[Array 15 (2022) 100207](https://doi.org/10.1016/j.array.2022.100207)

|  |  |  |
| --- | --- | --- |
|  | Contents lists available at [ScienceDirect](http://www.elsevier.com/locate/array) |  |
| Array |
| journal homepage: [www.elsevier.com/locate/array](http://www.elsevier.com/locate/array) |
|  | | |

CE-Fed: Communication efficient multi-party computation enabled federated learning

Renuga Kanagavelua,∗, Qingsong Weia, Zengxiang Lib, Haibin Zhanga, Juniarto Samsudina, Yechao Yanga, Rick Siow Mong Goha, Shangguang Wangc

a *Institute of High Performance Computing, A\*STAR, Singapore*   
b *Digital Research Institute ENN Group, China*   
c *Beijing University of Posts and Telecommunications, Beijing, China*

|  |  |  |
| --- | --- | --- |
| A R T I C L E | I N F O | A B S T R A C T |
| *Keywords:*  Federated learning  Edge computing  Multi-party computation Committee selection | | Federated learning (FL) allows a number of parties collectively train models without revealing private datasets. There is a possibility of extracting personal or confidential data from the shared models even-though sharing of raw data is prevented by federated learning. Secure Multi Party Computation (MPC) is leveraged to aggregate the locally-trained models in a privacy preserving manner. However, it results in high communication cost and poor scalability in a decentralized environment. We design a novel communication-efficient MPC enabled federated learning called CE-Fed. In particular, the proposed CE-Fed is a hierarchical mechanism which forms model aggregation committee with a small number of members and aggregates the global model only among committee members, instead of all participants. We develop a prototype and demonstrate the effectiveness of our mechanism with different datasets. Our proposed CE-Fed achieves high accuracy, communication efficiency and scalability without compromising privacy. |

**1. Introduction**

Artificial intelligence (AI) techniques find useful applications in various domains, such as healthcare, smart building, autonomous ve-hicles, remote monitoring and predictive maintenance of machines in manufacturing plant. The training of such AI models requires large amount of data to achieve acceptable accuracy and throughput and thus improving the user experience. The huge amount of data gener-ated at different organizations are aggregated at a cloud-based server to produce more-effective inference model. Though this approach is beneficial, the transfer of sheer volume of data to the centralized server burdens the back-bone network. It also introduces long latency [1], which is not acceptable for applications where real-time decisions are required, like self-driving cars and automated car assembly plant [2]. Apart from those, the legal restrictions and rising concerns about shar-ing private information [3] among data owners make them reluctant to send their data to the data centre for the model training [4]. As a result, huge volume of data generated by individual organizations remains in the form of fragmented data silos.

To address the above-mentioned problems, Federated learning (FL) [5] was introduced by Google’s AI research team. It has emerged as the

intersection of deep learning and edge computing, where the training dataset remains in the hands of data owners and there is no need to pool data into a single location. Instead, model training is brought to the edge of the network, so that data never leaves the network and only the model is sent to the central coordinator for aggregation as shown in Fig. 1. It enables the clients to learn a model without sharing the raw data and not relying on any trusted third party to hold the data. However, FL depends on the central server/coordinator for aggregating the model. All clients participating in the FL should have a complete unanimity on the role of central server/coordinator as model aggregators. The central server/coordinator may encounter single point of failure which would crash the entire system. We choose the decentralized federated learning framework [6] to overcome this issue. It is a server-less, decentralized approach, where clients commu-nicate directly among themselves without a central server/coordinator as shown in Fig. 2.

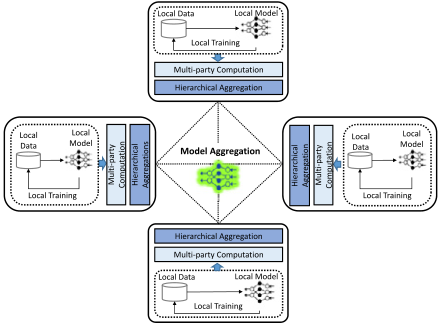
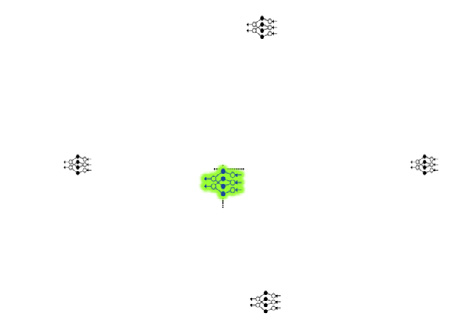
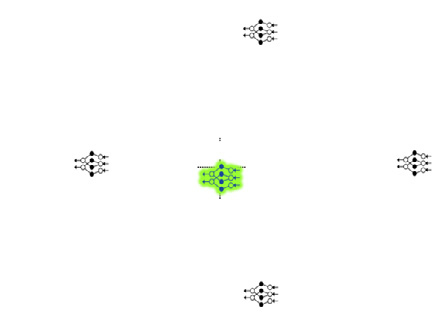
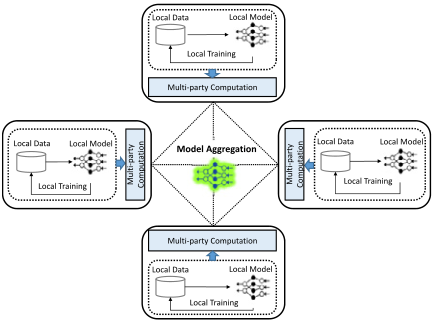
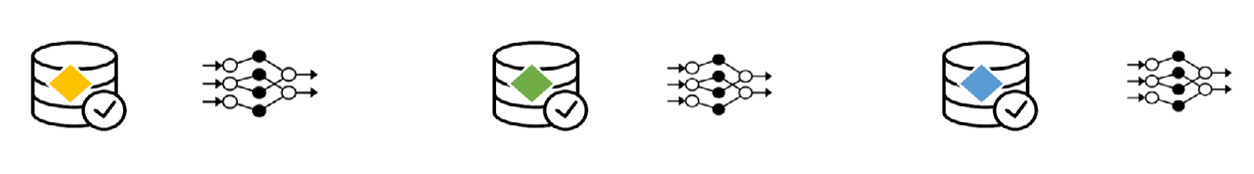
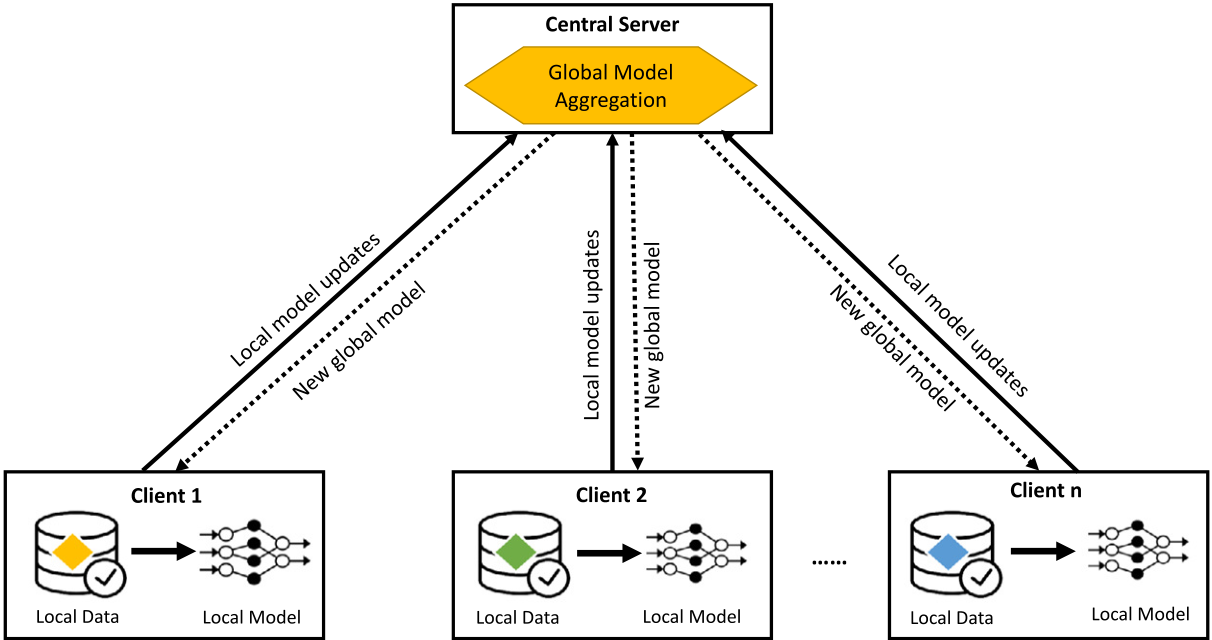
Though federated learning keeps the training data private to the local user, still it is vulnerable to various privacy attacks during the training process and causes privacy leakage. There is also a possibility

