Towards an Efficient, Privacy-aware Federated Learning Scheme

Levente Alekszejenkó (alelevente@mit.bme.hu)
Budapest University of Technology and Economics: Budapesti Muszaki es Gazdasagtudomanyi Egyetem
https://orcid.org/0000-0002-3196-1950

Tadeusz Dobrowiecki
Budapest University of Technology and Economics: Budapesti Muszaki es Gazdasagtudomanyi Egyetem

Research

Keywords: privacy-preserving, federated learning, raw data sharing, real-time decision-making

Posted Date: October 11th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-2072660/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License
Towards an Efficient, Privacy-aware Federated Learning Scheme
Levente Alekszejenkó* and Tadeusz Dobrowiecki

Abstract
With the rapid evolution of Internet-of-Things (IoT) devices and the emergence of Autonomous Vehicles (AVs), machine learning processes pose a growing privacy issue. Federated learning (FL) and current cryptography can mitigate this problem; however, these solutions might not be efficient enough during the decades-long lifespans of such gadgets.

In this paper, a generalization of FL schemes, incorporating sharing a part of raw data, is presented with a proof-of-concept experiment. Besides anonymization, the exchanged data portion can also directly support real-time decision-making. In contrast with cryptographical approaches, the proposed FL scheme can guarantee a certain level of privacy during the whole lifetime of IoT devices or AVs.

Keywords: privacy-preserving; federated learning; raw data sharing; real-time decision-making

1 Introduction
Recent years’ evolution of communication, electrical, and battery technology helped the emergence and the spreading of intelligent Internet of Things (IoT) devices. These gadgets support numerous fields such as renewable energy, manufacturing, environment monitoring, and navigation. Moreover, we can also encounter IoT devices in health, medicine, smart cities, and transportation.

1.1 Problem Statement
Besides collecting zettabytes of data, IoT tools can also support real-time decision-making by employing artificial intelligence (AI) solutions [1]. As IoT solutions can monitor our health [2], or process and influence our transportation habits [3], a security breach can lead to a significant privacy violation.

In recent years, federated learning (FL) [4] has also gained popularity in IoT. Besides saving network resources and enhancing learning quality, FL can provide defense against certain kinds of cyber attacks [5]. However, FL also has limitations. Without direct raw data sharing, we might lose the potential of IoT devices in real-time decision-making. Moreover, FL provides a theoretical defense against active cyberattacks, but an honest-but-curious eavesdropper can easily carry out a membership inference attack if the participants’ data is not independent and identically distributed (non-IID) [6]. To this end, in this paper, we provide a proof-of-concept experiment of a novel privacy-preserving FL scheme based on statistical anonymity achieved by a small amount of raw data sharing [7].

1.2 Related Works
Our privacy preservation is fundamentally similar to the local differential privacy techniques [8], but in our case, there is a theoretical guarantee that the method does not cause any loss of model utility. Additionally, raw data sharing is possible among participants, which can be vital in the case of Autonomous Vehicles (AVs), which are one of the most anticipated manifestations of IoT devices [1].

The proposed algorithm allows AVs to communicate (raw data) in different situations, e.g. while moving in a platoon [9] or passing an intersection [10]. Moreover, the AVs can also build predictive models e.g. for traffic flow forecasting. In contrast with [11], the novel FL scheme does not require any additional infrastructural background.

According to the survey in [12], many privacy-preserving FL schemes are based on cryptography. We shall also note that IoT devices have a relatively long lifespan especially compared to mobile phones or desktop computers. While the latter tools have only a couple of years of lifetime, IoT devices have to be operational for more than 10 years in the case of smart city
For AVs, the lifespan can be even longer: for traditional vehicles, it is not unprecedented that 15 years old cars are on the road [14], and originally designed to travel at least 160000 km. In Germany, an average passenger car traveled 17400 km in 2017 [15], implying that vehicles have more than 9 years of designed lifespan.

