Performance and Communication Cost of Deep Neural Networks in Federated Learning Environments: An Empirical Study

Basmah K. Alotaibi[1,2], Fakhri Alam Khan[1,3,4*], Yousef Qawqzeh[5], Gwanggil Jeon[6], David Camacho[7]

[1] Information and Computer Science Department, King Fahd University of Petroleum and Minerals, Dhahran 31261 (Saudi Arabia)
[2] Department of Computer Science, College of Computer and Information Sciences, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh 13318 (Saudi Arabia)
[3] Interdisciplinary Research Centre for Intelligent Secure Systems, King Fahd University of Petroleum and Minerals, Dhahran (Saudi Arabia)
[4] SDAIA-KFUPM Joint Research Center for Artificial Intelligence, King Fahd University of Petroleum and Minerals, Dhahran (Saudi Arabia)
[5] College of Information Technology, Fujairah University (UAE)
[6] Department of Embedded Systems Engineering,Incheon National University, 119 Academy-ro, Yeonsu-gu, Incheon, 22012 (Korea)
[7] Department of Computer Systems Engineering Universidad Politécnica de Madrid Madrid (Spain)

* Corresponding author: fakhri.khan@kfupm.edu.sa

Received 14 May 2024 | Accepted 6 September 2024 | Early Access 16 December 2024

Abstract

Federated learning, a distributive cooperative learning approach, allows clients to train the model locally using their data and share the trained model with a central server. When developing a federated learning environment, a deep/machine learning model needs to be chosen. The choice of the learning model can impact the model performance and the communication cost since federated learning requires the model exchange between clients and a central server in several rounds. In this work, we provide an empirical study to investigate the impact of using three different neural networks (CNN, VGG, and ResNet) models in image classification tasks using two different datasets (Cifar-10 and Cifar-100) in a federated learning environment. We investigate the impact of using these models on the global model performance and communication cost under different data distribution that are IID data and non-IID data distribution. The obtained results indicate that using CNN and ResNet models provide a faster convergence than VGG model. Additionally, these models require less communication costs. In contrast, the VGG model necessitates the sharing of numerous bits over several rounds to achieve higher accuracy under the IID data settings. However, its accuracy level is lower under non-IID data distributions than the other models. Furthermore, using a light model like CNN provides comparable results to the deeper neural network models with less communication cost, even though it may require more communication rounds to achieve the target accuracy in both datasets. CNN model requires fewer bits to be shared during communication than other models.

Keywords

Communication Cost, Convolutional Neural Network (CNN), Deep Neural Networks, Distributive Learning, Federated Learning, Neural Network, Performance, Residual Neural Network (ResNet), Visual Geometry Group (VGG).

DOI: 10.9781/ijimai.2024.12.001

I. Introduction

The expansion of information and communication technology has increased the availability of data and computing resources, resulting in the Big Data era. This increasing data generated in the network requires efficient knowledge extraction and processing mechanisms to benefit from it. The data generated can be utilized as training data to provide the edge devices in the network with intelligence. However, traditional machine/deep approaches necessitate the collection of data to a central location to train the model and extract knowledge from it. Collecting the data to a central location can cause a significant transmission delay and raise privacy concerns due to sharing some private information through the network. Therefore, traditional machine learning approaches that require data collection in a central

Please cite this article as: B. K. Alotaibi, F. A. Khan, Y. Qawqzeh, G. Jeon, D. Camacho. Performance and Communication Cost of Deep Neural Networks in Federated Learning Environments: An Empirical Study, International Journal of Interactive Multimedia and Artificial Intelligence, vol. 9, no. 4, pp. 6-17, 2025, http://dx.doi.org/10.9781/ijimai.2024.12.001

location face various challenges, such as network communication and data privacy [1]. To tackle these issues, Federated Learning (FL) was introduced [2] to allow the model to be trained locally at the network edge and share the learned model rather than the data. FL is a distributive collaborative learning process that allows devices (known as clients) to train the model locally using their local data and share the trained model with a central server. The central server aggregates the received local models to create the global model. The learning process in federated learning is performed in rounds where, in each round, the central server provides the global model to the clients to train the model locally using their data [2].

Federated learning technology offers numerous advantages over traditional learning methods. It provides for more effective usage of network bandwidth and protects data privacy, as raw data is not demanded to be transferred to the server. Moreover, federated learning can employ the computing resources and diverse datasets on clients' devices to enhance the quality of the global model [3], [4], [5]. With these benefits, federated learning can be applied in various areas such as healthcare, transportation, IoTs, and mobile applications (such as next-word prediction) [6], [7].

However, federated learning faces various challenges because of its decentralized approach such as quality of the training data, the distributed architecture, the type of devices used to train the model, and the communication and aggregation mechanisms used; which affect the learning process in FL [8]. The clients' data in FL is non-Independent and Identically Distributed (non-IID) data due to varying device usage and location, as each client's data is dependent on their device usage and location. This means that the assumption of the IID data used in machine learning algorithms cannot be applied in FL. Therefore, FL encounters the additional challenge of data heterogeneity [4], [9]. Furthermore, during FL training, the clients and the central server exchange the local and global models in multiple rounds. However, this communication process can be a bottleneck due to the network's limited resources [10], [11]. The devices in FL vary in their computational power, storage, and network connectivity, and this heterogeneity could unbalance the training time and affect global model training [12], [13], [14].

Despite federated learning's potential, most research has focused on overcoming challenges like communication efficiency, data heterogeneity, and privacy preservation. However, an overlooked aspect of FL is the influence of various deep learning models on the overall performance and efficiency of FL. Understanding the impact of different neural network architectures within FL is crucial for optimizing model performance and communication resource efficiency.