∗ Corresponding au[thor.](mailto:renuga_k@ihpc.a-star.edu.sg)

*E-mail addresses:* [renuga\_k@ihpc.a-star.edu.sg](mailto:renuga_k@ihpc.a-star.edu.sg) (R. Kanagavelu), [wei\_qingsong@ihpc.a-star.edu.sg](mailto:wei_qingsong@ihpc.a-star.edu.sg) (Q. Wei), [zengxiang\_li@outlook.com](mailto:zengxiang_li@outlook.com) (Z. Li), [zhang\_haibin@ihpc.a-star.edu.sg](mailto:zhang_haibin@ihpc.a-star.edu.sg) [(H. Zhang),](mailto:renuga_k@ihpc.a-star.edu.sg) [junia](mailto:renuga_k@ihpc.a-star.edu.sg)[rto\_samsudin@ihpc.a-star.edu.sg](mailto:juniarto_samsudin@ihpc.a-star.edu.sg) [(J. Samsudin),](mailto:wei_qingsong@ihpc.a-star.edu.sg) [ya](mailto:wei_qingsong@ihpc.a-star.edu.sg)[ng\_yechao@](mailto:yang_yechao@ihpc.a-star.edu.sg)[ihpc.a-star.edu.sg (Y. Yan](mailto:zengxiang_li@outlook.com)g), [gohsm@ihpc.a-star.edu.sg](mailto:gohsm@ihpc.a-star.edu.sg) [(R.S.](mailto:zhang_haibin@ihpc.a-star.edu.sg)M. Goh), [sgwang@bupt.edu.cn](mailto:sgwang@bupt.edu.cn) [(S. Wang).](mailto:juniarto_samsudin@ihpc.a-star.edu.sg)

<https://doi.org/10.1016/j.array.2022.100207>  
[Received 10 March 2022; Received in revise](https://doi.org/10.1016/j.array.2022.100207)d form 10 June 2022; Accepted 10 June 2022   
Available online 18 June 2022   
[2590-0056](http://creativecommons.org/licenses/by-nc-nd/4.0/)/© 2022 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY-NC-ND license ([http://creativecommons.org/licenses/by-](http://creativecommons.org/licenses/by-nc-nd/4.0/)

[nc-nd/4.0/](http://creativecommons.org/licenses/by-nc-nd/4.0/)).



*R. Kanagavelu et al.*  *Array 15 (2022) 100207*

**Fig. 1.** Central federated learning.

**Fig. 2.** Decentralized FL.

secret shares of locally trained model with all other clients in the FL as shown in Fig. 3. As a result, when the number of FL clients increases, the communication overhead also increases in an exponential manner. These observations motivated us to propose a novel communication-efficient MPC enabled federated learning called CE-Fed, with the objec-tive of reducing the communication overhead and scalability. The key approach of our proposed CE-Fed is to select a few clients as the com-mittee members, who perform the aggregation of the local models of all FL clients in a hierarchical manner using the MPC service. Therefore it avoids sharing the model parameter from each client to everyone else in the FL list. Our proposed CE-Fed is executed in two phases. In the first phase, we group the FL clients that are located close to each other. The local models of all FL clients in the same group are securely aggregated using MPC to form an intra-group model. Based on the latency, one client is elected from each group to form the aggregation committee. In the second phase, the committee members work together to aggregate the inter-group models using MPC, as shown in Fig. 4.

The main contributions of our paper are summarized as follows:

|  |  |
| --- | --- |
| **Fig. 3.** MPC enabled FL. | • We quantitatively analysed the communication overhead when  secure MPC is integrated into decentralized federated learning  with experimental results.  • We propose a hierarchical model aggregation to reduce the com-  munication cost incurred in MPC enabled Federated Learning.  • We propose latency-based committee election algorithm for MPC-  based hierarchical model aggregation.  • We demonstrate the effectiveness of our decentralized  communication-efficient MPC based federated learning through  extensive experiments on various datasets. |

**Fig. 4.** MPC enabled FL-hierarchical aggregation.

to extract the local user’s dataset by reverse engineering the commu-nicated model parameters [7–9]. To ensure confidentiality and provide data privacy guarantee of the model aggregation, privacy-preserving techniques, such as Differential Privacy (DP) [10], secure Multi-party Computation (MPC) [11], and Homo-morphic Encryption (HE) [12,13] can be used together with FL [8].

Secure MPC allows for secure computation among multiple parties to compute a function jointly over their sensitive input data. All partici-pating parties know only the output while keeping those inputs private. The parties can learn only the final output.

Though MPC has the advantage of secure data sharing, it tends to have significant communication and computation overhead when implemented on a large-scale decentralized federated learning sys-tem [14]. In the traditional MPC-enabled FL, each client exchanges its

2

*R. Kanagavelu et al.*  *Array 15 (2022) 100207*

has no control over the data as well as the constructed machine learning model, since both are no longer residing at the premise of the data owner. There is a possibility that the centralized service providers can obtain extra revenue on the data as well as on model and use them for some illegal purposes. This cannot be prevented by data owners. Federated learning (FL) has emerged as an effective approach that introduced by Google’s AI research team in 2017 [15]. The central server send the baseline ML model to clients. Then, each client train its own sub-model using its local data and send the fine-tuned model back to the central server. The central server collects all trained sub-models and aggregates them. Then the aggregated models are sent back to clients out over and over again until they have learnt all there is to learn. Throughout the entire process, raw data are kept on client devices and instead the local models are transferred and shared. The federated model improves its accuracy by aggregating many local models iteratively, taking advantage of complementary data/inputs from a large number of devices.

Federated Learning (FL) has been extended to the collaborations across multiple organizations. It is categorized to horizontal FL and vertical FL [16], based on the data distribution over the sample and feature spaces. In horizontal FL, datasets of different organizations like different industrial organizations, have the same features but different sample sizes. In vertical FL, different organizations like banks, or insurance companies that are located with the same city, have collected data with different features.

