Privacy Preserving Federated RSRP Estimation for Future Mobile Networks

Omer Haliloglu
Ericsson Research
Istanbul, Turkey
omer.haliloglu@ericsson.com

Elif Ustundag Soykan
Ericsson Research
Istanbul, Turkey
elif.ustundag.soykan@ericsson.com

Abdulrahman Alabbasi
Ericsson Research
Stockholm, Sweden
abdulrahman.alabbasi@ericsson.com

***Abstract*— Leveraging location information for machine learning applications in mobile networks is challenging due to the distributed nature of the data and privacy concerns. Federated Learning (FL) helps to tackle these issues and is a big step towards enabling privacy-aware distributed model training; however still prone to sophisticated privacy attacks such as membership inference. In this paper, we implement an FL approach to estimate Reference Signal Received Power (RSRP) values using geographical location information of the user equipment. We propose a privacy-preserving mechanism using differential privacy to protect against privacy attacks and demonstrate the impacts and the privacy-utility trade-off via privacy accounting measures.**

Keywords*— *Federated learning, differential privacy, network optimization, RSRP estimation, 5G, 6G.

I. INTRODUCTION

With the advent of beyond 5G and 6G, next-generation wireless systems are expected to connect everything with enhanced coverage, capacity, energy efficiency, latency for different application-oriented use-cases. Such challenging requirements could be achieved via utilizing advance wireless techniques or intelligent data-driven methodologies.

Existing approaches like network densification, optimal resource allocation, and massive and distributed Multi-Input Multi-Output (MIMO) systems help reaching these requirements. However, for instance, network densification requires frequent carrier and cell measurements to enable interference mitigation and coordination schemes due to high interference resulting from densifying the network. Such intense measurements should be performed by the User Equipment (UE) on the primary, secondary, and any potential target cells on different frequency bands and spatially distributed network sites. Although UE measurement reports are important to guarantee service continuity, frequent Reference Signal Received Power (RSRP) measurements, e.g., in every 40 ms, produce large signaling overhead and increase power consumption.

The application of data-driven intelligent techniques on wireless systems is considered as a potential solution to the problem of high overhead, energy consumption, interference mitigation, resource allocation. Integrating the intelligence to monitor and predict the network status enables network automation and improves UE experience by learning from history and promoting proactive real-time network decisions [1]. Machine Learning (ML) algorithms can be utilized for RSRP prediction, which reduces the signaling overhead, enables proactive network actions, and increases computation efficiency. RSRP prediction with relevant features, e.g., UE contextual information, further enhances beam management and mobility robustness.

Despite the increasing popularity of ML, centralized processing of a big volume of data increases both computation complexity at network sites and privacy concerns for many UEs against data leakage. However, local datasets are valuable assets to increase learning accuracy. In order to benefit from a large dataset that captures independent and heterogeneous scenarios without transferring it, while avoiding centralized exhausting computation, distributed learning mechanisms drew considerable attention recently. Federated learning (FL) [2], [3], [4] is one of such distributed learning schemes addressing privacy concerns by performing coordinated learning on clients without revealing their local dataset to a central entity.

FL provides a privacy-aware topology since the client's data is never shared with the server instead, only model updates are exchanged, which do not contain any personally identifiable information, and the size of exchanged updates is less than the raw data. Although these are considered significant privacy enhancements compared to the centralized training scenarios, where clients need to send raw data to the server, it is still possible to launch privacy attacks. Recent advances show that sophisticated privacy attacks such as membership inference, model inference, and model extraction attacks can still exploit the model updates [5]. Differential Privacy (DP), which is a privacy-enhancing technology, is used to prevent such attacks.

In this paper, the geographical location of UEs is utilized as a feature to predict RSRP. Since location sharing with network sites is not preferable due to privacy concerns, differentially private FL is employed to realize the privacy-preserving prediction. We defined each capable UE as a client; thus, each UE can help the system by mapping its location to RSRP via updating model parameters. Targeting not only 5G networks but also beyond 5G and 6G networks, this study:

- Utilizes UEs' location information in the context of FL to predict RSRP (obtained from a realistic testbed experiment) to enhance beam management and mobility robustness.
- Brings a privacy-preserving approach with DP against possible privacy attacks on the above proposed framework.
- Provides comprehensive experiment results including, but not limited to, privacy versus utility trade-off and performance metrics.