Therefore, when designing IoT devices, we shall also consider the technological advances of the upcoming ten years. For example, the deployment of quantum computers will have a fundamental impact on present cryptography [16]. Consequently, privacy-preserving FL techniques based on the affected cryptographic primitives would be no longer adequate. Supposing the arrival of quantum computers, statistics, or perturbation-based privacy-preserving FL schemes might be more effective for IoT devices in the long run.

Using pseudonyms might be an alternative solution for privacy-aware FL of AVs [17, 18]. In [19], a recommender system is presented of which suggestions are based on the sequence of the user actions. These actions can correspond to a concrete route in a city. Moreover, this recommender system does not need to know the user profiles, making it capable of performing well even when pseudonyms are used.

In the rest of this paper, we will evaluate a novel privacy-preserving FL strategy based on a minimal amount of raw data sharing. As a proof-of-concept experiment, we will model data collecting and sharing as a digit classification problem using the MNIST dataset. We will partition the MNIST dataset [20] into ten non-IID parts. We will distribute these pieces of datasets among ten learning participants, supposing that an honest-but-curious server agent is present. The goal of the system is to recognize the hand-written numbers. During the training phase, each communication message is supposed to be broadcasted on open channels. This proof-of-concept experiment can be seen as a simplified problem of AVs (the participants) traversing on a road network, collecting some data (non-IID data portions) from which they can build a predictive model (the pattern recognition).

In section 2, we will give a detailed description of the FL system. Section 3 defines the metrics used to evaluate the system performance. We have already mentioned that the proposed method requires a minimal amount of raw data sharing. Section 4 introduces some strategies to select which part of the participants’ data has to be shared. The capabilities of the adversarial server are detailed in section 5. Section 6 presents the results of our experiments which are discussed further in section 7. Finally, section 8 concludes this paper.

2 The Federated Learning Setup
As a proof-of-concept experiment, we set up a simple FL scenario, see in Figure 1, in which the 10 participants’ goal is to learn to recognize handwritten numbers of the MNIST [20] dataset. Each participant has a unique, distinct portion of the MNIST training data. There is also an honest-but-curious server in the setup, aiming to uncover which participant owns a particular part of the MNIST data.

2.1 Partitioning the MNIST dataset
Naturally, if the training datasets are independent and identically distributed (IID, e.g. as in Figure 2), the server cannot distinguish them. On the other hand, the malicious server can be successful if the participants train on non-IID data. Unfortunately, IoT devices or AVs seldom have IID measurements due to the natural heterogeneity of the environment.

To model this, we divided the original MNIST training dataset into 10 partitions in the following way: Every partition contains half of the samples of a particular digit, while the other 9 partitions received the remaining part. Consequently, this forms a non-IID data partitioning, see Figure 3. We will match the data partitions with a particular participant, therefore every one of the 10 participants can train to recognize a specific digit more accurately than the others. We refer
to this partitioning as “non-iid50”, and to the digit with most populous number of samples as the representative digit, which effectively identifies the participant. Following the regular notation, we call the pixel map of the dataset as features, and the corresponding digit as labels.

2.2 Capabilities of the Server

The server that coordinates the FL process is believed to be honest-but-curious. During the operation, the server aggregates the received model parameters by the FedAvg algorithm \[4\]. On the other hand, the server is also interested in carrying out a membership inference. Its specific goal is to determine which participant possesses a particular “non-iid50” partition. To this end, it stores the received model parameters of the participants of the entire training process. Moreover, it has access to the whole MNIST dataset. Therefore, it can measure the actual accuracies of each participant in every communication round.

2.3 Capabilities of the Participants

The participants train their local models in the FL system. They have a unique, distinct “non-iid50” partition for training. Since the original MNIST dataset contains 54210 examples\[5\], the 10 participants have around 5421 samples each.

The participants can share and receive additional data from each other during the training process. At the beginning of each epoch, the received data gets merged with the participants’ own measurements. After that, a new training set is sampled from this mixture such that it will have a size of 5421 samples and approximates the uniform label distribution. We only mixed the most recently received samples with the participants’ data in our experiments. However, it is theoretically possible to maliciously store the whole exchanged dataset to make membership attacks against fellow participants. Consequently, participants might neither be trusted. On the other hand, as IoT devices usually have limited storage and processing capabilities, using only the most recent data may cover the majority of use cases.