Few open-source FL frameworks are available. Google’s TensorFlow Federated (TFF) is a lightweight open source framework, a first attempt in the community. It is designed for android users, to predict keyboard next word on their mobile phones [17] using TFF. FATE [18], is the federated learning framework developed by Webank. It supports various federated learning methods. Pysyft [19] is a python library that only supports FedAvg algorithm. It supports MPC and differential privacy. It can either run on stand-alone or on multiple machines and uses web socket API for the communication between different clients. Clara [20] is a framework for building AI accelerated medical imaging workflows. Facebook’s privacy preserving machine learning framework is CrypTen [21].

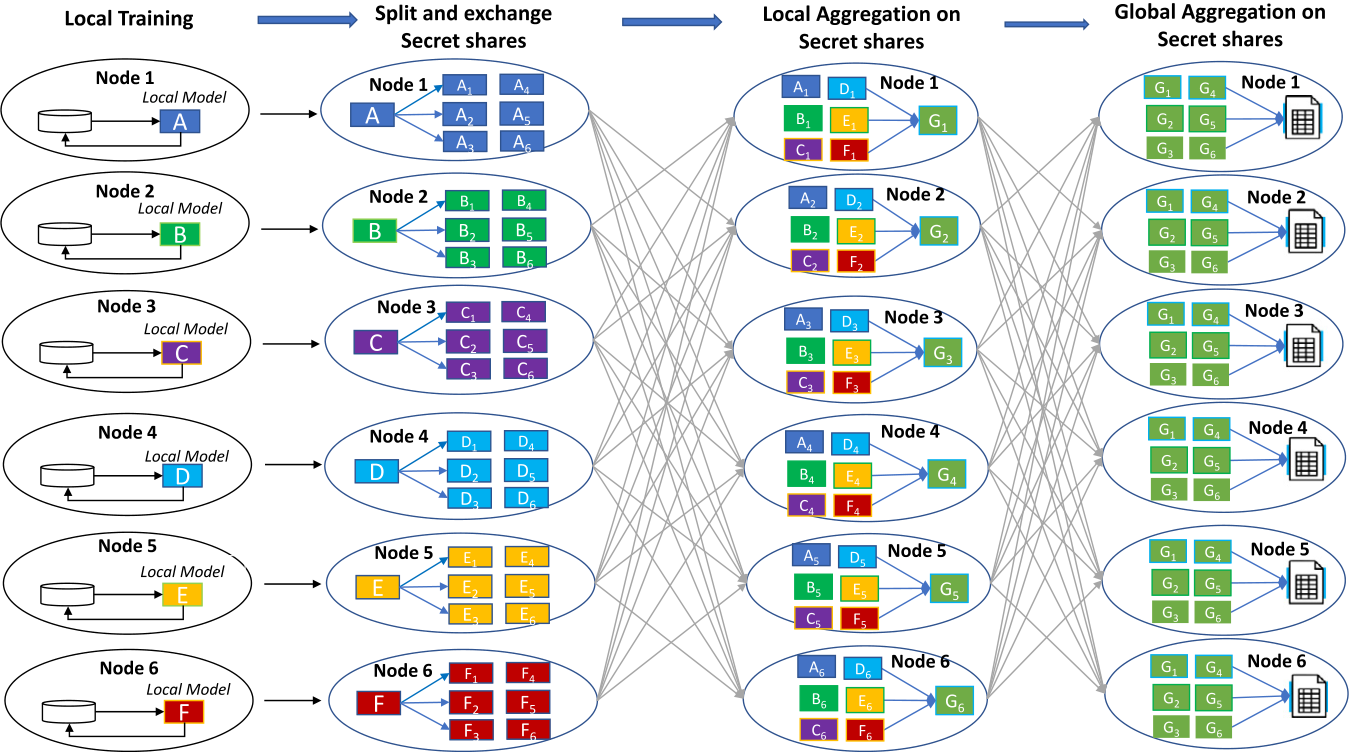
*2.2. Privacy-preserving technologies*

Secure MPC is a privacy preserving technique that allow for secure computation over sensitive data. It was firstly introduced by Yao in 1986 [22]. Garbled circuits and secret sharing are the two dominated MPC techniques [23] followed today. The secret sharing is a commonly used MPC protocol. It splits the sensitive data into secret shares. The original data is obtained from the combination of these secret shares. The clients cannot learn anything other than final output.

In federated learning, the model parameters and gradients are shared with the server for model aggregation. There is a possibility that the malicious user can intercept the model parameters and could per-form reverse engineering to extract the sensitive data, while it is shared with the server on federated learning [24,25]. To address this issue, Multi party computation methods like additive secret sharing [26] or Shamir secret [27] sharing can be used to encrypt the gradients/model updates before performing the aggregation so that no one will be able to see the gradients.

Differential Privacy (DP) is adopted by researchers recently [28] to ensure data security and confidentiality. Data privacy is protected by adding random noise to data. The introduction of noise results in a compromise between security and accuracy. It is not suitable for FL as adding noise to model parameter data from each participating party may affect the accuracy of global model [29].