The organization of this paper is described below. In Section 2, we present a background on FL, its privacy concerns, and DP. We explain the proposed solution for RSRP prediction and how it can be performed using differentially private FL in Section 3. Details about our implementation and evaluations are discussed in Section 4. We discuss state of the art in Section 5, and we conclude the study in Section 6.

II. BACKGROUND

A. Federated Learning

FL was proposed by [3] and is one kind of distributed learning utilized for the minimization of data transfer, joint optimization, and privacy purposes, and its performance depends on the variety of implementations. Depending on the distribution of samples and features among the clients, training can be done via horizontal or vertical splits over samples or features. Horizontal FL is applicable when the clients share the same feature space but differ in sample space, whereas vertical FL is appropriate when the clients have overlaps in the sample space. When clients have small overlaps in both sample and feature space, federated transfer learning is more suitable.

From an optimization point of view, Federated Stochastic Gradient Descent (FedSGD) and Federated Averaging (FedAvg) are the most used methods. In FedSGD, each client sends every Stochastic Gradient Descent (SGD) update to the server, while in FedAvg, each client performs some number of iterations (called epoch) over a local mini-batch then sends the updated model to the server. In our study, we follow horizontal FL case and FedAvg as it is more communication efficient. Hence, in each iteration, the clients (UEs) do training on their local batch, send the resulted model back to the server located in a centralized network site (e.g., hereafter we call the centralized network site as gNB, but this work is not limited to 5G, as it could be adopted in 6G network), via uplinks signals. It follows that the server aggregates models by averaging the corresponding weights, then shares the aggregated model with the clients (UEs) in the downlink.

Due to the nature of its operation, FL exhibits considerable advantages for the mobile network, described as follows:

- Exchange learnings among clients (UEs): The individual learning of each UE is shared among each other when building the global model. This sharing of learning could be global sharing among all UEs (e.g., when no clustering is employed before aggregating), or it could be a cluster-based sharing of learning (e.g., when clustering for a specific type of service or UE, is deployed before aggregating [6]).
- Privacy: Since training is performed locally, FL preserves UE information privacy. The original operation of FL does not require the UE to send its private information (e.g., UE location) to gNB, or reveal its identity. Instead, it learns from aggregated information produced by multiple UEs.
- Efficient network footprint by reducing the amount of signaling required at gNB by UEs, due to the need of sending on the gradient of the model's weights.

B. Privacy Concerns

FL is a big step towards enabling privacy-aware model training, as it aims joint model training by only sharing the necessary parameters and not client data. On the other hand, recent studies demonstrate sophisticated privacy attacks by exploiting the observed gradients or using the collected inference results. Privacy attacks to FL such as membership inference [7], attribute property inference [8], deep leakage [9] , and model extraction [10] may be posed by a malicious client or the server trying to infer sensitive information during training or inference phase. The adversarial goal of privacy attacks is to gain more information about the training data and the ML model parameters. The attacker may try to determine whether a specific client's data was included in the training dataset, called membership inference, or try to infer certain properties of other clients' training data, which is not explicitly shared, called property inference. In model extraction attacks, attacker uses a prediction interface as an oracle to obtain the structure of the model by inspecting the probabilities returned from each class. Deep leakage attacks attempt to infer both training input and labels, which is more dangerous since the raw training data can be extracted [9].

From the solution perspective, existing methodologies including DP, Homomorphic Encryption (HE), and Secure Multi-Party Computation (SMPC) are investigated in the literature. However, for large-scale FL scenarios and cross-device settings, HE and SMPC may not be the best options as they introduce additional computation and communication overhead. DP satisfies the constraints at the cost of reducing accuracy which is controllable based on the privacy budget.

C. Differential Privacy

DP provides a mathematically provable guarantee of privacy protection based on the available privacy budget, which is the limit on the amount of difference that an individual's participation can generate. Hence, DP prevents distinguishing whether or not the individual's data is included in the training.

Definition: A randomized function M is ε-differentially private if for any subset of the output S in the range of M, and for all data sets D_1 and D_2 differing in a single entry [11]:

$$Prob\,[M(D_1) \in S] \leq exp\,(\varepsilon)\,Prob\,[M(D_2) \in S] + \delta \qquad (1)$$

The given formulation is called (ε, δ)-differential privacy where δ is the relaxation parameter. ε is the control parameter for privacy level, denoting privacy budget. δ limits the probability that the privacy guarantee will not hold; in other words, the probability of information accidentally being leaked. The best practice is to set δ to be less than the inverse of the data size. In the context of FL, D_1 and D_2 datasets correspond to client -in our case UE- training datasets. Adjacent datasets are those where D_2 can be formed from D_1 by adding or removing all the training samples associated with a single client. This approach is called user-level privacy [12].

