Improving Response Time of Home IoT Services in Federated Learning

Dongjun Hwang*  
Chungnam National University  
Daejeon, Republic of Korea  
enter031@cnu.ac.kr

Hyunsu Mun  
Chungnam National University  
Daejeon, Republic of Korea  
munhyunsu@cnu.ac.kr

Youngseok Lee  
Chungnam National University  
Daejeon, Republic of Korea  
lee@cnu.ac.kr

ABSTRACT

For intelligent home IoT services with sensors and machine learning, we need to upload IoT data to the cloud server which cannot share private data for training. A recent machine learning approach, called federated learning, keeps user data on the device in the distributed computing environment. Though federated learning is useful for protecting privacy, it experiences poor performance in terms of the end-to-end response time in home IoT services, because IoT devices are usually controlled by remote servers in the cloud. In addition, it is difficult to achieve the high accuracy of federated learning models due to insufficient data problems and model inversion attacks. In this paper, we propose a local IoT control method for a federated learning home service that recognizes the user behavior in the home network quickly and accurately. We present a federated learning client with transfer learning and differential privacy to solve data scarcity and data model inversion attack problems. From experiments, we show that the local control of home IoT devices for user authentication and control message transmission by the federated learning clients improves the response time to less than 1 second. Moreover, we demonstrate that federated learning with transfer learning achieves 97% of accuracy under 9,000 samples, which is only 2% of the difference from centralized learning.

CCS CONCEPTS

• Human-centered computing → Ubiquitous and mobile computing systems and tools;

KEYWORDS

Internet of Things (IoT), federated learning, transfer learning, differential privacy, response time

1 INTRODUCTION

Internet of Things (IoT) equipped with sensors and machine learning has been explosively popular. Home IoT devices such as built-in sensors, cameras, light bulbs, speakers, door locks, or window chains are managed by a smartphone for automation services. Intelligent IoT services increase the efficiency and the convenience to users. For example, a smart bulb like Philips Hue can change light colors. In addition, a smart speaker such as Google Nest Hub is connected to IoT devices controlled by users through voice commands.

Home IoT services are often vulnerable to privacy problems because they can be accessed from a remote server in the cloud and their data in the cloud can be exposed to the external attacks. As home IoT data contains personal information, it is difficult to share the private data publicly. Today, many countries have laws or regulations to protect privacy. General Data Protection Regulation (GDPR) is issued by the European Union for data privacy and security. In particular, as most IoT services depend on the centralized cloud, information leakage might be possible. It is an important challenge to address the personal data protection in home IoT services.

A recent machine learning approach, called federated learning (FL), protects user data by keeping them on the device in a distributed computing environment. In the federated learning model, each client performs the local learning job on the device, and then it uploads only the parameters of the local model to the FL server. A FL server aggregates all parameters to compile the new global model. Federated learning protects privacy because only the parameters of the model are shared. Google demonstrate Gboard application in federated learning to predict next words typed by a user. As words are stored on the device, there is no risk of data leakage to the outside.

Yet, home IoT services with federated learning meet challenges in maximizing the user satisfaction: the response time of IoT devices controlled by a cloud server is slow; the accuracy of the model is not high because of insufficient data; threats to privacy are possible from a model inversion attack. The response time is one of the important factors to user experience. However, the response time of IoT services becomes slow when devices are controlled by the cloud server. In addition, insufficient data of the home network is the cause of lowering the accuracy of the training model. Model inversion attacks can extract training data from parameters between FL clients and a server.

In this paper, we propose a local IoT control method for federated learning home IoT service. We minimize the response time...
by employing the local control of IoT devices method. We also improve the accuracy of FL under insufficient data by applying transfer learning and mitigate private information leakage through model inversion attacks on the FL client with differential privacy.

From experiments, we show that local control of home IoT devices reduce the end-to-end response time by up to 18% when compared to centralized learning (CL). Our method provides the fast intelligent IoT service within 1 second. Furthermore, federated learning with transfer learning achieves the accuracy of 97% under about 9,000 samples, which is only 2% different from centralized learning. The accuracy of federated learning with differential privacy is 93%, which is 4% difference compared to the case without differential privacy.

## 2 RELATED WORK

### Response time of IoT service:

In [7], the authors used edge computing environments with SDN networks to reduce the response time of IoT applications. [2] presents different types of queuing models for QoS settings of IoT device interactions, showing that they have a significant impact on delivery success rate and response time. [5] proposes a service cache policy by utilizing the combinability of services to improve the performance of the service providing system. The author states that the average response time of IoT services can be improved as a result of conducting a series of experiments to evaluate the performance of the approach.