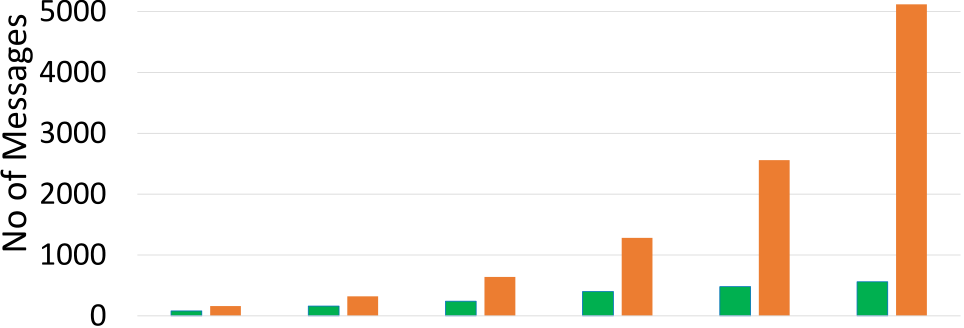
3



*R. Kanagavelu et al.*  *Array 15 (2022) 100207*

**Fig. 5.** MPC- enabled model aggregation.





|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |



**Fig. 6.** Communication overhead.

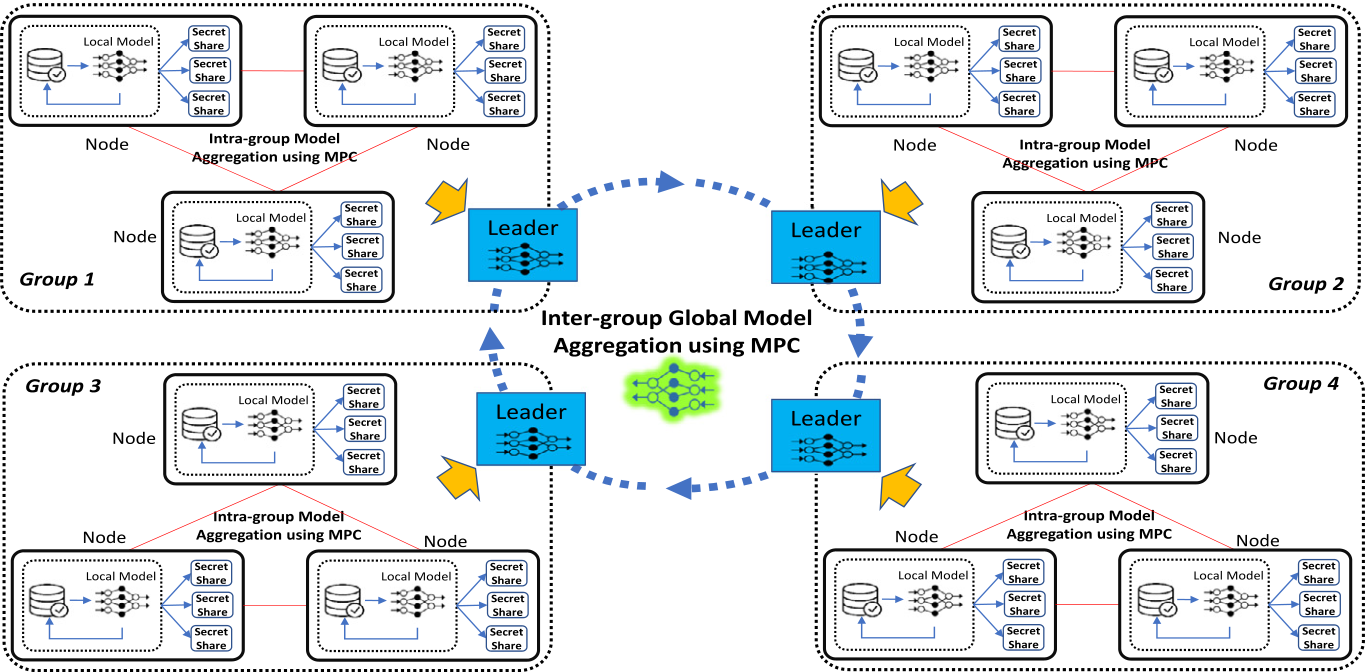
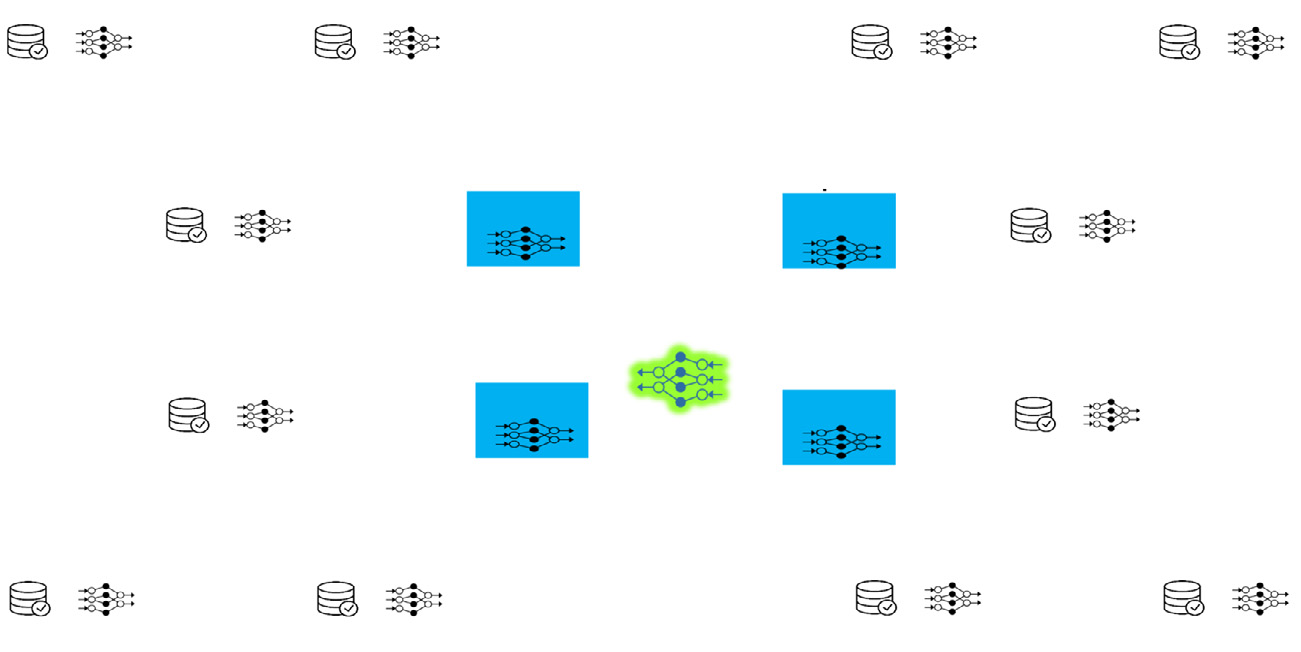
We have carried out a preliminary experimental study to get insights into how the consideration of the number of clients impacts on the communication overhead in privacy preserving FL using traditional MPC. We evaluate the performance on MNIST dataset [41], using the following parameters: total number of clients participated(N) = varying from 4 to 128; Local epoch(E)=10; Learning rate(lr)=0.01; batch size=10; number of communication rounds(T)=50. The com-munication overhead for two scenarios: FL with traditional MPC and FL without MPC is shown in Fig. 6. It is observed from Fig. 6 that the number of communicated messages in MPC enabled FL increases with increasing number of clients, as each client exchange their secret shares of model parameters with all the other clients during each communication round. The results show that, the number of messages exchanged is 5×–10× more when compared with FL without MPC. These findings motivated us to propose a novel FL framework to reduce the communication cost incurred in traditional MPC enabled Federated Learning.

**4. CE-Fed framework**

In this section, we describe the proposed CE-Fed framework as shown in Fig. 7, with the objective of reducing the traditional MPC communication cost.

The proposed CE-Fed selects a few clients as the committee mem-bers who use MPC service to aggregate the local models of all FL clients in a hierarchical manner. Therefore it avoids sharing the model parameter from each client to everyone else in the FL list. Our proposed CE-Fed is executed in two phases. In the first phase, we group the FL clients that are located close to each other. The local models of all FL