There are two different DP implementations in the FL setting, called Central DP and Local DP. The difference comes from the trust relation. In Central DP, noise is added in and controlled by the server which aggregates the updates, so clients trust the server. Local DP allows the client to control and add noise for cases the server is not trusted by the clients. Although Local DP allows the individual client to set different local privacy guarantees and removes the trusted server assumption, it reduces the model accuracy. In our case, we use Central DP where the noise is controlled by the gNB.

Privacy accounting is essential in DP as it is the control mechanism for the privacy versus accuracy trade-off. The mechanism to control and track privacy uses the moments accountant method [13]. It assures that the defined privacy budget given with the via (ε, δ) parameters stays within the allowed limit. In FL, privacy accounting for multiple iterations can be done using the composability feature of DP to compute and accumulate the privacy cost at each round of training.

III. RSRP PREDICTION

A. System Model

Signal quality (either layer-1 Channel State Information-Reference Signal (CSI-RS) or layer-3 "RSRP") prediction is an important feature that can be obtained with the introduction of intelligent and proactive management of the network resources. RSRP measurements is obtained, in the legacy system, via measuring the reference signal in CSI-RS, then it is reported back to the network in a specific configuration. Instead of reporting legacy RSRP, reporting predicted RSRP to the network could be used to proactively 1) perform handover (HO) or allow UE to perform conditional handover, 2) switch beams or carrier for a UE to prevent service interruption. For instance, the UE must report predicted RSRP when the network needs to decide, whereas it could report its own predicted decision if the scenario is like that of conditional HO.

In order for the UE to predict such RSRP, it requires input and output labels to train the UE model (as illustrated better in a later section). The input of the learning model is the UE location. And not any other feature (e.g., time advance (TA)), because running feature importance proves that location is one of the most critical features, plus other features like TA would introduce signaling overhead. The training label is obtained from CSI-RS reference signal.

B. RSRP Prediction via FL

An ML model is created at the gNB and shared with the clients (UEs), in which each UE trains its individual model by mapping geographical location features to RSRP measurements of the serving carrier/cell. Let the data set throughout the network be defined as

$$D_i^k = \{(x_i^k, y_i^k), i = 1, \dots, N, k = 1, \dots, K\} \quad (2)$$

where $x_i \in \mathbb{R}^d$ represents the feature space and d is the number of features, y_i denotes the output label per sample i, k is the UE index out of K UEs, and i is the dataset sample. The training algorithm $A: D_i^k \rightarrow \hat{f}^{k,i} \in F$ finds the estimated function, $\hat{f}^{k,i}$, for the unknown function, $f^{k,i}$, that maps the UE location information to RSRP information. x_i contains only geographical location information of the UEs, i.e., latitude and longitude, and y_i consists of the maximum measured RSRP values out of all beams. Then the prediction problem is formulated to establish a relation between the geographical location and the highest RSRP values. The features contain sensitive information, and learning has been performed in a federated framework over the UEs. The global model minimizing the objective function given as,

$$\min_{\omega \in \mathbb{R}^d} \quad f(\omega) \quad (3)$$

where global objective function is defined as average of local objective functions of each UE such that

$$f(\omega) = \frac{1}{K}\sum_{k=1}^{K} f^k(\omega) \quad (4)$$

Herein, local objective functions are defined via global loss function and evaluated at local datasets $\{x^k, y^k\}$ of k-th UE such that $f^k(\omega) = l(\omega; x^k, y^k)$.

The procedure is given in Algorithm 1 and summarized as follows: Global model is created at the gNB and shared with the UEs who will participate in training. Each UE trains its individual model by mapping its location data to RSRP measurements. After local training, each UE sends its trained model parameters to the gNB. gNB aggregates the parameters using FedAvg and sends back the updated model parameters to the UEs, which contains an implicit mapping of all UEs to RSRP, embedded by local training.

Algorithm 1. Pseudo-code for federated learning procedure

Initialize the global learning model (θ) at gNB.
gNB shares θ, ω^0 with the UEs participating the training.
for each round $t = 0,1, \dots, T$ *do*
  for each user k
    $\Delta\omega_k^{t+1} \leftarrow$ Local training (D, θ, ω^t)
  $\omega^{t+1} \leftarrow \omega^t +$ FedAvg$(\sum \Delta\omega_k^{t+1})$
  gNB shares ω^{t+1} updated parameter with UEs.