In our previous study, we measured and analyzed response times for IoT devices with and without cloud environments [10]. In [9], the author present context-aware IoT services in remote control. They use the trigger-action third-party IoT management service, IFTTT. However, the use of IFTTT cloud servers when controlling IoT devices often results in long response time.

**IoT service in federated learning:** [1] and [15] propose a personalized federated learning framework for protecting user privacy in the existing IoT service environment. Rodriguez-Barroso et al. [12] and Cao et al. [3] applied differential privacy to the existing federated learning framework for privacy protection.

Table 1 compares related work with our proposed method. We examine the bottleneck of the slow response time and improve the latency of federated learning IoT control. In addition, we support transfer learning and differential privacy together to improve the accuracy of FL. Previous studies applied federated learning to IoT, but they did not consider the response time.

### Table 1: Comparison between related work and our proposal.

|                  | IoT Control Service | Federated Learning | Transfer Learning | Differential Privacy |
|------------------|---------------------|--------------------|-------------------|----------------------|
| Aivodji et al. [1] | Remote              | ✓                  | ✗                 | ✓                    |
| Wu et al. [15]    | Remote              | ✓                  | ✗                 | ✓                    |
| Cao et al. [3]    | Remote              | ✓                  | ✗                 | ✓                    |
| Rodriguez et al. [12] | Remote         | ✓                  | ✗                 | ✓                    |
| Our approach      | Local               | ✓                  | ✓                 | ✓                    |

## 3 ACCELERATING FEDERATED LEARNING HOME IOT SERVICE

### 3.1 Home IoT Service in Federated Learning

Figure 1 is the overview of the home IoT service in federated learning. First, the FL client performs the local learning job using sensor data to detect user activities. For local training, the FL client communicates with the FL server. We combine the federated learning model with transfer learning to compensate for insufficient data. Additionally, we have enhanced privacy protection from model inversion attacks by applying differential privacy to our training model.

The local server (FL client + IoT controller) controls the home IoT device suited for the scenario corresponding to the classified activity. We apply transfer learning (TL) and differential privacy (DP) to the federated learning model in the local training process in FL client. As the model trained through federated learning resides on the local server, the FL client does not need to communicate with the server for the classification job. The IoT controller on the local server manages the IoT device according to the classified user activity. The IoT controller authenticates users and sends control messages directly to the IoT device for home services. In Table 2, we summarize user activities and the corresponding IoT service scenarios.

### 3.2 IoT Device Control: Local vs. Remote

Home IoT services typically require servers to perform complex tasks such as connecting IoT devices and generating control commands through machine learning models. In centralized learning, a cloud server trains a machine learning model for home IoT services, and the inference process is also performed on the server. IoT devices are remotely controlled through centralized learning. On the other hand, federated learning runs machine learning models on a local device, minimizing communication with remote servers. Therefore, local control of home IoT devices through federated learning reduces the communication process with the server to the minimum, enabling fast service within a short time.

Figure 2 shows how local control is combined with federated learning. We assume a home network consisting of sensors, a local server (FL client and IoT controller), and IoT devices. The FL client detects user activities through federated learning. The IoT controller authenticates the user allowed to control the device, and controls the device based on the classified activity. Our home IoT service, combined with federated learning and local control, can quickly improve the response time by performing all processes locally from data analysis to user authentication and control.

### Table 2: IoT service scenarios according to user activity.

| User activity | IoT service scenario                                  |
|--------------|------------------------------------------------------|
| Reading      | Turn on smart light                                  |
| Drinking     | Record user water intake in the local server database and notify with smart speaker |
| Using laptop | Block harmful URL at WiFi router                     |
| Using mobile phone | Manage traffic at WiFi router                      |
| Washing dishes | Play YouTube with smart speaker                   |

## ACKNOWLEDGMENTS

The authors would like to acknowledge the support of XYZ Foundation for their research. This work was also supported by the Government of the Republic of Korea through the National Research Foundation of Korea (NRF) grant number 2020R1F1A1015175.
Figure 1: The overview of home IoT service with local control in federated learning.

Figure 2: Local control of home IoT service with the federated learning model.

Figure 3: Remote control (IFTTT) of home IoT service with the centralized learning model.