4



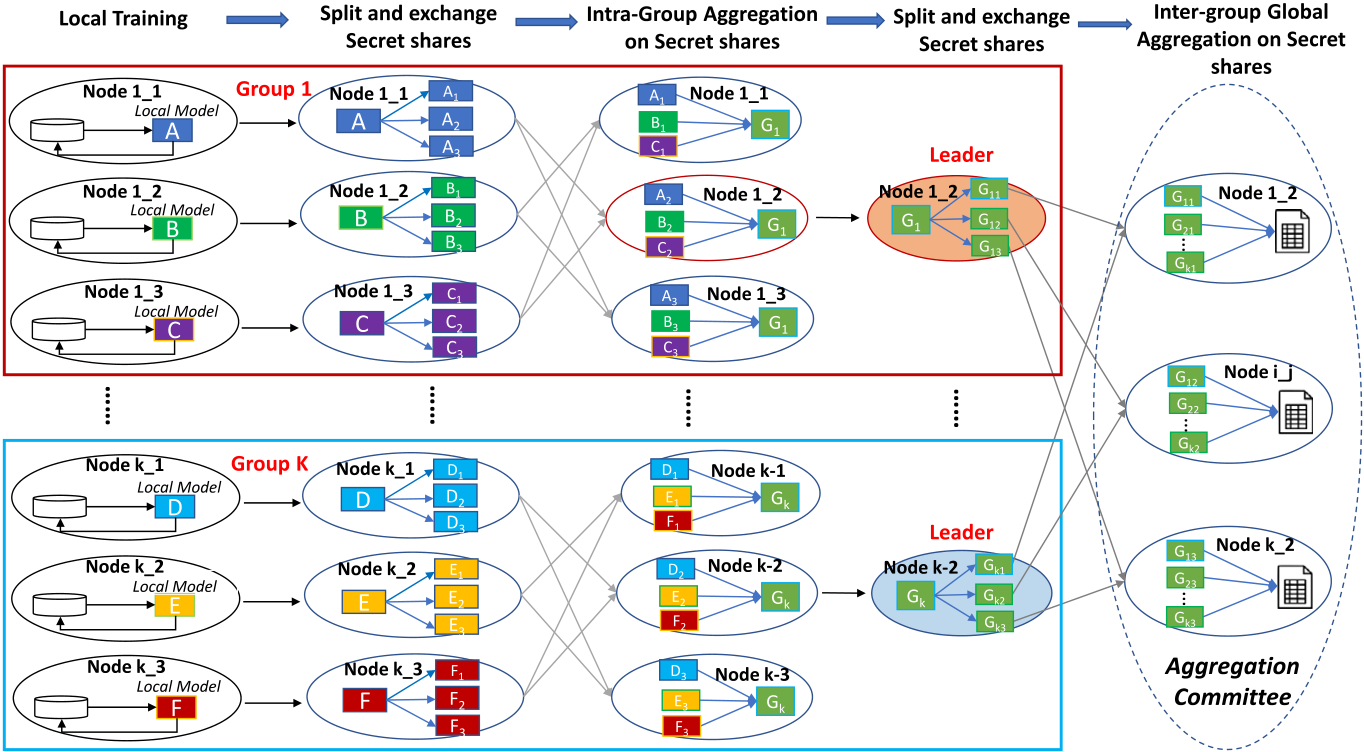
*R. Kanagavelu et al.*  *Array 15 (2022) 100207*

**Fig. 7.** CE-Fed framework.

|  |  |  |  |
| --- | --- | --- | --- |
| **Table 1**  Variables in federated learning framework. | |  |  |
| Category | Symbol | Type | Description |
| *𝑛* | | Int  Int  Int  Int | No. of participating FL clients  Global model training iterations  Local model training rounds  Batch Size of each communication round |
| Federated learning | *𝑎*  *𝑏* |
| *𝑏𝑆𝑖𝑧𝑒* | |
| Committee | *𝑚*  *𝐶* | Int  Array | Elected committee members  committee members List |
| CNN model | *𝑇*  *𝑊* | Tensor  Tensor  Integer | Local model parameters  Parameters/weights of aggregated model  The size of parameters/weights of local/aggregated models |
| *𝑠* | |
| *𝐺* | | Int  Int  Array  Int  Array | Number of mutually exclusive Groups Number of clients in group  Group members List  inter-node latency  Mutually exclusive Group Leaders List |
| Network | *𝑘*  *𝑆𝐺* |
| *𝐿* | |
| *𝑄* | |

|  |
| --- |
| leader is selected based on the average latency with the other clients. Each client is assumed to be at fixed position and the estimated la-tency between any pair of clients is known in advance. The elected group leader may have a chance to participate in the global model aggregation. There is only one leader in the group and rest of them are his followers. The leader sends *Alive* messages to its followers at frequent intervals. If there is no *Alive* message from the leader within a particular time interval, the client with minimal inter-node latency to the other followers will be elected as a new leader. The pseudo code of group formation and leader election is shown in Algorithm 1. Our algorithm considers many mutually exclusive groups with varying number of clients.  **Algorithm 1** Group Leader Election |
| 1: Function{**Leader Selection(G,k, m,C)}** 2: Group Initialization  3: Read the value  4: **for** *𝑖* ∈ [1*, 𝐺*] **do**  5: *𝑚𝑖𝑛* ← *𝐿𝑎𝑡*(*𝑆𝐺*[1])  6: **for** *𝑗* ∈ [1*, 𝑘*] **do**  7: **if** *𝐿*j *< 𝑚𝑖𝑛* **then**  8: *𝐿𝑒𝑎𝑑𝑒𝑟* = *𝑗*  9: **end if**  10: *𝑄* ← *𝐿𝑒𝑎𝑑𝑒𝑟*  11: **end for**  12: **end for**  13: EndFunction |

5



*R. Kanagavelu et al.*  *Array 15 (2022) 100207*

**Fig. 8.** CE-Fed Hierarchical Model Aggregation.

*4.3.1. Intra-group model aggregation*   
 During the training process, all the clients in the same group start training their models locally using their local datasets. Each client in a group has two main tasks:

• Local training on its local data : After extracting the features from raw data, the client’s data analytics module performs local training on its local data.

• MPC Enabled intra-group Model Aggregation : The locally trained model parameters are split into secret shares and securely aggre-gated using MPC protocol.

|  |
| --- |
| Initially, each client locally trains their local dataset for few itera-tions, till the convergence criterion is met. After completing the local training, the clients in the group collaboratively learn the model in a peer-to-peer fashion. As there is no central server available to initiate the model aggregation process in the decentralized environment, any random client *𝐴𝑖* in the group initiates that process. The tensor **𝐓** of individual local model parameters/weights are privacy-sensitive. To preserve privacy, MPC secret sharing technique is used to securely aggregate the tensor **𝐓**, without sacrificing accuracy. The tensor **𝐓** is split into multiple tensors as the secret shares based on the number of clients in the same group. Each client holds one share and exchanges the remaining shares with all the other participating clients in the same group. Then, clients perform secure model aggregation of the secret shares using MPC to obtain their intra-group model. After learning a  **Algorithm 2** Intra-group Model Aggregation |
| 1: Function{**Intra-Group.Model.aggregation(n, u, v, m)}** 2: **for** *𝑛* ∈ [1*, 𝑢*] **do**  3: **for** *𝑎* ∈ [1*, 𝑣*] **do**  4: **for** *𝑏* ∈ [1*, 𝑧*] **do**  5: *𝑇* ← *𝑙𝑜𝑐𝑎𝑙𝑡𝑟𝑎𝑖𝑛𝑖𝑛𝑔*  6: **end for**  7: **for** *𝑡* ∈ [1*, 𝑢*] **do**  8: Split locally-trained model into *u* secret shares) 9: share it with members of aggregation committee 10: **end for**  11: **end for**  12: 13: *𝑆* = *𝑠𝑢𝑚𝑡ℎ𝑒𝑠𝑒𝑐𝑟𝑒𝑡𝑠ℎ𝑎𝑟𝑒𝑠*  **for** *𝑎* ∈ [1*, 𝑢*]&*𝑎* ≠ *𝑥* **do**  14: Broadcast *𝑆* to peer clients  15: Receive *𝑆* from peer clients  16: **end for**  17: *𝑊* ← *𝐼𝑛𝑡𝑟𝑎* − *𝑔𝑟𝑜𝑢𝑝𝐴𝑔𝑔𝑟𝑒𝑔𝑎𝑡𝑒𝑑𝑀𝑜𝑑𝑒𝑙*  18: **end for**  good intra-group model, the group leader sent their group’s intra-group |