In contrast to the traditional centralized learning approach, FL perform the training in distributed manner on the clients' devices and shares the local models with the central server through the network for several rounds. However, sharing these local models can be costly when using a deep neural network with many parameters. For that, this study aims to investigate the impact of using different neural networks on the model performance and communication cost, focusing on image classification tasks. Specifically, the research intends to evaluate and analyze the performance of a Convolutional Neural Network (CNN) model along with two complex variations of it, namely Visual Geometry Group (VGG) and Residual Neural Network (ResNet), which are widely used by the researchers when evaluating their proposed work in image classification tasks [15], [16], [17], [18], [19]. These three models are mainly used with Cifar-10 and Cifar-100 datasets [20]. In this study, we aim to address the following research question: Do we need a deeper network in a federated learning environment? by studying the performance of VGG-11, ResNet-18, and comparing them with a lighter CNN model, that is the same model used in the study that proposed the FL approach [2]. We aim to gain a deeper understanding of the benefits and drawbacks of using these models in FL environment.

The main contributions of this study are as follows:

1. Conducting an empirical study that investigates the impact of utilizing three different neural networks (CNN, ResNet-18, and VGG) for image classification task in a FL environment using two widely recognized datasets (Cifar-10 and Cifar-100).
2. We performed a comparative experiment and analyzed the performance of these three models under different data distribution settings (IID and non-IID), providing valuable insights into their behavior in different FL settings.
3. We studied and analyzed these models' performance and associated communication costs with different batch sizes and epoch values.
4. We provide insights into the trade-offs between model accuracy and communication efficiency.
5. Our findings shed light on the suitability of each neural network for FL, enabling researchers and practitioners to make informed decisions when selecting a learning model for their work.
6. To best of our knowledge, this study is the first to explore how the choice of different neural network models impacts the performance of federated learning.

The remainder of this work is organized as follows: Section II presents the literature review and highlight the datasets and local models in image classification tasks in federated learning. Section III show the system model. In section IV, we show the neural networks design, and the experiment settings. Section V shows the experiments results, while section VI shows the discussion. The conclusion and future work are provided in section VII.

II. Literature Review

Numerous studies have attempted to tackle the FL challenges using various techniques. For instance, Zhong *et al.* [21] uses a hierarchical clustering algorithm to overcome the non-IID data challenge by clustering the clients based on their model similarity and merge similar clusters. While Wu *et al.* [22] addresses communication cost and non-IID data challenges by using a threshold value to determine the importance of the local model to be uploaded to the server or skipped. Other studies [23], [24], [25] focus on non-IID data challenges by aiming to reduce client drift using different techniques such as decoupling and correction the local drift, rescaling the gradients, primal-dual variable that can adapt to data heterogeneity. The studies in [15], [16], [17] address communication challenges by reducing the number of bits exchanges using different compression techniques, such as Quantization and Count Sketch. The work in [18] improves the communication efficiency of FL by parallelizing the communication with computation to cover the communication phase with the training phase. Asynchronous technique is to overcome the communication bottleneck in FL [26], and another work uses partial synchronous technique to accelerate the training process in FL over the two-tier network using relay nodes to aggregate the model partially and reduce the communication rounds required [19]. The work of Li *et al.* aims to tackle the diversity in the computational capacity between devices and avoid waiting for slow devices by approximating the optimal gradients with a complete local training model using the Hessian estimation method to achieve the approximation, based on the heterogeneous local updates that has been received [27]. Another work uses a tier approach to overcome the latency caused by slow clients, by grouping clients into the same tier based on their response time to overcome the system heterogeneity [28]. Zeng *et al.* [29] proposes an energy-efficient bandwidth allocation and client selection scheme. The work

aims to reduce energy consumption while maximizing the selection of clients participating by adapting to both channel states and device computation capability when selecting clients and allocating the channel for them. Jebreel *et al.* [30] propose a mechanism to overcome the label-flapping attack in the federated learning environment to overcome malicious clients flipping their labels to poison the global model. Their work clusters clients based on their gradient parameters and analyzes the clusters to filter any potential threat. A novel backdoor attack in FL is introduced by Zhang *et al.* [31]. Their approach enables the attacker to optimize the backdoor trigger using adversarial training to enhance its persistence within the global training dynamics. In their work, they study the performance of existing techniques to overcome their attack and show their limitations.

These studies addressing FL challenges commonly use image classification tasks as their application when evaluating their proposed methods with different learning models. According to [20], [32], the most widely used datasets to test the model performance in FL are image datasets, and image classification tasks being the most commonly employed applications in FL. Furthermore, the study in [4] indicates that image datasets are the most used in FL.

CNN learning model is used in the study that propose FL [2] to evaluate its performance with MNIST [33] dataset. Many studies [21], [23], [26] use the same learning model when evaluating their proposed work, on different datasets such as CIFAR-10, CIFAR-100, and MNIST. Others have opted for a deeper neural network, such as VGG and ResNet, mainly when using Cifar-10 and Cifar-100 as their training dataset [15], [16], [17].

Table I shows various learning models and the datasets used to evaluate the performance in the FL environment. The table highlights that deeper network such as VGG and ResNet used different datasets such as Cifar-10 and Cifar-100, which include colored images, unlike the MNIST datasets that are gray-colored images and widely used with simpler learning models. However, some research also uses Cifar-10 and Cifar-100 with a simpler model, such as CNN. The table indicates that CNN, ResNet, and VGG are commonly used with Cifar-10 and Cifar-100 datasets. Therefore, this study aims to investigate the performance of the three aforementioned learning models on the Cifar-10 and Cifar-100 datasets in a federated learning environment.