3.3 Federated Learning with Transfer Learning and Differential Privacy

We combine federated learning with transfer learning and differential privacy to improve model performance against insufficient data and enhance privacy protection against model inversion attacks. Federated learning combined with TL and DP is shown in Fig. 4. Before starting training, a FL server in the cloud imports the transfer model, $T$. The FL server sets the initial value of $\epsilon$ and $\delta$, which are hyperparameters for differential privacy. FL client $i$ applies the global model parameter $W_g$ downloaded from a FL server and hyperparameter to model, $T$. In the next step, the FL client $i$ updates the parameters of the local model $\Delta W_i$ based on the data.

$^2$https://ifttt.com/
Figure 4: The flowchart of federated learning with TL and DP.

Figure 5: The testbed of local IoT control for federated learning home service.

$X_i$ and the model parameter $W_i$. Each FL client then uploads the updated local model parameter $\Delta W_i$ to the FL server. Finally, the FL server aggregates the parameters $\Delta W_i$ for all clients. The training process builds the model iteratively in each round. In federated learning, the transfer model, $T$, learns the characteristics of training data in advance, and it solves the insufficient data for each client. The model inversion attack can estimate the training data $X_i$ using the parameter $\Delta W_i$ of the model. We add noise to the parameters via the differential privacy parameters $\varepsilon$, $\delta$.

4 EXPERIMENTS

4.1 Experiment Environment

Figure 5 is the experiment environment of home IoT service with federated learning. We have implemented a home IoT service in federated learning with Tensorflow\(^3\) and OpenCV\(^4\). We configure a FL server with VM in Google Cloud. FL clients and a FL server communicate with WebSocket\(^5\). We connect the controller of IoT devices such as Philips Hue, TP-Link, and Google Nest Hub to a local server with a camera. The source code is available on GitHub\(^6\).

We summarize the components in our experiment as follows.

- Camera: webcam (Logitech C920).
- Local Server (FL client + IoT controller): laptop (Lenovo ThinkPad X1) in Ubuntu 20.04 LTS.
- IoT devices: smart light (Philips Hue), WiFi router (TP-Link), and a smart speaker (Google Nest Hub).

**Building a model:** For experiments, we use models of MobilenetV2 \(^8\), and EfficientnetB0 \(^13\). Both models are initially trained with an input image of size $224 \times 224 \times 3$. We describe the parameters required for model training in Table 3. To initially obtain a learning baseline, we train transfer learning model. In the $1_{st}$ round, the conv2D and dense layers are not updated during training, only the weights of the softmax layer that have been changed to match the new class are updated. In other words, all layers are set to be frozen except the last softmax layer. The initial learning rate is set to $10^{-3}$ and the model is trained for 10 epochs. After training the last softmax layer, we fine-tune the training model. We train the fine-tuned model by changing the epoch to 30 and the learning rate to $10^{-4}$. After the $1_{st}$ round, the model trains for 10 epochs. We limit the epochs to avoid overfitting because the overall amount of data is small and we reuse the model trained in the previous round.

**Datasets:** We collect 8,948 image frames through the camera for the five activity categories discussed earlier in Table 2. For the accuracy test, after recording a video file for three seconds in 10 frames per second, we label the corresponding action for each image. Data is divided into training and test dataset as shown in Table 4. We set up three participants for the experiment and collect five activity images. We have 6,920 frames of training data and 2,028 frames of test data.

4.2 Response Time

We investigate the end-to-end response time of home IoT services consisting of local control and federated learning steps. We compare local control with FL and remote control with CL as an IoT services. The response time is the time between capturing image and control an IoT device. Figure 6 shows the end-to-end IoT service response time consisting of capturing, transmitting images, and detecting

---

3https://www.tensorflow.org/
4https://opencv.org/
5https://websockets.readthedocs.io/en/stable/
6https://github.com/HwangDongJun/Federated_Learning_using_Websockets
user activities, user authentication, and issuing IoT action trigger commands.

Figure 7 shows the IoT service response time from $T_C$ to $T_R$. We compare FL and CL under local, remote, or IFTTT for smart light control. The service response time with FL and local IoT control is only 0.81 seconds. However, the response time increases to 3.67 seconds with FL in remote IoT control, and 4.27 seconds with CL in remote IoT control. Remote IoT control from CL using IFTTT has the response time of 4.61 seconds. In CL with remote IoT control, it takes 0.64 seconds to transfer image to the server, and 2.86 seconds for the cloud server to trigger an action to the IoT device, which is the bottleneck of the IoT control. In the case of IFTTT, it takes a long time (3.16 seconds) for the IoT control because the authentication and IoT control are performed through the IFTTT server and the remote IoT server.