6

*R. Kanagavelu et al.*  *Array 15 (2022) 100207*

|  |
| --- |
| **Algorithm 3** Inter-group Model Aggregation |
| 1: Function{**Inter-group.Model.aggregation( Q, m, n)}** 2: **for** *𝑄* ∈ [1*, 𝑚*] **do**  3: *𝑇* ← *𝐼𝑛𝑡𝑟𝑎* − *𝑔𝑟𝑜𝑢𝑝𝑀𝑜𝑑𝑒𝑙*  4: **end for**  5: **for** *𝑄* ∈ [1*, 𝑚*] **do**  6: Split Intra-group model into *m* secret shares  7: share it with members of aggregation committee 8: **end for**  9: **for** *𝑥* ∈ [1*, 𝑚*] **do**  10: 11: S=sum the secret shares **for** *𝑛* ∈ [1*, 𝑚*]&*𝑎* ≠ *𝑥* **do**  12: Broadcast *𝑆* to peer clients  13: Receive *𝑆* from peer clients  14: **end for**  15: *𝑊* ← *𝐺𝑙𝑜𝑏𝑎𝑙𝐴𝑔𝑔𝑟𝑒𝑔𝑎𝑡𝑒𝑑𝐼𝑛𝑡𝑒𝑟* − *𝐺𝑟𝑜𝑢𝑝𝑀𝑜𝑑𝑒𝑙*  16: **end for**  17: **return** *𝑊*  18: EndFunction |

aggregation committee members. The number of messages denoted as *𝐼𝑛𝑡𝑒𝑟*\_*𝐺𝑟𝑝*\_*𝑀𝑠𝑔*\_*𝑁𝑢𝑚* exchanged in the inter-group model aggregation among *m* aggregation committee members can be calculated as follows:

*𝐼𝑛𝑡𝑒𝑟*\_*𝐺𝑟𝑝*\_*𝑀𝑠𝑔*\_*𝑁𝑢𝑚* = (*𝑚* × (*𝑚* − 1)) × 2 × *𝑎*  (5)

The size of messages denoted as *𝐼𝑛𝑡𝑒𝑟*\_*𝐺𝑟𝑝*\_*𝑀𝑠𝑔*\_*𝑆𝑖𝑧𝑒* exchanged in the intra-group model aggregation is

*𝐼𝑛𝑡𝑒𝑟*\_*𝐺𝑟𝑝*\_*𝑀𝑠𝑔*\_*𝑆𝑖𝑧𝑒* = *𝐼𝑛𝑡𝑒𝑟*\_*𝐺𝑟𝑝*\_*𝑀𝑠𝑔*\_*𝑁𝑢𝑚* × *𝑏𝑆𝑖𝑧𝑒*  (6)

The hierarchical model aggregation reduces the number of shares exchanged across all participating clients, because secret shares are shared only with the elected committee members *m*, where m *≪* n. The total number of messages exchanged between clients is denoted as Total\_Msg\_Num and is calculated as follows:

*𝑇 𝑜𝑡𝑎𝑙*\_*𝑀𝑠𝑔*\_*𝑁𝑢𝑚* = *𝑇 𝑜𝑡𝑎𝑙*\_*𝐼𝑛𝑡𝑟𝑎*\_*𝑀𝑠𝑔*\_*𝑁𝑢𝑚*+

*𝐼𝑛𝑡𝑒𝑟*\_*𝐺𝑟𝑝*\_*𝑀𝑠𝑔*\_*𝑁𝑢𝑚*  (7)

The total number of messages exchanged is proportional to *𝑂*(*𝑘*2) + *𝑂*(*𝑚*2), is very few when compared with traditional MPC enabled FL method [O(n2) ]where k, m *≪* n.

**5. Experiment and results**

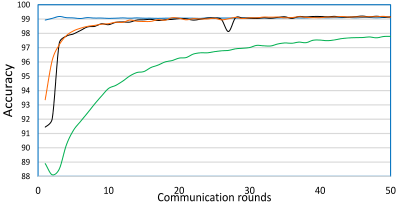
*5.1. Experimental setup*

To implement and evaluate the effectiveness of our proposed CE-Fed framework, we used PyTorch 1.2.0 and Python 3.74 as machine learn-ing library. We considered three public image datasets, MNIST [44], CIFAR-10 [45] and Fashion-MNIST dataset [41]. We used the CNN model that consists of two 5 × 5 convolution layers. The first layer has 32 channels and the second one has 64 channels. Each layer is followed with 2 × 2 max pooling, with ReLu activation, and a final softmax output layer. The model has about 600,000 trainable parameters. In our experiments, we study the performance and effectiveness in using IID (Independent and identically distributed) and non IID datasets. In the IID distribution, the data is shuffled and partitioned across all the clients, whereas in the non-IID distribution, first the data is sorted by the label and then partitioned across the clients in such a way that each party has fixed number of labels and there is no overlap between samples of different clients. To eliminate the randomness caused by client sampling, all clients are participated in each and every round of training. The learning rate is set to 0.01, local epoch number is set to 5 and batch size is set to 64. The stochastic gradient descent (SGD)is used as optimizer. We consider 4 groups and use *FedAvg* for model aggregation. We use Accuracy and communication cost as performance metrics in our experiments.

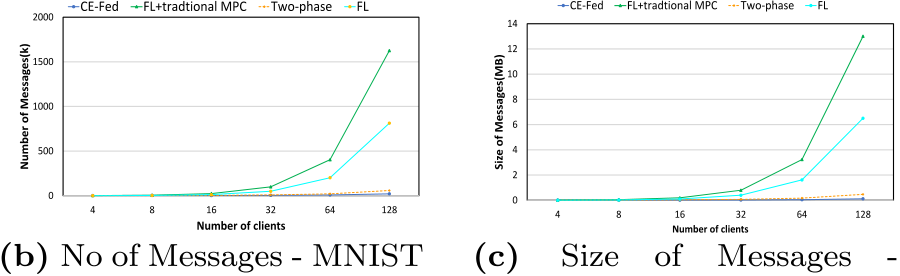
7

*R. Kanagavelu et al.*  *Array 15 (2022) 100207*









|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |



|  |  |  |
| --- | --- | --- |
|  |  |  |

**Fig. 9.** MNIST dataset.

|  |  |  |
| --- | --- | --- |
|  |  |  |



|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |



|  |  |  |
| --- | --- | --- |
|  |  |  |

**Fig. 10.** CIFAR-10 dataset.

rounds to converge when compared with MNIST dataset both in non-IID case, which indicates that non-IID has stronger data heterogeneity. There is a rapid increase in the number of shares exchanged by an average of 80%–90% with the increasing number of clients in the MPC enabled FL when compared with our proposed FL with committee selection as shown in Fig. 10(b), and Fig. 10(e). This shows that the traditional MPC enabled FL is not scalable when more clients joined the FL. It proves that our method is scalable to support more clients participated in FL. Our proposed CE-Fed reduces the communication cost by an average of 80%–90% when compared with Two-phase MPC enabled FL method [38].

CIFAR-10 and MNIST datasets have the similar observations when consider the size of messages exchanged between clients. It has been observed that our CE-Fed method reduces the communication cost significantly with increasing number of clients by an average of 80%–90% in both IID and non-IID distributions as shown in Fig. 10(c) and Fig. 10(f). It proves that our proposed CE-Fed method is more efficient in incurring low communication overhead.