2.4 Training Process

In the training process, the participants train their neural network implemented in Pytorch, while Flower \[21\] provided the federated system. The neural networks have a stochastic gradient descent (SGD) optimizer with fixed hyperparameters\[2\] for every measurement. We used pre-computed datasets to minimize their influence on the results’ variance.

The FL rounds start with the server gathering a random initialization of the trained parameters from one of the participants. Then, the server broadcasts this random initialization to each participant. From this point, a traditional FL process begins with the possibility of sharing some raw data as the first step of each communication round. Within each round, the server selects all of the 10 participants for training, and the participants execute exactly 1 epoch of learning. Hence, the server will have a very fine resolution of the performance of the participants.

In each experiment, we performed 500 training rounds.

3 Metrics

We aim to show how different raw data sharing strategies can help to mitigate a possible membership inference.\[1\] The entire MNIST dataset contains approximately 60000 samples; however, digit 5 only has 5421 samples. Consequently, we used only 5421 samples from each digit label.\[2\] With a learning rate of 0.01 and a momentum of 0.9.
ence attack, while it also helps the training process itself, resulting in faster convergence and higher final accuracy. For comparing the strategies, we defined metrics to measure utility and also the revealingness of the shared data.

3.1 The Baseline
To have an absolute baseline, we have run a non-federated training with the whole MNIST train dataset of the neural network (following [22]) used by the participants. In the following, we will refer to this measurement as the baseline. We will compare the learning performance of the different methods to this case.

However, a non-federated scenario cannot be the reference for assessing the revealingness. To this end, we run baseline measurements of a federated system trained on IID data. We will denote this case as “fed_iid”. The 10 participants of the “fed_iid” case trained on the MNIST dataset evenly divided into 10 partitions. In this case, the partitions had an approximately uniform label distribution.

3.2 Accuracy
We define a method’s accuracy ($a_i$) as the rate of the number of successfully recognized ($r_i$) and the total number of samples ($n_i$) in a given training round $i$:

$$a_i = \frac{r_i}{n_i}. \tag{1}$$

3.3 Convergence Speed
The convergence speed (“CS”) is a relative metric to the baseline measurement. The “CS” is the first training round from which a method’s accuracy is at least the 95% of the baseline accuracy. Therefore, a faster process will have a lower “CS” value.

3.4 Average Accuracy after Convergence
After reaching the “CS” point, a method shall provide a roughly stable accuracy in the remaining training epochs. Denoting the accuracy values after the “CS” point as $a_{[CS,500]}$, the average accuracy after convergence (“AAC”) is the $\overline{a}_{[CS,500]}$ mean value of these roughly stable accuracy values compared to the baseline accuracy.

3.5 Revealingness
Apart from measuring the learning performance, we also calculate how successful the server is in the membership inference. To this end, we compute how many of the 10 “non-iid50” partition were correctly matched with the corresponding participants. Let us denote the number of successful matchings as $m$. Thus, revealingness $R$ is calculated as:

$$R = \frac{m}{10}. \tag{2}$$

According to [23], the $R$ value determines the privacy of the users as it is also the correctness of the server. By repeating the experiments, the average of the resulted $R$ values will asymptotically estimate the probability of a successful membership inference attack.

4 Data Sharing Strategies
Traditionally, hiding raw data is one of the most important benefits of the FL scheme. However, it is also a threat when the participants do not possess IID data. This vulnerability might be mitigated if the participants are allowed to share some of their raw data. Apart from security, the exchanged data can be beneficial for real-time decision-making if the participants are in the same physical environment.

Unfortunately, selecting an appropriate part of data for sharing is quite challenging. In this section, we present some data-sharing strategies, also considering the level of trust between fellow participants.

