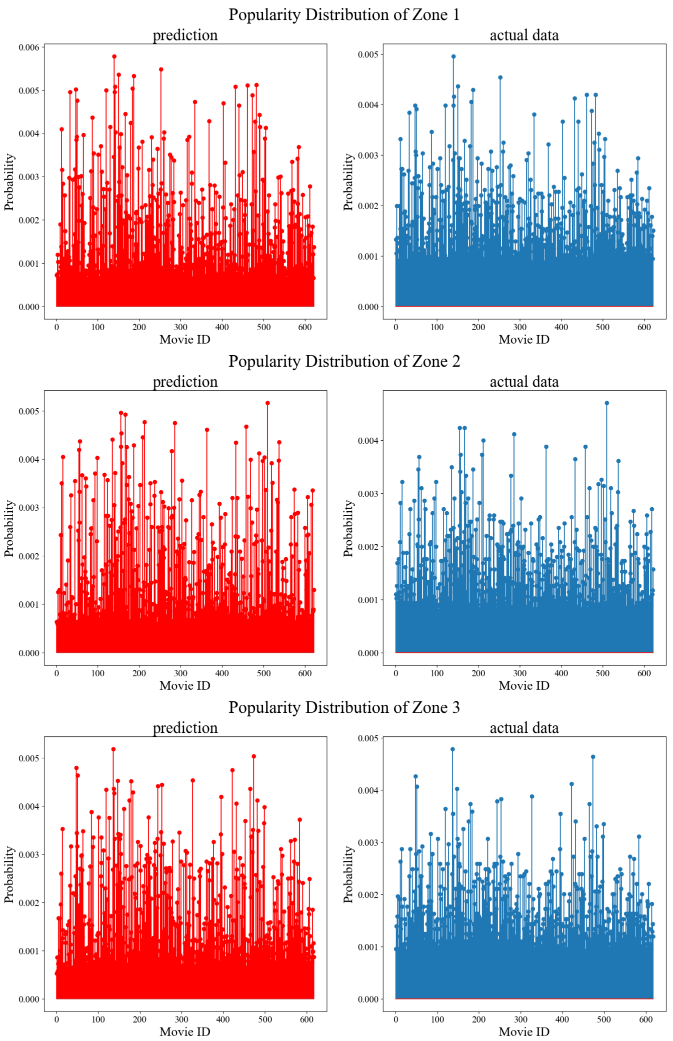

Fig. 2: Popularity prediction vs. actual per zone

In Fig. 2, the predicted popularity distributions (left) and real data statistics (right) of three zones are described. We can observe that the overall shape is similar between the prediction and actual distribution. In the predicted distribution, probabilities of popular movies are exaggerated and those of unpopular ones are underestimated, compared to the real one. This issue can be partially addressed by fine-tuning the temperature value of the gumbel-softmax function. Finally, the result of our federated learning with VAE achieves the prediction success rate of 90% in the top-100 most popular movie list. To measure the difference between the predicted popularity distribution and real one, the mean squared error (MSE) values are calculated for all zones in Table I. Given this relatively low MSE and high prediction success rate in top-100 movie list, we can conclude that the model's predictions are notably accurate.