*5.2.3. Fashion-MNIST dataset*   
 The effectiveness of our proposed FL is studied on Fashion-MNIST dataset. For IID data distribution, we first shuffle the data and equally

8

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *R. Kanagavelu et al.* |  |  |  | *Array 15 (2022) 100207* |



|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |



|  |  |  |
| --- | --- | --- |
|  |  |  |

**Fig. 11.** Fashion-MNIST dataset.

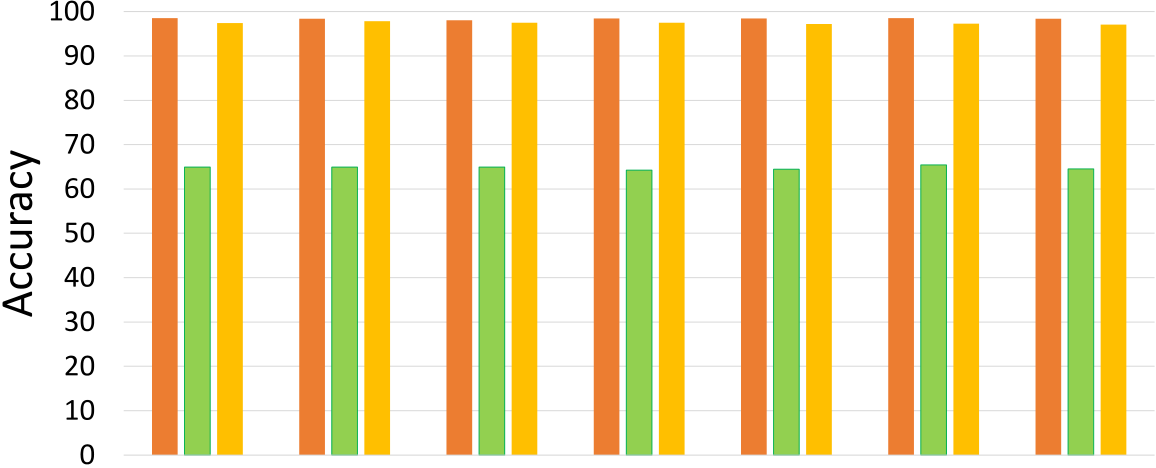
|  |  |  |
| --- | --- | --- |
|  |  |  |



|  |  |  |
| --- | --- | --- |
|  |  |  |

**Fig. 12.** Accuracy.





|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |



**Fig. 13.** Accuracy (Varying no of clients)- Comparison.

using local datasets of individual clients. Centralized training trains the model on centralized data. Experiment results shows that the test accuracy of federated learning is comparable with centralized learning and outperforms local training. It has been observed that FL training achieves an improvement in accuracy of 22%, 33%, 17% on MNIST, CIFAR-10 and Fashion-MNIST respectively when compared with local training. The key reason for the low accuracy in local training is that the size of the local dataset is not large enough.

Fig. 13 shows the accuracy score of proposed CE-Fed over MNIST, CIFAR-10 and Fashion MNIST datasets by varying the number of

9

*R. Kanagavelu et al.*  *Array 15 (2022) 100207*

the communication cost incurred in MPC protocols and also aggregates the models in a privacy-preserving manner without compromising the accuracy. The effectiveness of our proposed CE-Fed framework was demonstrated on various datasets. The above experiments indicates that our proposed CE-Fed is able to reduce the communication cost significantly while achieve the similar accuracy and privacy.

**CRediT authorship contribution statement**

**Renuga Kanagavelu:** Conception and design of study, Writing –original draft. **Qingsong Wei:** Conception and design of study, Writing– original draft. **Zengxiang Li:** Conception and design of study. **Haibin Zhang:** Revising the manuscript critically for important intellectual content. **Juniarto Samsudin:** Revising the manuscript critically for important intellectual content. **Yechao Yang:** Revising the manuscript critically for important intellectual content. **Rick Siow Mong Goh:** Conception and design of study. **Shangguang Wang:** Conception and design of study.

**Declaration of competing interest**

The authors declare that they have no known competing finan-cial interests or personal relationships that could have appeared to influence the work reported in this paper.

**Acknowledgements**

All authors approved the final version of the manuscript.

**References**

[1] [Ding C, Zhou A, Liu L, Ma X, Wang S. Resource-aware feature extraction in](http://refhub.elsevier.com/S2590-0056(22)00054-6/sb1)  [mobile edge computing. IEEE Trans Mob Comput 2020.](http://refhub.elsevier.com/S2590-0056(22)00054-6/sb1)

[2] [Lian Xiangru, Zhang Ce, Zhang Huan, Hsieh Cho-Jui,](http://refhub.elsevier.com/S2590-0056(22)00054-6/sb1) [Zhang Wei, Liu Ji. Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent In: NIPS. 2017.](http://refhub.elsevier.com/S2590-0056(22)00054-6/sb2)

[3] [Ren](http://refhub.elsevier.com/S2590-0056(22)00054-6/sb2)  [H,](http://refhub.elsevier.com/S2590-0056(22)00054-6/sb2)  [Li](http://refhub.elsevier.com/S2590-0056(22)00054-6/sb2)  [H,](http://refhub.elsevier.com/S2590-0056(22)00054-6/sb2)  [Dai](http://refhub.elsevier.com/S2590-0056(22)00054-6/sb2)  [Y,](http://refhub.elsevier.com/S2590-0056(22)00054-6/sb2)  [Kan](http://refhub.elsevier.com/S2590-0056(22)00054-6/sb2)  [Y,](http://refhub.elsevier.com/S2590-0056(22)00054-6/sb2)  [Lin](http://refhub.elsevier.com/S2590-0056(22)00054-6/sb2)  [X.](http://refhub.elsevier.com/S2590-0056(22)00054-6/sb2)  [Querying](http://refhub.elsevier.com/S2590-0056(22)00054-6/sb2)  [in](http://refhub.elsevier.com/S2590-0056(22)00054-6/sb2)  [inter](http://refhub.elsevier.com/S2590-0056(22)00054-6/sb2)[net](http://refhub.elsevier.com/S2590-0056(22)00054-6/sb3)  [of](http://refhub.elsevier.com/S2590-0056(22)00054-6/sb3)  [things](http://refhub.elsevier.com/S2590-0056(22)00054-6/sb3)  [with privacy preserving: Challenges, solutions and opportunities. IEEE Netw](http://refhub.elsevier.com/S2590-0056(22)00054-6/sb3)  [2018;32(6):144–51.](http://refhub.elsevier.com/S2590-0056(22)00054-6/sb3)

[4] [Garcia Y, Kassa M, Cuevas A, Cebrian M, Moro E, Rahwan I, Cuevas R. Analyzing gender inequality through large-scale facebook advertising data. Proc Natl Acad Sci 2018;115(27):69586963.](http://refhub.elsevier.com/S2590-0056(22)00054-6/sb4)

[5] [McMahan B, Ramage D. F](http://refhub.elsevier.com/S2590-0056(22)00054-6/sb4)ederated learning: Collaborative machine learning without centralized training data. 2017, [https://ai.googleblog.com/2017/04/ federated-learning-collaborative.html](https://ai.googleblog.com/2017/04/federated-learning-collaborative.html). Googl[e AI Blog.](https://ai.googleblog.com/2017/04/federated-learning-collaborative.html)

[Shay](https://ai.googleblog.com/2017/04/federated-learning-collaborative.html)an, Polsterl Sebastian, Navab Nassir, [6] [Roy](https://ai.googleblog.com/2017/04/federated-learning-collaborative.html)  [bhijit](https://ai.googleblog.com/2017/04/federated-learning-collaborative.html)  [Guha,](https://ai.googleblog.com/2017/04/federated-learning-collaborative.html)  [Siddiqui](https://ai.googleblog.com/2017/04/federated-learning-collaborative.html)   
 Wachinger Christian. BrainTorrent: A peer-to-peer environment for decentralized federated learning. 2019, <https://arxiv.org/pdf/1905.06731.pdf>.