Figure 8 compares how the response time varies with the number of clients in FL and CL. We measure the response time from $T_C$ to $T_M$ for activities classified by a machine learning model. The response time in FL is 0.4 seconds for 10 clients. On the other hand, in CL, we observe that it took 1.1 seconds to complete the classification and IoT control job. For 100 clients, it took 4.8 seconds with FL and 9.5 seconds with CL, resulting in the difference of 4.7 seconds. The response time of CL under many clients is slow because the overhead of large file transmission and training increases to waste the computation resources of a CL server.

Table 5 shows the response time for five user activities. In local control with FL, the response time is 0.81 seconds for the reading event; 3.09 seconds for the drinking water event; 1.58 seconds for events using laptop and mobile phone. On the other hand, in remote control with CL, the response time for the washing dish event that plays YouTube on Google Nest Hub is 14.05 seconds.

### 4.3 Accuracy and Privacy

In this section, we perform two experiments. First, we compare the accuracy of the FL models with and without transfer learning.
4.3.1 Federated Learning with Transfer Learning vs. Federated Learning without Transfer Learning. In Fig. 9, we observed that federated learning with transfer learning (FL with TL) outperforms federated learning (FL without TL). FL without TL starts with low accuracy due to insufficient data. The accuracy of the FL with TL is 73% higher than the FL without TL in the first round and 17% higher in the 10th round. Compared to the FL without TL, the FL with TL quickly achieves high performance.

4.3.2 Federated Learning with Differential Privacy. We train the model using a differentially-private stochastic gradient descent (DP-SGD) optimization algorithm. As shown in Table 6, we set the parameters to examine the performance of FL with TL. \( \epsilon \) is the privacy loss metric or privacy budget to measure the strength of privacy, and the probability of accidentally leaking information is \( \delta \), which limits the probability that privacy is not guaranteed. Moreover, we limit the exposure of personal information by setting the noise multiplier and the clipping threshold.

![Figure 9: Accuracy of federated learning with or without transfer learning.](image)

**Table 6:** Sample parameters regarding privacy constraints. num_microbatches is 32 (equal to batch_size), and \( \delta \) is \( 10^{-4} \) (\( \delta < 1/n \), n is the number of training examples).

| Noise multiplier | Clipping threshold | \( \epsilon \) |
|------------------|--------------------|---------------|
| 0.3              | 0.5                | 62.5          |
| 0.5              | 0.7                | 10.9          |
| 1.3              | 1.5                | 0.9           |
| -                | -                  | Non-private   |

using TensorFlow Hub\(^7\). Second, we investigate the impact of differential privacy on the accuracy and privacy of the FL model with TensorFlow Privacy\(^8\).

Figure 10 shows the accuracy of the FL with TL model (MobilenetV2) with different levels of protection \( (\epsilon = 0.9, \epsilon = 10.9, \text{ and } \epsilon = 62.5) \). In this experiment, we calculate the \( \epsilon \) value from the parameters in Table 6. Since \( \delta \) is set to be less than the inverse of the number of training data in privacy, we set \( \delta \) to \( 10^{-4} \) in our experiment. We can observe that as \( \epsilon \) decreases, the level of privacy protection becomes high due to noise. For MobilenetV2, the accuracy of the model with \( \epsilon = 0.9 \) in the final round is 93%, which is 2% different from the model with 10.9. In addition, we observe a slight difference of 4% compared to the model without DP. In the last round, the 95% accuracy of the model with \( \epsilon = 0.9 \) results in 3% difference compared to the 98% accuracy of the simple FL model without DP. This means that our FL with TL and DP model can classify user activities even if we set the highest privacy strength \( (\epsilon = 0.9) \) in our experiment.

**5 CONCLUSION AND FUTURE WORK**

In this paper, we present a local control method for federated learning home IoT services that minimize the end-to-end response time. The local control can minimize the end-to-end response time because there is no communication overhead with the cloud server. In the learning process, the FL client directly trains the individually collected data and sends the results to the federated server. We apply transfer learning to the federated learning model to improve the user context classification model accuracy due to insufficient data. We also evaluate federated learning methods using differential privacy applied to provide improved privacy protection against model inversion attacks.