4.1 No Data Sharing
Let us begin with the traditional FL setup that prohibits exchanging raw data. We call this case the “fed_noshare" strategy, and the participants will train on their own “non-iid50” data portion. As participants share no data, they do not have to trust each other either.

However, they are likely to be vulnerable to the membership inference attack of the server. Considering the training data is highly non-IID, the server can be accurate even up to the $R = 1.0$ level. Moreover, as participants are believed to learn to recognize a particular digit (their representative digit) accurately due to their training data, the FL process might be slower than other methods due to the weighted model averaging.

4.2 Preselected Data
As the privacy leakage mainly comes from the non-IID data portions, we shall balance the datasets to be IID. To this end, let us denote $n_c$ the number of data categories, $n_p$ the number of participants, and $n_s$ the number of samples possessed by one participant. Furthermore, $p$ is the rate of the representative digit assigned to one participant compared to the total number of measurements of that particular category. E.g. if a participant has 50% of the samples of digit five, then $p$ will be 0.5. With this notation, eq. (3) describes how to calculate the $x$ required number of samples per category per participant to be shared in case of a datasets distributed similarly to “non-iid50”.

$$\frac{n_s(1-p)}{n_c-1} + (n_p - 1)x = \frac{n_s}{n_c}. \tag{3}$$
On the left-hand side of eq. (3), the first summmand expresses the number of samples a participant owns from a given (not its representative digit) category. The second term describes how many records shall be received from fellow participants to reach the uniform distribution, expressed on the right-hand side.

Substituting the parameters of the “non-iid50” partitioning as described in Table 1 into eq. (3), we will get that each participant need to share approximately $x = 27$ records of each digit to balance their data to IID.

Now, we know how many records have to be shared. Supposing that the participants do not trust each other, we shall sample the required data for sharing before the federated training process only once. Then, with this precomputed data, every participant can balance its training data portion to be IID, hence reducing the server’s probability to carry out a successful membership inference attack. In the meantime, sharing $10x = 270$ records out of a “non-iid50” portion of size 5421 means that sharing only approximately 5% of the local raw data is sufficient. Additionally, as each participant has roughly 301 records per category, they can sample 27 out of it without any privacy risk.

During the training process, to utilize the maximal amount of data that a participant has, it merges with the preselected, shared data with its own records at the beginning of each training round. As the representative digit category will have more diverse samples at the beginning of each training round. As the representation with the same number of digits out the N possibilities, only $f = \binom{301}{26} = \frac{300!}{9! \cdot 27!}$ will contain a specific sample. Similarly, the probability of a duplicate coming from the representative digit category is:

$$P_{sd} = \frac{27}{2710}$$

Following a binomial distribution, the probability $P_{do}$ that a sample, coming from a non-representative category, will not be repeated in r communication rounds can be calculated as:

$$P_{do} = \binom{r}{0} \cdot (P_{so})^0 \cdot (1 - P_{so})^r = \frac{27}{301}^r$$

Similarly, the probability $P_{dd}$ that a sample, coming from the representative category, will not be repeated in r communication rounds is:

$$P_{dd} = \left(\frac{2683}{2710}\right)^r$$

### Table 1 Parameters to balance the “non_iid50” data partitioning.

<table>
<thead>
<tr>
<th>parameter</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n_s$</td>
<td>5421</td>
</tr>
<tr>
<td>$n_c$</td>
<td>10</td>
</tr>
<tr>
<td>$p$</td>
<td>0.5</td>
</tr>
<tr>
<td>$n_p$</td>
<td>10</td>
</tr>
</tbody>
</table>

In this case, we can execute raw data sampling and sharing before each training round. This method is called the “fed_resel”.

Before discussing the “fed_resel” method in detail, we will prove that the participants must trust each other. Let us assume that there is an honest-but-curious participant that intends to infer whether a fellow client possesses a particular “non-iid50” data portion. The malicious participant can only observe the data shared by its target. Firstly, we shall note that it has a $\frac{1}{5}$ probability of randomly matching a specific data portion with a participant. The participant can also remember the series of shared records of its target from the previous r communication rounds. The key to the identification is to check the (absence of any) record duplications. A duplicated record can have two origins. They either come from the representative digit category of the target’s data or from another digit. An attacker can exploit this difference to distinguish the representative digit category from the other ones; consequently to carry out a successful membership inference.