TABLE I. Learning Models and Datasets Used at Different Studies in Federated Learning Environment

Model	Dataset	Ref.
CNN	Cifar10	[21], [23], [25], [18], [26], [28]
	Cifar-100	[23]
	MNIST	[21], [23], [25], [28], [29], [30]
	Fashion MNIST	[25], [26], [28]
ResNet	Cifar-10	[22], [15], [16], [17], [18], [19], [27], [30], [31]
	Cifar-100	[15], [17], [18], [26], [27]
	FEMNIST	[17], [31]
	Tiny-ImageNet	[23], [31]
VGG	Cifar-10	[22], [15], [18], [24]
	Cifar-100	[22], [24]
MLP	MNIST	[16], [18]
	Fashion MNIST	[27]
Logistic regression	MNIST	[19]
	Cifar-10	[19]

Despite the extensive research addressing various challenges in federated learning, the impact of different learning models on federated learning performance has not been thoroughly investigated. Table I shows that some studies utilized more than learning model, however these models were utilized with different datasets. As shown in the Table II, the studies did not compare the impact of different

TABLE II. Summary of the Literature Studies

Ref.	Focus	Methodology	No. of Models for Same Dataset	Comparison between models	Hyper-parameter Tuning (Epoch-Batch)
[21]	Non-IID	Hierarchical clustering algorithm	1	***X***	***X***
[22]	Communication cost and Non-IID	Select model update	2	***X***	***X***
[23]	Non-IID	Decoupling and correcting local drift	1	***X***	***X***
[24]	Non-IID	Rescaling the gradient	1	***X***	***X***
[25]	Non-IID	Primal-dual variable to adapt to data heterogeneity	1	***X***	***X***
[15]	Communication cost	Compression	2	***X***	***X***
[16]	Communication cost	Compression	1	***X***	***X***
[17]	Communication cost	Compression	1	***X***	***X***
[18]	Communication efficiency	Parallelizing communication with computation		***X***	***X***
[26]	Communication bottleneck	Asynchronous technique	1	***X***	***X***
[19]	Accelerating training process	Partial synchronous technique using relay nodes to aggregate the model partially		***X***	***X***
[27]	Computational capacity	Approximating the optimal gradients with a complete local training model	1	***X***	***X***
[28]	Latency	Tier approach	1	***X***	***X***
[29]	Energy	Select client based on device computation capability and channel states	1	***X***	***X***
[30]	Security	Cluster clients based on gradients parameter and filter any potential threat	1	***X***	***X***
[31]	Security	Optimize attack trigger through an adversarial adaptation loss	1	***X***	***X***
Ours	Impact of Learning Models in Federated Learning	Evaluation of various deep learning models	3	✓	✓

deep learning models on the performance of federated learning. Even when multiple models are utilized, they are often used with different datasets, which makes direct comparisons difficult. Additionally, these studies did not thoroughly investigate the effects of hyper-parameter tuning, such as varying epochs and batch sizes, on model performance and communication efficiency. Selecting the learning model is essential to federated learning, as it impacts both model performance and communication cost. Therefore, an evaluation that compares multiple neural networks on the same datasets while considering hyper-parameter variations is important and is the focus of this study. Our study fills these gaps by evaluating the performance of various deep learning models within a Federated Learning framework, providing a unique contribution to the existing body of knowledge.

III. System Model

In this section, we will provide a detailed explanation of the system model used in our study. This includes the principles and mechanisms of FL, the utilized aggregation algorithm, and the network architecture that we used.

A. Federated Learning

FL is a distributed collaborative learning process that was proposed by Google researchers in 2016 [2]. It is different from distributed (on-site) learning in that, in the latter, the central server provides the clients with an initial or pre-trained model, which the clients use to train their personalized models using their data. In this type of learning approach, there is no sharing of data or information [8], [34], [35].

In FL, there is a fixed set of Clients C, where each client c has its own datasets d_c, and at each round a fraction R of the clients C is selected to participate in this round to train the model [2]. Fig. 1 illustrates the FL architecture, where the central server sends the initial model to the participating clients. These clients then use this model to train a local model using their dataset. Afterward, the clients send the trained models to the server. The server then aggregates all the received local models using an aggregation mechanism. The process will be repeated for several rounds until a target is reached [2]. Typically, FL aims to minimize the objective function shown in (1):

$$\min_{\omega} F(\omega), \quad where \quad F(\omega) := \sum_{c=1}^{C} p_c F_c(\omega) \tag{1}$$

Where C is the total number of clients, $p_c \geq 0$ and $\sum_c p_c = 1$, the p_c term define the impact of each client on the global model, where there are two natural settings existing which are: $p_c = \frac{1}{d}$, *or* $p_c = \frac{d_c}{d}$, where d represents the total data sample of all clients and d_c represents the data sample for client c, and F_c is the local objective function of client c [2], [7].

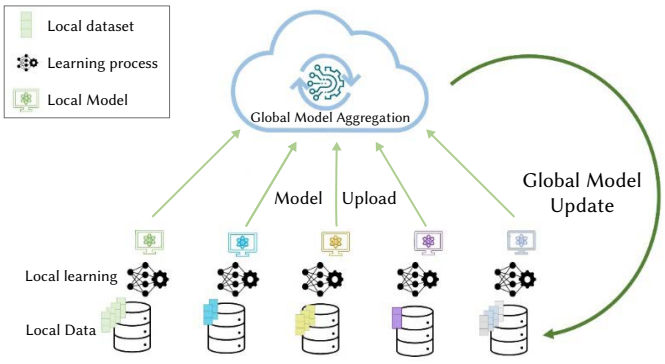


Fig. 1. The federated learning architecture.