C. RSRP Prediction via Differentially Private FL

Differential privacy mechanism perturbs the averaged updates conducted at the gNB using a randomized Gaussian mechanism. The purpose of the randomization process is to hide each UE's contribution within the federated aggregation and thus within the learning procedure.

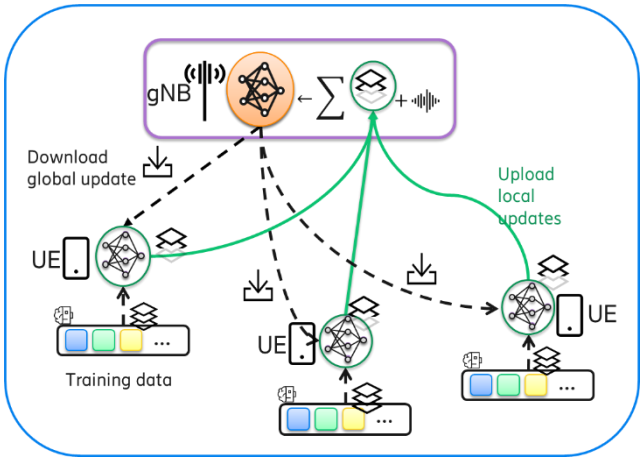

Fig. 1. Differentially private federated learning architecture

Algorithm 2. Pseudo-code for differentially private federated RSRP estimation.

Initialize global learning model (θ) at gNB, noise multiplier (z), clip scale (c), target delta (δ)
gNB shares θ, ω^0 with the UEs
for each round $t = 0,1, \dots, T$ *do*
  for each UE k
    $\Delta\omega_k^{t+1} \leftarrow$ Local training $(D, \theta\ \omega^t)$
    $\Delta\omega_k^{t+1} \leftarrow$ Clip $(\Delta\omega_k^{t+1}, c)$
  $\sigma \leftarrow zc$
  $\Delta\omega^{t+1} \leftarrow$ FedAvg $(\sum \Delta\omega_k^{t+1})$
  $\omega^{t+1} \leftarrow \omega^t + \Delta\omega^{t+1} + \mathcal{N}(0, \sigma^2)$
  get_privacy_spent*(z, q, δ)*
  If δ is achieved, *then* exit
  gNB shares ω^{t+1} parameter updates with UEs.

Clip$(\Delta\omega_k^{t+1}, c)$:
  return $\Delta\omega_k^{t+1} * c/max(\|\Delta\omega_k^{t+1}\|, c)$

The procedure is depicted in Fig. 1 and Algorithm 2, and summarized as below:

- Global model is created at the gNB and shared with the UEs who will participate in the training.
- Each UE trains its individual model. After local training, each UE sends its trained model parameters to the gNB. Since the contribution of each UE update can be different, the gradients will be clipped after each local update iteration with the clip-scale (c) parameter.
- gNB aggregates the parameters via FedAvg, and perturbs the model by adding random Gaussian noise, controlled by noise multiplier (z) and clip-scale.
- gNB sends back the aggregated parameters to the UEs, which contains an implicit mapping of all clients to RSRP, embedded by local training.
- Using get_privacy_spent method, privacy accounting is accomplished, and epsilon (ε) is calculated.

Training will be continued until the target delta (δ) reaches a certain threshold to limit the probability of a client's contribution accidentally being leaked.

D. Threat Model

In our study, we consider UEs not being malicious during training activities. UEs obey the FL steps, do not try to inject malformed data to poison the model or alter the protocol. gNB, as the FL server, is considered as trusted data aggregator, i.e., it follows the FL steps and does not attempt to share parameters with anyone. Our framework protects the model against untrusted analysts who can send queries using an inference interface and try to infer sensitive information by collecting model inference results.

IV. Experiments and Results

A. Simulation Environment

We realized our implementation in the Tensorflow environment. Keras, which is a high-level ML API used to define a Neural Network (NN) in the Tensorflow. Federated computations on decentralized data are performed with Tensorflow Federated (TFF) framework. The corresponding NN is created by using Keras comprised of an input, an output and 3 hidden layers including 10 neurons each. To measure the training loss, Mean Squared Error (MSE) is used.

We integrated DP using Tensorflow Privacy library which enables us to analyze the performance of our implementation by turning on/off DP. Tensorflow privacy library gives the opportunity to set different privacy levels by adjusting privacy hyper-parameters such as noise multiplier, gradient clip-scale and δ. In our setup, we used a fixed clip-scale for all updates. In performance evaluation subsection C, we provide the impact of these parameters. The experimental setup of our implementation is described in Table 1.