After receiving a duplicate, we can calculate the $P_{so}$ chance that this sample does not come from the representative digit category as follows. Altogether, we have roughly $N = \binom{301}{27}$ possibilities of sampling records from a given digits out the N possibilities, only $f = \binom{301-1}{26} = \frac{300!}{9! \cdot 27!}$ will contain a specific sample. From here, we can calculate $P_{so}$ simply as:

$$P_{so} = f \cdot \frac{1}{N} = \frac{300!}{274! \cdot 26!} \cdot \frac{274! \cdot 27!}{301!} = \frac{27}{301}$$

Similarly, the $P_{sd}$ probability of a duplicate coming from the representative digit category is:

$$P_{sd} = \frac{27}{2710}$$

### 4.3 Reselected Data

Following the “fed_resel” method of section 4.2, we can either maximize the data utilization and training performance or minimize the revealingness. However, it is possible to improve these metrics if fellow participants trust each other. Furthermore, it is also necessary to separate the participant-to-participant and server-participant communication channels. Because...
Now, to prove that participants must trust each other, we shall find a finite bound for the number of communication rounds \( r \), such that above this bound, the malicious participant is more likely to distinguish the two category types than random choosing. I.e. if it observes a record duplication, then above a certain \( r \), the probability difference between coming from the dedicated category and from the others shall be greater than \( \frac{1}{9} \):

\[
|P_{dd} - P_{do}| > \frac{1}{9} \tag{8}
\]

where \( r = 0, 1, 2, \ldots \)

After substituting eq. (6) and eq. (7) into eq. (8), it is easy to check that \( r = 1 \) is not revealing yet, but \( r \geq 2 \) is. Consequently, “fed_presel” method (where \( r = 0 \), since in that way there are only one data sharing in the beginning) works even between untrusted participants, but “fed_resel” indeed requires the trust.

The “fed_resel” case can provide a higher data utilization than “fed_presel”, thus we expect to achieve higher “AAC” values and faster “CS”. Varying the training data shall also reduce the \( R \) revealingness value.

5 Membership Inference

After studying the possibility of the participants hiding their identities, now, let us discuss the capabilities of the honest-but-curious server.

Calculating the accuracies helps the server carry out the membership inference as follows. First, the server selects \( S_0, S_1, \ldots, S_9 \) sets of samples for digits 0, 1, …, 9 respectively from the entire MNIST training database. Upon each communication round, the server estimates the performance of every participant on the \( S_0, \ldots, S_9 \) subsets according to their current model parameters to get \( a_{S_0}, \ldots, a_{S_9} \) accuracy values respectively. Then, the server estimates which \( c_i \) “non_iid50” portion is possessed by a participant by checking the computed accuracies:

\[
c_i = \arg \max S_i a_{S_i}. \tag{9}
\]

According to eq. (9), the server assumes that a digit is recognized most accurately by the participant that possesses the most samples of that digit category. This estimation would mean that the “fed_noshare” case is the most revealing, while the server shall not be successful in the “fed_iid” case, see Figure 4. We can only have a qualitative expectation that “fed_presel” and “fed_resel” cases are in between the first two methods in terms of revealingness.

6 Results

To prove the aforementioned concepts, we have trained the federated system (described in section 2) 10 times for each data-sharing strategy. By fixing the hyper-parameters and training datasets, we can assess the performance of the introduced methods without any further influencing factors.

Table 2 Results, average of 10 measurements.