In FL, the central server plays a vital role in providing an initial model, receiving updated local models from participating clients, aggregating these received local models, and subsequently disseminating a new global model to the participating clients. The most commonly used aggregation scheme in this type of learning is called Federated Averaging (FedAvg). FedAvg involves averaging the local stochastic gradient descent (SGD) updates. This method is usually implemented in a few general steps as follows [2], [7], [8]:

1. The server sets up the initial global model.
2. The server selects the participating clients (R, C), and sends the global model to them.
3. The clients that receive the global model train the received model using their local dataset. The most used technique is using SGD to compute the update.
4. The clients train the model for some epochs and upload the trained local model to the server.
5. The server then aggregates all the received local models using an averaging aggregation mechanism based on the clients' dataset size to create a new global model.
6. The steps from 2 to 5 are repeated for several rounds until a predefined target is reached.

Algorithm 1 (FedAvg):

The C Clients are indexed by $\boldsymbol{c}$.

$\boldsymbol{\eta}$: learning rate.

$\boldsymbol{B}$: Batch size.

$\boldsymbol{E}$: Local epochs.

Server executes:

Initialize the initial model ω_0

for each round t = 1, 2, 3, … do

$m \leftarrow max\,(R.C, 1)$

$C_t \leftarrow$ (*Select random set of m clients*)

for every client $c \in C_t$ in parallel do

$\omega_{t+1}^{c} \leftarrow ClientUpdate\,(c, \omega_t)$ …

$m_t \leftarrow \sum_{c \in C_t} n_c$

$\omega_{t+1} \leftarrow \sum_{c \in C_t} \frac{n_c}{m_t}\, \omega_{t+1}^{c}$

ClientUpdate(c, ω): // run on client c

$\beta \leftarrow$ *Spilt client dataset into batches of size B*

for each local epoch i from 1 to E do

for batch $b \in \beta$ do

$\omega \leftarrow \omega - \eta \nabla l\,(\omega; b)$

return ω to the server

Algorithm 1 illustrates the process of the FedAvg algorithm where the (Server executes) section shows the steps that the server performs. In contrast, the (CleintUpdate) section illustrates the process that are performed at each client. The clients train a local model using a deep/machine learning approach and send the locally trained model to the server.

B. Network Architecture

The network design used in this study for FL is a centralized architecture. In this architecture, there is a central server S and a set of clients C, where each client c has its own local dataset d_c. The central server is responsible for initializing the global model and selecting the participating clients ($R.C$) from the clients set C. For this work, the central server selects the participating clients randomly.

In this network, each participating client c_i receives the global model from the central server and trains it locally using its own dataset d_c for a predefined local epoch. After the completion of training, the local model is shared with the central server for global aggregation. The central server applies the FedAvg aggregation scheme to aggregate the local models received from all participating clients. Then, the process of selecting clients and providing the global model is repeated. The process will continue for several communication rounds. The FL architecture used in this work is shown in Fig. 1.

IV. Methods

This section presents the local learning model implemented by the federated learning clients and the experimental settings. It covers the local model architecture and the specific experimental parameters employed.

A. Local Model Design

In this subsection, we will discuss three deep learning models, namely CNN, ResNet, and VGG, that are utilized in a federated learning environment for image classification tasks. We will provide a general overview of their concepts and architecture in this subsection.

1. Convolutional Neural Network

CNN is a deep learning approach that can be utilized for tasks such as speech recognition and computer vision [36], [37]. CNN typically comprises three primary types of layers: convolutional, pooling, and fully connected layers. The convolutional layer is used for extracting features from the data. The pooling layer, on the other hand, reduces the size of the output from the convolutional layer and combines similar features to avoid redundancy. Finally, the fully connected layer establishes connections between the previous layer's output and the subsequent layer's input [38], [39], [40]. There are different well-known CNN architectures such as LeNet, AlexNet, GoogleNet, ResNet, VGG, which differ in terms of the used layers, the number of layers, the activation function used, and other factors [41], [38], [37]. The CNN model architecture used in this study is adopted from [2] and includes two convolutional layers, each followed by a max pooling layer, a fully connected layer, with ReLU activation function, and final SoftMax output layer.

2. Visual Geometry Group

The VGG model is a deep convolution model developed by the Visual Geometry Group [42]. To enhance the learning process, VGG uses a small convolution filter (3x3), which increases the depth of the network [42], [43]. The VGG has three (3x3) convolutional layers, which are equivalent to having a single (7x7) convolutional layer. However, VGG uses three (3x3) convolutional layers to reduce the number of parameters as it contains more ReLU layers (one after each convolution layer), which makes the decision function more discriminatory [42], [44]. VGG has different variations depending on the number of layers used (VGG-11, VGG-13, VGG-16, and so on). The VGG-11 model consists of two stacks of convolution layers and a Max pool layer, followed by three stacks of two convolution layers and a Max pool layer, followed by three fully connected layers, resulting in having eight convolution layers and three fully connected layers [42], [45].