In future work, we plan to extend the IoT service of federated learning to various IoT devices and user activities. We need a way to train models with scalable user activity for IoT devices. This requires experimentation with real users’ activities so that they can be generalized to federated learning environments. We believe that a crowd-sourcing test that uploads an image of an activity by a user should also be developed as a method. In addition, we consider the use of personal information in public places that value

\[^7\]https://www.tensorflow.org/hub

\[^8\]https://github.com/tensorflow/privacy
personal information, such as rest rooms and toilets, rather than in an environment where IoT devices are individually controlled.

ACKNOWLEDGMENTS

This work was supported by Institute for Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government (MSIT)(No.2019-0-01343, Training Key Talents in Industrial Convergence Security) and this work was supported by Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No.2020-0-00901, Information tracking technology related with cyber crime activity including illegal virtual asset transactions). Corresponding author is Youngseok Lee.

REFERENCES

[1] Ulrich Matchi Aïvodji, Sébastien Gambs, and Alexandre Martin. 2019. IOTFLA: A secured and privacy-preserving smart home architecture implementing federated learning. In 2019 IEEE Security and Privacy Workshops (SPW). IEEE, 175–180.

[2] Georgios Bouloukakis, Ioannis Moscholios, Nikolaos Georgantas, and Valérie Issarny. 2021. Performance Analysis of Internet of Things Interactions via Simulation-Based Queueing Models. Future Internet 13, 4 (2021), 87.

[3] Hui Cao, Shubo Liu, Renfang Zhao, and Xingxing Xiong. 2020. Ifed: A novel federated learning framework for local differential privacy in Power Internet of Things. International Journal of Distributed Sensor Networks 16, 5 (2020), 15502720919698.

[4] Rebecca Deczynski. 2021. Thanks to This Color-Changing Light Bulb, I Can Finally Fall Asleep Faster at Night. (2021). https://www.yahoo.com/lifestyle/thanks-color-changing-light-bulb-201529163.html

[5] Shuiguang Deng, Zhengze Xiang, Jianwei Yin, Javid Taheri, and Albert Y. Zomaya. 2018. Composition-Driven IoT Service Provisioning in Distributed Edges. IEEE Access 6 (2018), 54258–54269. https://doi.org/10.1109/ACCESS.2018.2871475

[6] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augustein, Hubert Eichner, Chloé Keddon, and Daniel Ramage. 2018. Federated learning for mobile keyboard prediction. arXiv preprint arXiv:1811.03604 (2018).

[7] Juan Luis Herrera, Jaime Galán-Jiménez, Javier Berrocal, and Juan Manuel Murillo. 2021. Optimizing the Response Time in SDN-Fog Environments for Time-Strict IoT Applications. IEEE Internet of Things Journal (2021), 1–1. https://doi.org/10.1109/JIOT.2021.3077992

[8] Andrew Howard, Andrey Zhmoginov, Liang-Chieh Chen, Mark Sandler, and Menglong Zhu. 2018. Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. (2018).

[9] SV Aswin Kumer, P Kanakaraja, A Punya Teja, T Harini Sree, and T Tejaswini. 2021. Smart home automation using IFTTT and google assistant. Materials Today: Proceedings (2021).

[10] Hyungjin Lee, Hyunsu Mun, and Youngseok Lee. 2020. Comparing Response Time of Home IoT Devices with or without Cloud. In 2020 IEEE International Conference on Consumer Electronics (ICCE). IEEE, 1–6.

[11] Molly Price. 2021. The new Google Nest Hub tracks your sleep without wearables or cameras. (2021). https://www.cnet.com/home/smart-home/the-new-google-nest-hub-tracks-your-sleep-without-wearables-or-cameras/

[12] Nuria Rodríguez-Barroso, Goran Stipcich, Daniel Jiménez-López, José Antonio Ruiz-Millán, Eugenio Martínez-Cámara, Gerardo González-Scoto, M Victoria Luzón, Miguel Ángel Vézanzones, and Francisco Herrera. 2020. Federated Learning and Differential Privacy: Software tools analysis, the Sherpa ai FL framework and methodological guidelines for preserving data privacy. Information Fusion 64 (2020), 270–292.

[13] Mingxing Tan and Quoc V Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. arXiv preprint arXiv:1905.11946 (2019).

[14] Paul Voigt and Axel Von dem Bussche. 2017. The eu general data protection regulation (gdpr). A Practical Guide, 1st Ed., Cham: Springer International Publishing 10 (2017), 3152676.

[15] Qiong Wu, Kuiwen He, and Xu Chen. 2020. Personalized federated learning for intelligent IoT applications: A cloud-edge based framework. IEEE Open Journal of the Computer Society 1 (2020), 35–44.