<table>
<thead>
<tr>
<th>method</th>
<th>“CS”</th>
<th>“AAC”</th>
<th>( R )</th>
</tr>
</thead>
<tbody>
<tr>
<td>fed_iid</td>
<td>143</td>
<td>97.96%</td>
<td>0.04</td>
</tr>
<tr>
<td>fed_noshare</td>
<td>163</td>
<td>96.84%</td>
<td>1.00</td>
</tr>
<tr>
<td>fed_presel</td>
<td>154</td>
<td>96.96%</td>
<td>0.37</td>
</tr>
<tr>
<td>fed_resel</td>
<td>143</td>
<td>96.96%</td>
<td>0.15</td>
</tr>
</tbody>
</table>

Table 2 summarizes the measured results. As expected, “fed_iid” and “fed_resel” provide the lowest “CS” values, which indicates a faster convergence with identical hyperparameters. The “fed_presel” case is 8%, and the original federated scheme, the “fed_noshare” 14% slower based on the average of 10 measurements, see also Figure 5.

According to the AAC values, there is only a negligible difference between the methods. That indicates that after reaching convergence, the FL systems can provide the same performance, regardless of the data sharing strategies.

On the other hand, there is a significant contrast between the data sharing methods regarding the \( R \) revealingness results. The IID (“fed_iid”) case naturally does not pose a privacy threat, while the traditional “fed_noshare” is the most revealing. We shall emphasize that it is so revealing that the server can always carry out membership attacks successfully. To mitigate this privacy issue, raw data sharing seems to be beneficial. Even in an untrusted environment, a minimal necessary set of shared data can noticeably reduce the \( R \) value. In a trusted environment, further improvements are achievable. However, \( R = 0.15 \) is still a higher success rate for the server than the random matching; the probability of a successful membership attack is almost a magnitude lower compared to the traditional FL scheme.

A comparison of Figure 4 and Figure 6 also demonstrates how raw data sharing can reduce the revealingness. Moreover, in Figure 6, we might also observe that while the per digit accuracy distributions of the clients using the “fed_resel” method are almost identical to each other, “fed_presel” produces a slight difference among the clients. This phenomenon is due to the variance of the training data. The “fed_presel” strategy provides a smaller variety of data. On the other hand, “fed_resel” can assure that every client might train on each data sample sooner or later, resulting in similar accuracy patterns.
7 Discussion

In this paper, we have considered the raw data sharing between FL clients in both a trusted and untrusted environment with the presence of an honest-but-curious server. To make proof of concept experiments, we used the well-known MNIST dataset and divided it into 10 non-IID partitions. Hence, we knew the exact distribution of the data, which helped the calculations. In real-world problems, these distributions might not be known. Therefore, we might be unable to analytically compute the required amount of raw data to be shared.

The characteristics of the MNIST dataset might also slightly influence our results as some of the digits (e.g., 0, 1, 6; see Figures 4 and 6) can be easier to recognize. Nevertheless, every real classification problem might have categories that are simple to distinguish. In our particular case, that might be the reason why “fed_iid” achieves an $R = 0.04$ score, instead of the theoretical $R = 0.1$, because the participants are likely to be more accurate on the easier digits than a randomly selected one, which results in an even lower $R$ value. We might conclude, that quantitative $R$ values are problem-specific (or even neural network architecture specific), but raw data sharing can effectively mitigate the possible privacy threats.

Let us continue with the used attacker model, which had only one restriction: it was not aware that a perfect matching exists between the participants and the data portions. Consequently, the server uses an arg max function in eq. (9) to estimate the best match instead of algorithmically finding a maximal or perfect match-
Figure 5 Learning performances of different raw data sharing methods. Average accuracy on the validation dataset of 10 training processes. Blue area indicates the 95% of the baseline accuracy.

Figure 6 Example of revealingness of “fed_presel” and “fed_resel” methods. When enough raw data is shared, the participantA-digit 0, participantB-digit 1, ..., participantJ-digit 9 matching is unlikely to be observed.
ing. In most use cases, a perfect matching does not even exist between data portions and participants, so this approach models most problems adequately.

Additionally, there is no other condition for the attacker. The server demonstrated in this paper:

1. Can access all train and test data.
2. Remembers all historical parameters.
3. Besides knowing the model parameters, it possesses the participants’ neural network structure. Hence, it can evaluate their performance on its own.