TABLE I: MSE of the predicted distribution

	Zone 1	Zone 2	Zone 3
MSE	8.825×10^{-5}	8.508×10^{-5}	7.667×10^{-5}

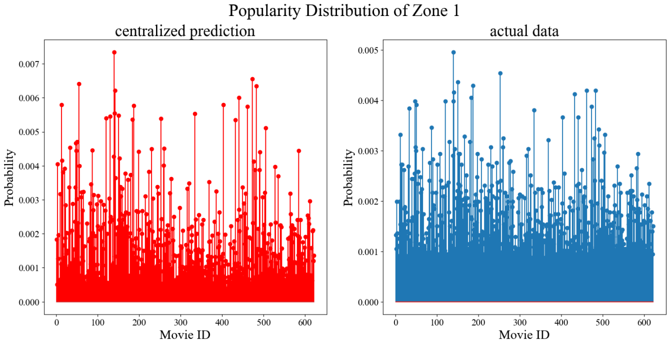

Fig. 3: Centralized prediction vs. actual

We also conduct the comparison work with centralized learning which trains a single VAE model based on all users' the movie request lists without considering the data privacy. Fig. 3 shows the comparison of the popularity distribution predicted by centralized learning with the real one. However, it shows larger differences with the actual data, compared to our federated learning approach. Thus, we can observe that federated learning demonstrates superior results compared to centralized learning when training a VAE model using the MovieLens dataset. This is because the target of VAE is the same as its input data and its objective is to mimic the distribution of input data, which makes the trained model biased to the training data. The goal of conventional federated learning is generalization, i.e., training a global model that also infers the unseen input data very well. On the other hand, in our scenario, we assume that request patterns of other users who do not participate in federated learning will follow the same popularity distribution. Nevertheless, this potential of federated learning for unsupervised learning whose target is the input data needs to be investigated more, and we leave this as future work.

Lastly, we evaluate how well our proposed popularity prediction algorithm estimates the requested movies of users who did not participate in the training process. In this experiment, zone 1 is considered, and 165 users participate in federated learning, but the remaining 18 users are excluded in the training process and used for testing the trained VAE model. The results, which show how similar the most popular movie list obtained by the trained VAE model is to the movie request list of 18 non-participant users, are presented in Table II. Since each user requested a different number of movies, the VAE model generates a list by selecting movies equal to the number of movies requested by the user, arranged in order of popularity. Table II then shows cache hit ratios for 18 non-participant users if the edge server caches the most popular movie list generated by the VAE decoder. Three different sets represent the cases in which participants of federated learning are differently selected. From these results, it can be realized that caching, based on the learned popularity distribution through federated learning with VAE, is considerably effective even for users who do not participate in learning within a zone.

TABLE II: Cache hit ratio for content requests

Client	Set 1 (%)	Set 2 (%)	Set 3 (%)
1	57.93	49.14	36.53
2	75.43	46.56	43.75
3	47.23	84.27	42.29
4	45.61	45.45	25.41
5	51.72	53.19	44.33
6	26.00	34.07	40.38
7	19.72	42.34	32.00
8	37.17	28.75	60.14
9	31.62	17.65	36.26
10	58.66	29.90	56.98
11	58.40	41.73	49.57
12	41.27	41.67	57.78
13	30.95	43.27	46.20
14	26.23	33.63	47.34
15	42.27	50.20	41.82
16	65.11	49.12	41.40
17	71.14	53.36	22.22
18	43.75	29.03	32.26
Avg	46.12	42.60	42.04

V. Conclusion

This paper proposes a federated learning approach with VAE to predict the content popularity profile, and we evaluate the prediction performances of our method based on the real-world dataset, specifically, MovieLens 1M dataset. VAE is efficient to anticipate the distribution even with the very sparse binary vector data indicating whether the video was seen or not. Also, we observe that federated learning could be more powerful to predict the complicated probability distribution using the model structure whose target is the same as the input, e.g., VAE, than centralized learning. However, there exists a risk that the user's private data could be easily revealed by the VAE decoder, and we present the approaches that can deal with this challenge.

Acknowledgment

This work was supported in part by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2021-0-02201, Federated Learning for Privacy-Preserving Video Caching Networks), in part by the National Research Foundation of Korea(NRF) grant funded by the Korea govenment(MSIT) (NRF-2022R1C1C1010766), and in part by NRF grant funded by MSIT (No. 2022R1A4A3033401).

References

[1] Cisco, "Cisco visual networking index: Global mobile data traffic forecast update, 2017-2022 White Paper," 2019. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/serviceprovider/visual-networking-index-vni/white-paper-c11-738429.html

[2] X. Cheng, J. Liu, and C. Dale, "Understanding the characteristics of Internet short video sharing: A YouTube-based measurement study," *IEEE Trans. Multimedia*, vol. 15, no. 5, pp. 1184–1194, Aug. 2013.