3. Residual Neural Network

ResNet is a deep neural network that was proposed by He *et al.* in 2015 for image detection [46]. In ResNet, the input of the layer is added to the output of the residual mapping, which can contain two or more layers. Fig. 2 shows that a shortcut connection is established between the input and output of the residual mapping along with an additional operation. This shortcut connection helps the network learn more effectively, thereby improving its performance. ResNet is commonly used for image classification and object detection [47]. ResNet has different variations depending on the number of layers used (ResNet-18, ResNet-34, and so on). ResNet-18 comprises 17 convolution layers and a fully connected layer. A batch normalization layer and activation function can follow each convolution layer. The first convolution layer is followed by a max pooling layer. The network also includes eight sets of two convolutional layers, then an average pooling layer, and finally a fully connected layer with SoftMax activation function. The residual map is applied between the output of the even-numbered stack of convolution layers and the output of the next stack. The residual function is shown in (2):

$$y = F(x) + x \tag{2}$$

Where y represents the output vector of the layer, $F(x)$ represents the residual mapping to be learned, and x represents the input vector.

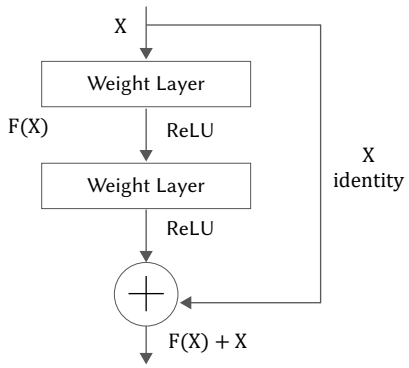


Fig. 2. Residual learning: a building block.

B. Experimental Setup

In this study, we evaluate the performance of the three learning models (CNN [2], ResNet-18 [46], and VGG-11 [42]), for image classification tasks using two datasets, Cifar-10 and Cifar-100 [48]. For ResNet-18, we replaced the batch normalization layers with group normalization as suggested and evaluated in [49]. The sizes of the three models transmitted by each client are presented in Table III. Cifar-10 dataset consists of 60,000 images that are categorized into 10 different classes. Each class has 6,000 images. Out of these 60,000 images, 50,000 are used for training and 10,000 for testing purposes. Similarly, the Cifar-100 dataset contains 60,000 images that have been classified into 100 distinct classes. Each class has 600 images. Out of these 60,000 images, 50,000 are used for training and 10,000 for testing purposes. In this study, we distributed the dataset among 100 clients in such a way that each client received 500 samples for training similar to [2], [22], [25]. We tried different settings and reported the best results obtained, we conducted the experiment over 300 rounds, with a learning rate of 0.01 and SGD as optimizer. At each round, we randomly selected 10 clients to participate as many studies used these settings [2], [25]. We performed the test on the client's side every five rounds. Following we show the experimental results of the three models, that highlight the model training accuracy and communication costs to reach a predefined target. We designed two cases to study the models' performance based on the data distribution. The first is the IID data distribution, and the second is the non-IID data distribution, following a similar setting as in [2]. In the IID data case, each client has data from all classes, where each client holds 50/5 samples from each class from the Cifar-10/Cifar-100 datasets. In the non-IID data setting, each client has data from a few classes, 2/20 classes, where each client holds 250/25 samples from the selected class from the Cifar-10/Cifar-100 datasets.

TABLE III. Model Size of the Three Models Transmitted by Each Client

Dataset	CNN	ResNet-18	VGG-11
Cifar-10	8.22MB	41.61MB	104.87MB
Cifar-100	8.4MB	41.87MB	107.25MB

V. Results

In this section, we present the experimental results of the three models with different numbers of epochs and batch sizes using two different datasets as we increase the computation per client. We compare the testing accuracy of the three models, communication bits exchanges and number of rounds to reach a predefined target accuracy.

A. Performance Comparison on Testing Accuracy

In this subsection, we show the performance of the three deep learning models using two cases when the clients have: (1) IID data and (2) non-IID data. Each case is evaluated using two different datasets, Cifar-10 and Cifar-100, and we varied the batch sizes and epoch values for each dataset.

1. IID Data Setting

To evaluate the performance of the three models, we tested them by varying the number of epochs and batch sizes. Fig. 3 illustrates the performance of the three models in the Cifar-10 and Cifar-100 dataset under the IID settings with an increase in batch size and local epoch value. In Cifar-10 the Fig. 3 (a) demonstrates that increasing the number of local epochs and decreasing batch sizes improves the model's performance for CNN model. Similar results were obtained in ResNet-18 model and VGG-11 model under the same settings, as shown in Fig. 3 (b) and Fig. 3 (c). The ResNet model converge faster with the increase of the local epoch value as shown in Fig. 3 (b), both batch sizes perform comparably well when trained with the same epochs value, indicating that the epoch count plays a crucial role in enhancing model performance. The VGG model in Fig. 3 (c) show a slow start especially with the 1-epoch configurations, however it obtains a higher accuracy with the increase of the global rounds with the 5-epoch configurations. In CNN and VGG-11 models, the best performance is achieved when the epoch size is 5, and the batch size is 16. While ResNet-18 performs best when the batch size =16 in all epoch values. We compared the performance of the three models at epoch=5 and batch size=16, 32, as shown in Fig. 3 (d). The VGG-11 model started slowly, but its performance improved with increasing rounds compared to ResNet-18 and CNN models. ResNet-18 and CNN provide comparable performance, as shown in Fig. 3 (d).