TABLE I. Table Type Styles

Parameters	Setup/Value
Model	Keras Sequential
Environment	Tensorflow v2.4.1 TFF v0.18.0 Tensorflow Privacy v0.5.1 Python v3.7.10
Batch size (B)	100/1000
Noise multiplier (z)	0.01/0.05/0.1/0.5/1/5
Data size	3.8e6
Client Learning rate	0.01
Number of Clients	38
Epochs	5
Target Delta (δ)	1e-7
Clip-Scale (c)	[0.1,1]
Train-Test data ratio	0.8:0.2
Workstation	8 Core Intel Xeon E3 with 16GiB DIMM RAM

B. Dataset

In a real-life environment, we ran a measurement campaign with a realistic base-station and UEs, which was conducted in an urban environment in Stockholm, Sweden. Fig. 2 illustrates the longitude and latitude values, X(m) and Y(m), with respect to a reference coordinate, and the color map indicates the RSRP values in dB scale, obtained at 3.5 GHz carrier on the evaluation area. The collected RSRP measurements over multiple days and locations are gathered in a single gNB node. Dataset generated over a specific region or during a specific period, is assigned a client tag, i.e., k-th client, and is denoted by D_i^k as in eq. (5),

$$D_i^k = \{(x_i, y_i), i \in [(k-1)Q + 1, kQ]\}, \forall i, k \qquad (5)$$

where i is the sample id, Q is the number of sequential samples per region or period of D^k and can vary for each UE.

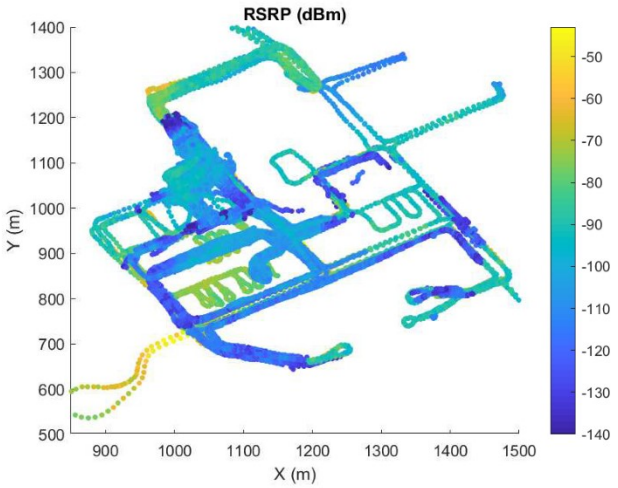

Fig. 2. RSRP values obtained at 3.5 GHz in the region.

C. Performance Evaluation

In this section, Mean Absolute Percentage Error (MAPE) and epsilon (ε) have been evaluated to show the impact of c, B, and z during the FL training.

In Fig. 3, it is demonstrated how different c values impact the training evaluation loss in terms of MAPE over FL rounds when noise multiplier, z=1 and batch size, B=100. Less gradient clipping, e.g., c={0.7, 1} results in faster convergence but a higher error with oscillations because not only the noise variance increases but also the gradients' sensitivity to the noise will not be the same in different clients. Nonetheless, if c decreases, the convergence becomes more stable. Therefore, the clip scale is chosen based on stability and convergence rate. Further, the privacy spent from the privacy budget increases during the training as the number of rounds increases, i.e., as MAPE decreases, ε, shown as privacy spent in Fig. 3, increases meaning that privacy decreases. Higher clip-scale values converge faster, i.e., require fewer rounds, thus results in better privacy (lower

ε), but clip-scale should have an upper bound for sensitivity aspects. Given the setting in Table 1, for c=0.1, the processing time cost of running the experiment is around 25 minutes.

Fig. 4 shows that *B* affects the clip-scale adjustment. When *c* is low (more clipping), bigger *B* can be used for faster convergence, thus less privacy will be spent from the privacy budget. When *c* is high, e.g., 0.7, bigger *B* impacts the norm

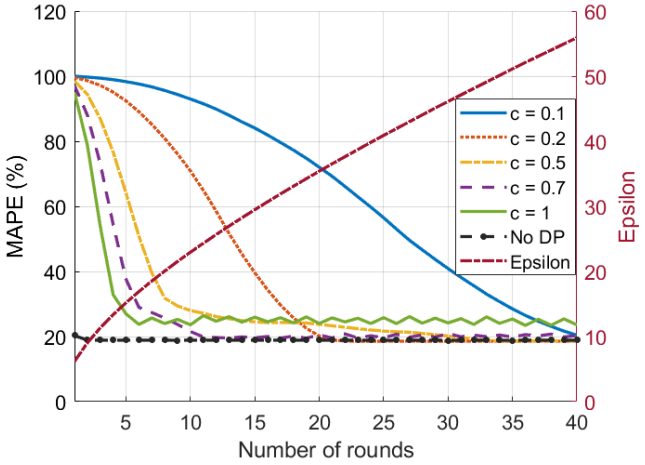

Fig. 3. Training evaluation loss for different *c* values when *z*=1 and *B*=100.