The above observations suggest that the server is possibly more powerful than the majority of the real-world attackers could be. If a centralized entity suffices condition 1, operating a FL system is not efficient as a single neural network can substitute it. Satisfying condition 2 and 3 would require either a vast amount of storage or computing capability. We shall also note, that as the proposed methods are based on statistical indistinguishability, thus they can provide even quantum security as well.

Our last observation is that we have worked on a classification problem in this paper. That made it simple to assess the accuracy values in eq. (1), where \( r_i \) is only a sum of exact binary indicators. Expressing accuracy in regression problems is at least as challenging as in the classification case since the regression accuracies can have continuous values. Thus, accuracy in the latter case can be a probability itself, which can be even more challenging to the server.

8 Conclusions

We have shown the design and evaluation of a novel FL scheme incorporating raw data sharing. Although sharing raw data was considered a privacy risk, it can also help to achieve anonymous. Moreover, it can also directly support real-time decision-making.

Our experiments showed that sharing a fixed subset of the participants’ training dataset (“fed-preSel”) helps mitigate the success rate of a passive membership inference attack of an honest-but-curious server. If the participants can trust each other, and can communicate with each other on a separate channel, they can exchange raw data more often (“fed_resel”). Thus, it can reduce the success rate of the adversary even further, close to the theoretical success rate corresponding to an IID data distribution.

As the presented method has only one cryptographic assumption (that the server is not aware that a perfect matching between data portions and participants exists), it will provide the designed privacy levels even after the emergence of quantum computers. Additionally, the scheme does not require encryption and decryption, which is beneficial in the limited power condition of IoT devices.

In the proof-of-concept experiments, we used the MNIST dataset, but the proposed FL scheme can be generalized to other problems. For adaptation, we shall accurately calculate the size and distribution of the shared raw data portion. E.g. for a world prediction task of an intelligent keyboard, we can use the world distribution of a given language.

Our future research focuses on implementing the demonstrated method for data collection and machine learning of AV participants. In that problem, it is important to hide the routes of the vehicles, as it is critically privacy sensitive; moreover, it is also the source of the non-IID nature of the data. On the other hand, a predictive model can forecast the traffic situations in a given area; hence, it can mitigate congestion and harmful pollution. Using the road-network layout and the traffic counts as prior knowledge, we can perform computations similar to the ones presented in sections 4.2 and 4.3.

To help the further investigation of the demonstrated FL scheme, the source code of our experiments is published at https://github.com/alelevente/FeLeSh.

List of Abbreviations

- AAC: Average Accuracy after Convergence
- AI: Artificial Intelligence
- AV: Autonomous Vehicle
- CS: Convergence Speed
- FL: Federated Learning
- IID: Independent, Identically Distributed
- IoT: Internet of Things
- SGD: Stochastic Gradient Descent

Declarations

Availability of data and material
The experiments presented in this paper were carried out using the MNIST data set, which is publicly available under various resources (e.g. at http://yann.lecun.com/exdb/mnist/, or in csv format at https://www.kaggle.com/datasets/oddrationale/mnist-in-csv at the time of the submission of this paper). Data preparation and the obtained results are reproducible following our source codes published at https://github.com/alelevente/FeLeSh.

Competing interests
The authors declare that they have no competing interests.

Funding
This research was funded by the National Research, Development, and Innovation Fund of Hungary under Grant TKP2021-EGA-02.

Author’s contributions
This paper outlines some findings of the Ph.D. research work of Levente Alekszejenkó. He is responsible for the concept of the raw data sharing based FL mechanism, together with the corresponding source codes. He wrote the majority of the present paper, including the illustrations. Tadeusz Dobrowiecki is the supervisor of Levente Alekszejenkó. His questions, ideas, and suggestions supported designing the experiments and evaluating the results. He helped to correct the drafts of this paper. He, furthermore, also extended the text at numerous points.
References