In Cifar-100 datasets, the performance of the three models under the IID settings is shown in Fig. 3. The models were evaluated with different numbers of epochs and batch sizes. Fig. 3 (e) shows the performance of the CNN model. The results indicate that the CNN shows better performance with a batch size of 16 in this setting. ResNet-18 and VGG-11 have similar performance, and they perform better with a batch size of 16 for different epoch sizes, as demonstrated in Fig. 3 (f) and Fig. 3 (g). The ResNet model converge faster with the increase of the local epoch value as shown in Fig. 3 (f), with the smaller batch size 16 outperforms the larger batch size 32. Also, the VGG model converge faster with the increase of the local epoch value as shown in Fig. 3 (g), with obtaining higher accuracy with the 5-epoch and 10-epoch configurations compared to the 1-epoch configurations for all batch sizes. We compared the performance of the three models at epoch=5 and batch size=16, 32, which is illustrated in Fig. 3 (h). The results indicate that VGG-11 performs worse compared to ResNet-18 and CNN models. Among these three models, CNN performs the best under the specified settings, as illustrated in Fig. 3 (h).

2. Non-IID Data Setting

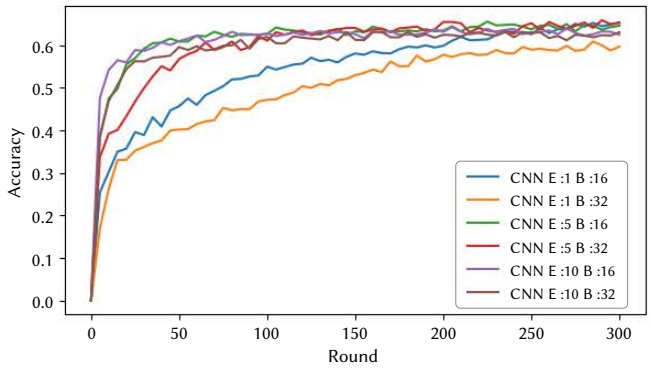
In the case of non-IID data settings, Fig. 4 illustrates the performance of the three models when tested using the Cifar-10 dataset and Cifar-100. In Cifar-10 the performance of all models improves as the number of local epochs increases and the batch size decreases as shown in Fig. 4 (a), Fig. 4 (b), and Fig. 4 (c). For all three model we observe that configurations with more epochs per round lead to higher accuracy, regardless of batch size. The best performance was achieved when the number of local epochs was equal to 5 for all models. However, the VGG-11 model performed the worst in non-IID settings compared to the CNN and ResNet-18 models, as illustrated in Fig. 4 (d). In Cifar-100 the performance of all models improves as the number of local epochs increases and the batch size decreases as shown in Fig. 4 (e), Fig. 4 (f), and Fig. 4 (g). For all three model we observe that configurations with more epochs per round lead to higher accuracy, regardless of batch size, with a slight edge for the smaller batch size 16. The best performance was achieved when the number of local epochs was set to 5 for CNN and VGG-11 models. However, in non-IID settings, CNN and ResNet with batch size = 16 perform better than the VGG-11 model that need more rounds to converge, as illustrated in Fig. 4 (h).

B. Performance Comparison on Communication Cost

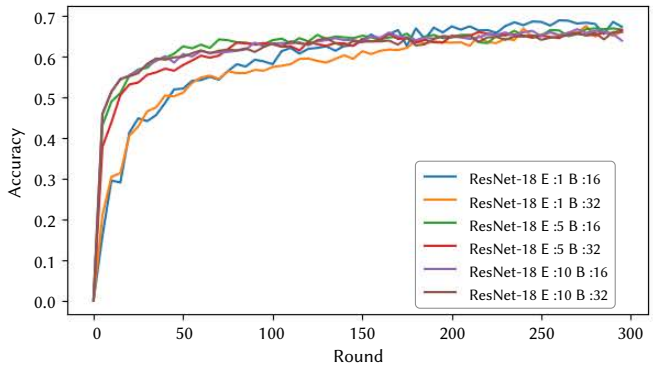
Communication cost is an important metric to evaluate FL, as the training process in FL is known to be a distributed process between clients, and requires clients to share their local models with a central server in multiple rounds through the network. For that, the number of rounds needed to reach a target accuracy and the number of bits sent by the clients are both an essential metric in FL. In this subsection, we evaluate the performance of the three learning models to achieve a predefined target accuracy in terms of the number of communication rounds (RoA@XX) and training bits exchanged from clients through the network. Table IV and Table V show the results for Cifar-10 and Cifar-100 datasets under the IID data settings, respectively. The "-" symbol means the target accuracy could not be obtained within the given number of communication rounds. However, the number of bits uploaded from clients during training is still reported.

Fig. 5 (a) illustrates the communication costs associated with different learning models for the CIFAR-10 and CIFAR-100 datasets. The figure shows that the CNN model consistently has the lowest communication cost across various configuration settings. In contrast, the VGG-11 model demonstrates the highest communication cost under all configurations. Notably, ResNet-18 falls between CNN and VGG-11 in terms of communication cost. Although ResNet-18 requires fewer communication rounds to achieve the target accuracy as shown in Table IV, and Table V, these rounds are more costly compared to those of the CNN model. This indicates that the CNN model can achieve the target accuracy with significantly lower communication overhead, making it a more efficient choice for federated learning scenarios.