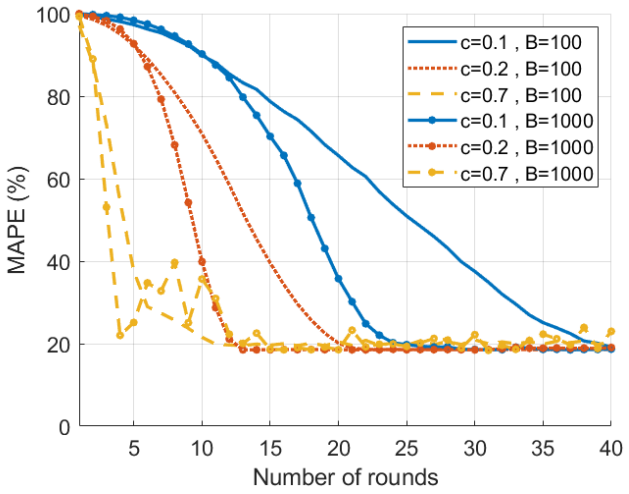

Fig. 4. Training evaluation loss for different *c* and *B* values when *z*=1.

of gradient vector, so the sensitivity to the noise, and MAPE eventually oscillates.

Fig. 5 shows that MAPE increases with increasing noise multiplier. How much accuracy is lost by turning on the DP can be seen by comparing with No DP results given in black dotted line. This shows the trade-off between utility and privacy. To get more privacy, one needs to sacrifice accuracy. If noise multiplier is set as too high, then the training will not converge as in the settings for *z*≥10.

Fig. 6 demonstrates that more privacy is spent as FL training continues with more rounds, and higher *z* values enhance privacy. However, one needs to consider the convergence while increasing the noise multiplier.

V. Related Work

ML technologies are used to predict different 5G metrics. For instance, the authors of [14] proposed a procedure to predict the strongest (highest RSRP) secondary serving cell. In their proposal, gNB does not require further signaling from UE but only uses existing 3GPP signals, i.e., timing advance and primary cell RSRP. Another work related to downloading auto-encoder ML model to enable UE to estimate radio measurements and reduce the signals transmitted over to the network [15]. This auto-encoder is already trained on the network side and downloaded to UEs. Also, the authors of [16] proposed a duo threshold-based classification to enhance reference signal received quality prediction.

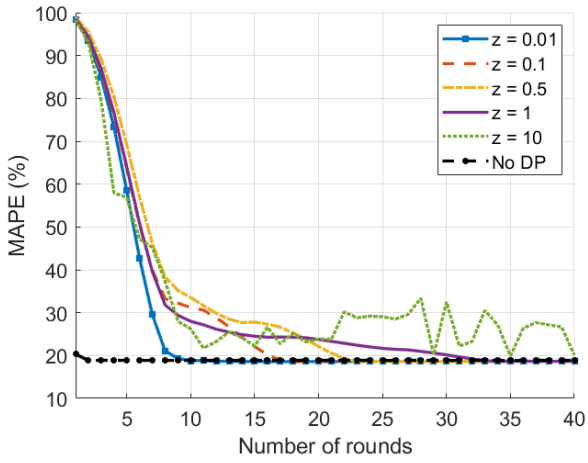

Fig. 5. Training evaluation loss for different noise multiplier values when *B*=100 and *c*=0.5.

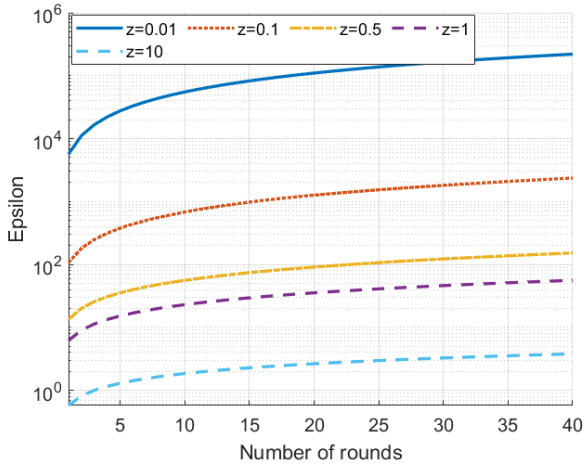

Fig. 6. Training evaluation privacy spent for different noise multiplier values when *B*=100 and *c*=0.5.