[3] N. Golrezaei, P. Mansourifard, A. F. Molisch, and A. G. Dimakis, "Basestation assisted device-to-device communications for high-throughput wireless video networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 7, pp. 3665–3676, Jul. 2014.

[4] M. -C. Lee, A. F. Molisch, N. Sastry and A. Raman, "Individual Preference Probability Modeling and Parameterization for Video Content in Wireless Caching Networks," *IEEE/ACM Transactions on Networking*, vol. 27, no. 2, pp. 676-690, April 2019.

[5] N. Ramakrishnan and T. Soni, "Network Traffic Prediction Using Recurrent Neural Networks," 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, USA, 2018, pp. 187-193.

[6] N. Garg, M. Sellathurai, V. Bhatia, B. N. Bharath and T. Ratnarajah, "Online Content Popularity Prediction and Learning in Wireless Edge Caching," *IEEE Transactions on Communications*, vol. 68, no. 2, pp. 1087-1100, Feb. 2020.

[7] B. Custers, A. Sears, F. Dechesne, I. Georgieva, T. Tani, and S. van der Hof, *EU Personal Data Protection in Policy and Practice*. Hague, The Netherlands: TMC Asser Press, 2019.

[8] McMahan, B., Moore, E., Ramage, D., Hampson, S. and Arcas, B.A.y.. (2017). Communication-Efficient Learning of Deep Networks from Decentralized Data. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics¡/i¿, in ¡i¿Proceedings of Machine Learning Research* 54:1273-1282 Available from https://proceedings.mlr.press/v54/mcmahan17a.html.

[9] Y. Li, S. Hu and G. Li, "CVC: A Collaborative Video Caching Framework Based on Federated Learning at the Edge," *IEEE Transactions on Network and Service Management*, vol. 19, no. 2, pp. 1399-1412, June 2022.

[10] Y. Jiang, Y. Wu, F. -C. Zheng, M. Bennis and X. You, "Federated Learning-Based Content Popularity Prediction in Fog Radio Access Networks," *IEEE Transactions on Wireless Communications*, vol. 21, no. 6, pp. 3836-3849, June 2022.

[11] C. Zheng, S. Liu, Y. Huang, W. Zhang and L. Yang, "Unsupervised Recurrent Federated Learning for Edge Popularity Prediction in Privacy-Preserving Mobile-Edge Computing Networks," *IEEE Internet of Things Journal*, vol. 9, no. 23, pp. 24328-24345, 1 Dec.1, 2022.

[12] K. Qi and C. Yang, "Popularity Prediction with Federated Learning for Proactive Caching at Wireless Edge," *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, Seoul, Korea (South), 2020, pp. 1-6.

[13] Z. Yu, J. Hu, G. Min, Z. Wang, W. Miao and S. Li, "Privacy-Preserving Federated Deep Learning for Cooperative Hierarchical Caching in Fog Computing," *IEEE Internet of Things Journal*, vol. 9, no. 22, pp. 22246-22255, 15 Nov.15, 2022.

[14] F. Haddadpour and M. Mahdavi, "On the Convergence of Local Descent Methods in Federated Learning," 2019, available at: arXiv:1910.14425.

[15] F. Harper and J. Konstan, "The movielens datasets: History and context," *ACM Transactions on Interactive Intelligent Systems*, vol. 5, no. 4, 2015.

[16] C. Hu, J. Jiang and Z. Wang, "Decentralized Federated Learning: A Segmented Gossip Approach," 2019, available at: arXiv:1908.07782.

[17] K. Wei et al., "Federated Learning With Differential Privacy: Algorithms and Performance Analysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454-3469, 2020.

[18] Diederik P Kingma and Max Welling, "Auto-encoding variational bayes," arXiv preprint arXiv:1312.6114, 2013.

[19] E. Jang, S. Gu and B. Poole, "Categorical reparameterization with Gumbel-Softmax", Proc. Int. Conf. Learn. Representations, 2017.