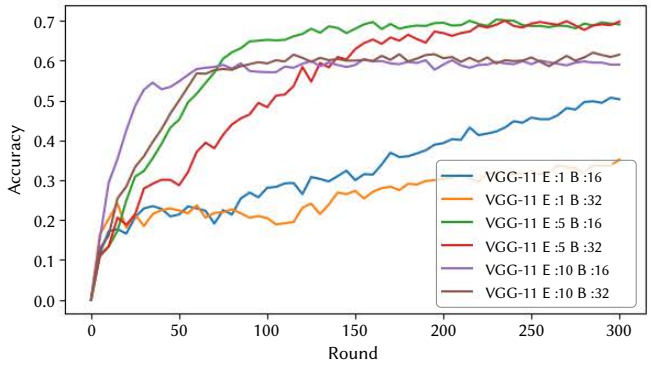
To sum up, the three models perform better when the data is IID data. In the Cifar-10 and Cifar-100 datasets, ResNet-18 shows faster convergence to reach the target accuracy compared to CNN and VGG-11 in most cases. However, since FL is a distributed learning process that shares a locally trained model instead of raw data to preserve privacy and provide an efficient communication, it is essential to consider the difference in the model weight of these three models. Although ResNet-18 requires fewer rounds, the number of bits transmitted is more than that of the CNN model as shown in Fig. 5.



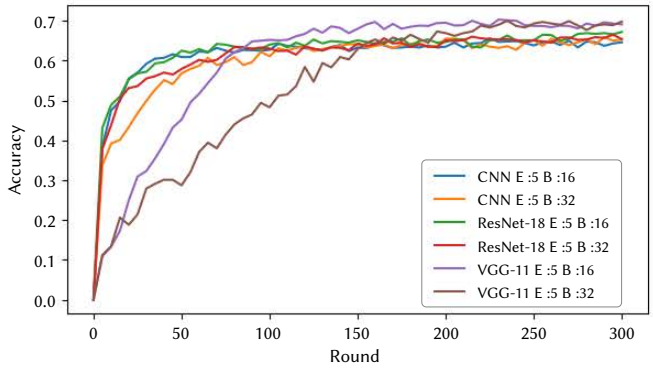
(a) CNN model for IID settings in Cifar-10



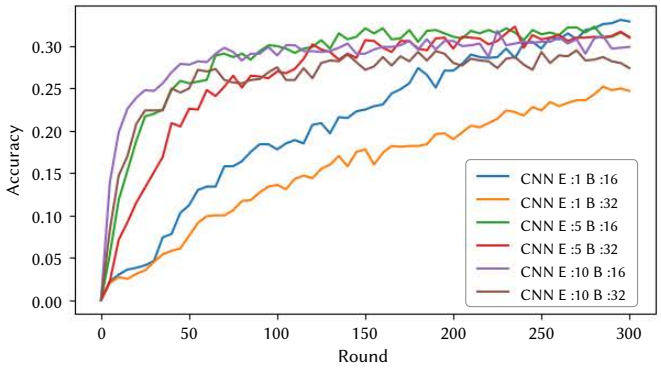
(b) ResNet-18 model for IID settings in Cifar-10



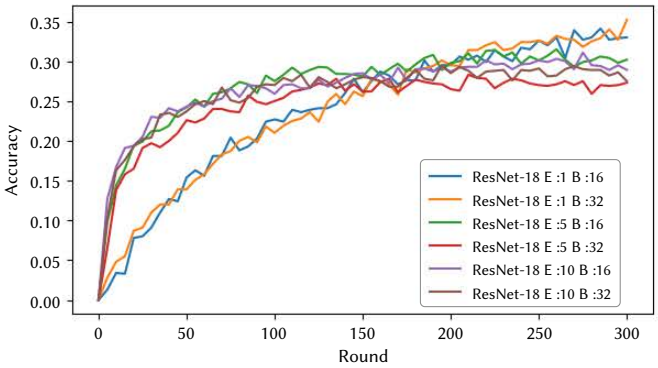
(c) VGG-II model for IID settings in Cifar-10



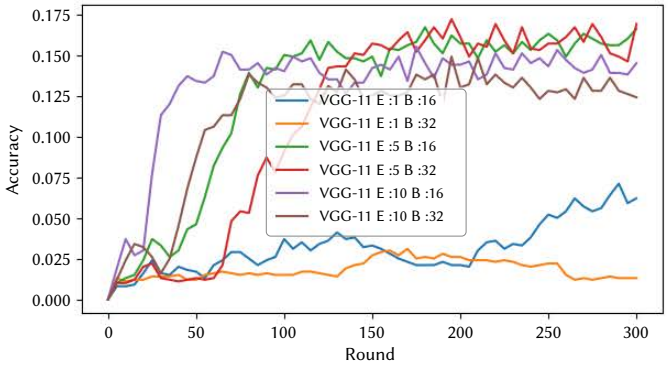
(d) The three models for IID settings in Cifar-10 with E=5



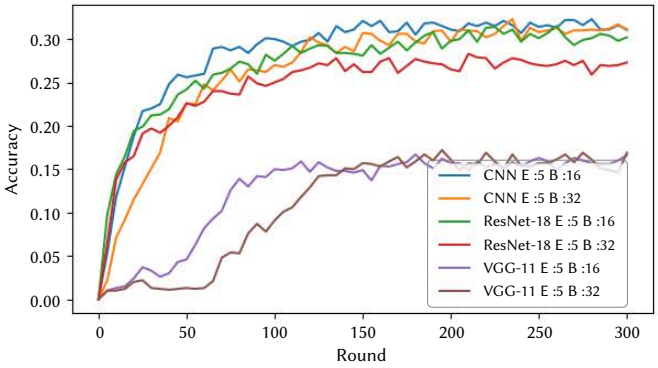
(e) CNN model for IID settings in Cifar-100