FL has been investigated intensively for 5G-beyond applications. A comprehensive review of the application of FL over 6G network is described in [17] and [18]. The authors of [19] proposed an FL scheme to predict the mmWave beamforming vector. In the proposed scheme, the users train the beamforming vector predictor network and share it with the network without sharing their data. Another interesting application of FL to 5G-beyond network is the usage of Gaussian process regression to track Channel State Information (CSI) by UEs [20]. Authors of [21] proposed digital and compressed analog distributed SGD to enable opportunistic scheduling of UEs based on channel conditions.

Preserving privacy and security of FL schemes is of main interest in the 5G and beyond networks. The authors of [22] designed a blockchain-based FL framework to achieve secure and reliable FL considered DP against inference attacks. In vehicular networks, the author of [23] introduced DP into the

gradient descent training scheme. In a 5G social Internet-of-Things context, authors of [24] proposed a hybrid of data and content privacy-preserving scheme incorporating Bayesian DP and a new encryption method.

Our work differs from existing literature in multiple aspects. Compared to those works that addressed RSRP prediction, we proposed to use only UE location in a federated learning context. Compared to the existing works that utilized FL to predict CSI for 5G or 6G applications, we proposed the usages of differential privacy in addition to FL to protect the model against inference attacks. We also consider our work to be the first of its kind to use realistic dataset in FL context with privacy-preserving technique.

VI. Conclusion

This paper demonstrates a privacy-preserving federated approach for an RSRP prediction framework and focuses on two important aspects: (i) the geographical location of UEs as a feature in an FL framework to predict RSRP (ii) bringing privacy guarantee to FL framework against inference attacks.

The FL training is performed locally on the UEs, using the location information. The local dataset, e.g., UE's location and targeted RSRP measure, are acquired from a real-life environment with realistic base-station and UE over multiple days and different areas. The local updates are aggregated in the gNB without accessing any location information. Our evaluation results showed that our model could successfully predict RSRP values with a 19% loss in terms of MAPE. We enabled DP during the training phase to prevent privacy attacks on the resulting model and presented the impact of DP parameters by turning the DP on/off.

RSRP estimation via differentially private FL not only enables location-aware communications and enhances the robustness of the beam management and mobility but also preserves the privacy of the UEs.

Acknowledgment

This work was supported by The Scientific and Technological Research Council of Turkey (TUBITAK) through the 1515 Frontier Research and Development Laboratories Support Program under Project 5169902 and has been partly funded by the European Commission through the H2020 project Hexa-X (Grant Agreement no. 101015956).

References

[1] "AI & machine learning: Next gen system|Whitepaper - Ericsson." https://www.ericsson.com/en/reports-and-papers/white-papers/machine-intelligence (accessed Jun. 08, 2021).

[2] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated Learning: Strategies for Improving Communication Efficiency," *arXiv:1610.05492 [cs]*, Oct. 2017, Accessed: Jun. 08, 2021. [Online]. Available: http://arxiv.org/abs/1610.05492

[3] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," *arXiv:1602.05629 [cs]*, Feb. 2017, Accessed: Jun. 08, 2021. [Online]. Available: http://arxiv.org/abs/1602.05629

[4] J. Konečný, B. McMahan, and D. Ramage, "Federated Optimization:Distributed Optimization Beyond the Datacenter," *arXiv:1511.03575 [cs, math]*, Nov. 2015, Accessed: Jun. 08, 2021. [Online]. Available: http://arxiv.org/abs/1511.03575

[5] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning," in *2019 IEEE Symposium on Security and Privacy (SP)*, May 2019, pp. 739–753. doi: 10.1109/SP.2019.00065.