[7] [Shokri R, Stronati M, Song](http://refhub.elsevier.com/S2590-0056(22)00054-6/sb7) [C, Shmatikov V. Membership inferenc](https://arxiv.org/pdf/1905.06731.pdf)[e attacks against machine learning models. In: Proceedings of IEEE symposium on security and privacy. 2017.](http://refhub.elsevier.com/S2590-0056(22)00054-6/sb7)

[8] [Nasr M, Shok](http://refhub.elsevier.com/S2590-0056(22)00054-6/sb7)[ri R, Houmansadr A. Comprehensive privacy analysis of deep learning: stand-alone and federated learning under passiveand active white-box inference attacks. In: Proceedings of IEEE ACM symposium on security and privacy. 2019.](http://refhub.elsevier.com/S2590-0056(22)00054-6/sb8)

[9] [Bonawitz K, Iv](http://refhub.elsevier.com/S2590-0056(22)00054-6/sb8)[anov V, Kreuter B, Marcedone A, McMahan HB, Patel S, Ramage D, Segal A, Seth K. Practical secure aggregation for privacypreserving machine learning. In: Proceedings of ACM computer and communications security. 2017.](http://refhub.elsevier.com/S2590-0056(22)00054-6/sb9) [10] [Alexandra W, Altman M, Bembenek A, Bun M, Gaboardi M, Honaker J, Nissim K,](http://refhub.elsevier.com/S2590-0056(22)00054-6/sb9) [OBrien R, Steinke T, Vadhan S. Differential privacy: A primer for a non-technical audience. Vanderbilt J Entertain Technol Law 2018;21(1):209–75.](http://refhub.elsevier.com/S2590-0056(22)00054-6/sb10)

[11] [Evans D, Kolesnikov V, Rosulek M. A pragmatic introduction to sec](http://refhub.elsevier.com/S2590-0056(22)00054-6/sb10)[ure multiparty](http://refhub.elsevier.com/S2590-0056(22)00054-6/sb11)  [computation. NOW Publishers; 2018.](http://refhub.elsevier.com/S2590-0056(22)00054-6/sb11)

[12] [Chillotti I, Georgieva M, Izabachne](http://refhub.elsevier.com/S2590-0056(22)00054-6/sb11) [M. Faster packed homomorphic operations and efficient circuit bootstrapping for TFHE. In: Advances in cryptology, Part I. Vol. 10624. Lecture notes in computer science, 2017, p. 377–408.](http://refhub.elsevier.com/S2590-0056(22)00054-6/sb12)

[13] [Phong LT, Aono Y, Hayashi T, Wang L, Moriai S. Privacy pre](http://refhub.elsevier.com/S2590-0056(22)00054-6/sb12)[serving deep learning via additively homomorphic encryption. IEEE Trans Inf Forensics Secur 2018;13(5):1333–45.](http://refhub.elsevier.com/S2590-0056(22)00054-6/sb13)

[14] [Ivan D, Ishai Y, Mi](http://refhub.elsevier.com/S2590-0056(22)00054-6/sb13)[kkel K. Perfectly secure multiparty computation and the computational overhead of cryptography. In: Proceedings of the 29th annual international conference on theory and applications of cryptographic techniques. 2010.](http://refhub.elsevier.com/S2590-0056(22)00054-6/sb14)

10

|  |  |  |  |
| --- | --- | --- | --- |
| *R. Kanagavelu et al.* | **Renuga Kanagavelu** received the Ph.D. degree in computer  science from Nanyang Technological University, Singapore,  in 2015. She is a scientist at the Institute of High-  Performance Computing, A\*STAR, Singapore. Her research  interests include Federated Learning, Privacy-preserving  techniques, Industrial IoT and Edge Analytics. |  | *Array 15 (2022) 100207* |
|  | **Juniarto Samsudin** is a research engineer at the Institute of  High-Performance Computing, A\*STAR, Singapore. He got  his master’s degree in Smart Product Design from Nanyang  Technological University, Singapore, 2007. His current  projects include federated learning, HPC, IoT and cloud  infrastructure and objectdetection using deep-learning. |
|  | **Qingsong Wei** received the Ph.D. degree in computer science from the University of Electronic Science and Tech-nologies of China, in 2004. He was with Tongji University as an assistant professor from 2004 to 2005. He is a Group Manager and senior research scientist at the Insti-tute of High-Performance Computing, A\*STAR, Singapore. His research interests include decentralized computing, fed-erated learning, high performance computing, emerging non-volatile memory and storage system. He is a senior member of the IEEE. |  | **Yechao Yang** received the bachelor’s degree in mechan-ical and electronic engineering from the University of Science and Technology of China, in 1998. He is a re-search engineer in Institute of High-Performance Computing (IHPC), A\*STAR, Singapore. His research interests include Blockchain, IoT, and Federated Learning. |
|  | **Zengxiang Li** is Executive Vice President of Digital Research Institute and Director of Collaborative Intelligence Lab, ENN Group, Beijing, China. His research group is focusing on Industrial IoT, Blockchain and Federate Learning research and development for trusted ecosystem across multiple in-dustry domains, including smart energy, healthcare, and city governance application domains. He completed his Ph.D. in Nanyang Technological University, Singapore in 2010. |  | **Rick Siow Mong Goh** is the Director of the Computing Science Department at the A\*STAR Institute of High-Performance Computing (IHPC). He received his Ph.D. degree in electrical and computer engineering from the National University of Singapore. At IHPC, he leads a team of more than 60 scientists in performing worldleading scientific research, developing technology to commercial-ization, and engaging and collaborating with industry. The research focus areas include artificial intelligence (AI), high performance computing (HPC), distributed computing, ma-chine & deep learning, complex systems, human–machine interaction, and modelling and simulation. |
| **Haibin Zhang** received the bachelor’s degrees of Computer Science and Animal Science from Shandong Agriculture University of China, in 2010. He was with NEC Corp and Alibaba Group as solution engineers to lead the team for researching and development from 2018 to 2020. He is a research engineer at the Institute of HighPerformance Com-puting, A\*STAR Singapore. His research interests include federated learning, blockchain, high-performance computing and distributed storage system. |  | **Shangguang Wang** is a Professor at the School of Com-puting, Beijing University of Posts and Telecommunications, China. He is a Vice-Director of the State Key Laboratory of Networking and Switching Technology. He has published more than 150 papers, and his research interests include service computing, cloud computing, and mobile edge com-puting. He served as General Chairs or TPC Chairs of IEEE EDGE 2020, IEEE CLOUD 2020, IEEE SAGC 2020, IEEE EDGE 2018, and IEEE ICFCE 2017, etc., and Vice-Chair of IEEE Technical Committee on Services Computing (2015–2018). He is currently serving as Executive Vice-Chair of IEEE Technical Committee on Services Computing (2021–), and Vice-Chair of IEEE Technical Committee. |
|  |

11