[6] Y. Kim, E. A. Hakim, J. Haraldson, H. Eriksson, J. M. B. da Silva Jr., and C. Fischione, "Dynamic Clustering in Federated Learning," *arXiv:2012.03788 [cs]*, Dec. 2020, Accessed: Jun. 09, 2021. [Online]. Available: http://arxiv.org/abs/2012.03788

[7] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership Inference Attacks against Machine Learning Models," *arXiv:1610.05820 [cs, stat]*, Mar. 2017, Accessed: Jun. 27, 2021. [Online]. Available: http://arxiv.org/abs/1610.05820

[8] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting Unintended Feature Leakage in Collaborative Learning," *arXiv:1805.04049 [cs]*, Nov. 2018, Accessed: Jun. 29, 2021. [Online]. Available: http://arxiv.org/abs/1805.04049

[9] L. Zhu, Z. Liu, and S. Han, "Deep Leakage from Gradients," *arXiv:1906.08935 [cs, stat]*, Dec. 2019, Accessed: Jun. 29, 2021. [Online]. Available: http://arxiv.org/abs/1906.08935

[10] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing Machine Learning Models via Prediction APIs," *arXiv:1609.02943 [cs, stat]*, Oct. 2016, Accessed: Jun. 29, 2021. [Online]. Available: http://arxiv.org/abs/1609.02943

[11] C. Dwork and A. Roth, "The Algorithmic Foundations of Differential Privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3–4, pp. 211–407, Aug. 2014, doi: 10.1561/0400000042.

[12] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning Differentially Private Recurrent Language Models," *arXiv:1710.06963 [cs]*, Feb. 2018, Accessed: Jun. 24, 2021. [Online]. Available: http://arxiv.org/abs/1710.06963

[13] M. Abadi *et al.*, "Deep Learning with Differential Privacy," *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 308–318, Oct. 2016, doi: 10.1145/2976749.2978318.

[14] H. Ryden, J. Berglund, M. Isaksson, R. Cöster, and F. Gunnarsson, "Predicting strongest cell on secondary carrier using primary carrier data," in *2018 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, Apr. 2018, pp. 137–142. doi: 10.1109/WCNCW.2018.8369000.

[15] H. Rydén and R. Moosavi, "Downloadable machine learning for compressed radiolocation applications in radio access networks," in *2020 IEEE Globecom Workshops (GC Wkshps*, Dec. 2020, pp. 1–6. doi: 10.1109/GCWkshps50303.2020.9367519.

[16] C. Svahn, O. Sysoev, M. Cirkic, F. Gunnarsson, and J. Berglund, "Inter-Frequency Radio Signal Quality Prediction for Handover, Evaluated in 3GPP LTE," in *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, Apr. 2019, pp. 1–5. doi: 10.1109/VTCSpring.2019.8746369.

[17] S. Niknam, H. S. Dhillon, and J. H. Reed, "Federated Learning for Wireless Communications: Motivation, Opportunities, and Challenges," *IEEE Communications Magazine*, vol. 58, no. 6, pp. 46–51, Jun. 2020, doi: 10.1109/MCOM.001.1900461.

[18] Y. Liu, X. Yuan, Z. Xiong, J. Kang, X. Wang, and D. Niyato, "Federated learning for 6G communications: Challenges, methods, and future directions," *China Communications*, vol. 17, no. 9, pp. 105–118, Sep. 2020, doi: 10.23919/JCC.2020.09.009.

[19] I. Chafaa, R. Negrel, E. V. Belmega, and M. Debbah, "Federated Channel-Beam Mapping: from sub-6GHz to mmWave," in *2021 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, Mar. 2021, pp. 1–6. doi: 10.1109/WCNCW49093.2021.9420006.

[20] M. M. Wadu, S. Samarakoon, and M. Bennis, "Federated Learning under Channel Uncertainty: Joint Client Scheduling and Resource Allocation," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, May 2020, pp. 1–6. doi: 10.1109/WCNC45663.2020.9120649.

[21] M. M. Amiri and D. Gündüz, "Federated Learning Over Wireless Fading Channels," *IEEE Transactions on Wireless Communications*, vol. 19, no. 5, Art. no. 5, May 2020, doi: 10.1109/TWC.2020.2974748.

[22] Y. Liu, J. Peng, J. Kang, A. M. Iliyasu, D. Niyato, and A. A. A. El-Latif, "A Secure Federated Learning Framework for 5G Networks," *IEEE Wireless Communications*, vol. 27, no. 4, pp. 24–31, Aug. 2020,

[23] 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, Mar. 2020.

[24] L. Yin, J. Feng, H. Xun, Z. Sun, and X. Cheng, "A Privacy-Preserving Federated Learning for Multiparty Data Sharing in Social IoTs," *IEEE Transactions on Network Science and Engineering*, pp. 1–1, 2021.