(f) ResNet-18 model for IID settings in Cifar-100

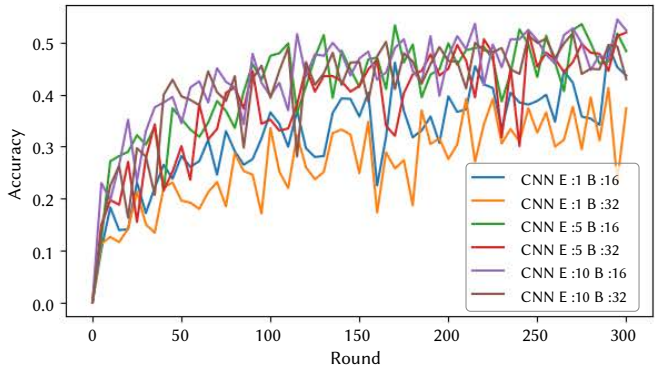


(g) VGG-II model for IID settings in Cifar-100

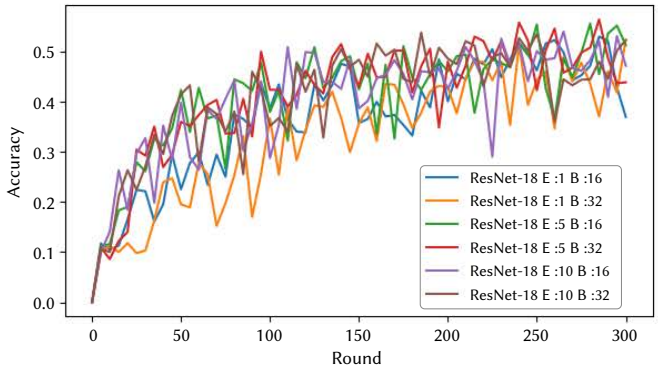


(h) The three models for IID settings in Cifar-100 with E=5

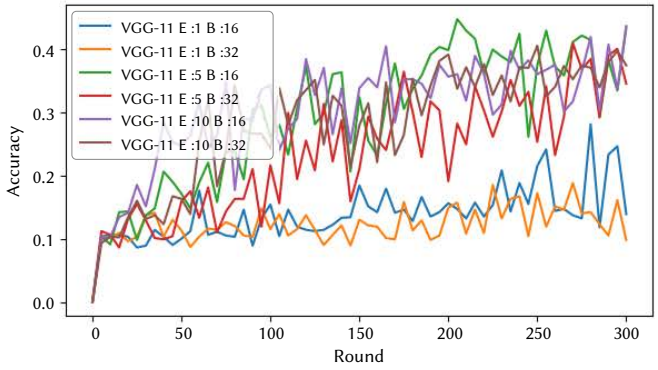
Fig. 3. The test accuracy of the three models for IID setting in Cifar-10 and Cifar-100 dataset.



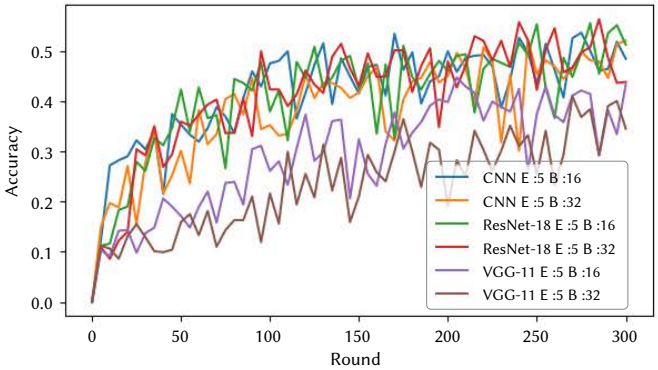
(a) CNN model for non-IID settings in Cifar-10



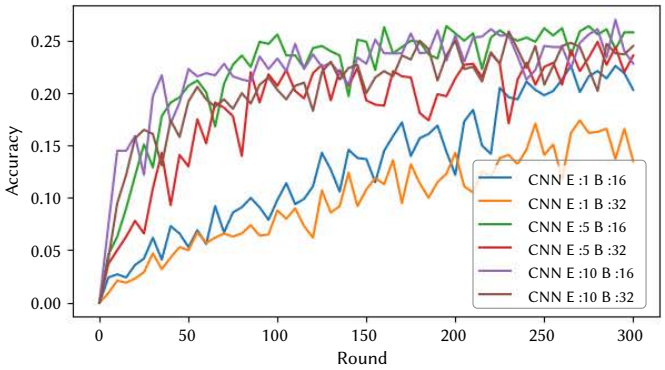
(b) ResNet-18 model for non-IID settings in Cifar-10



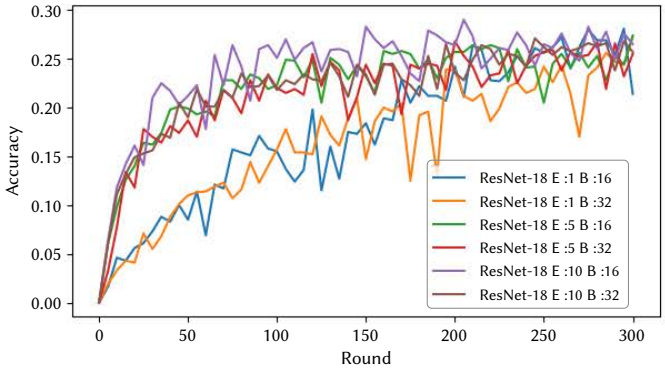
(c) VGG-II model for non-IID settings in Cifar-10



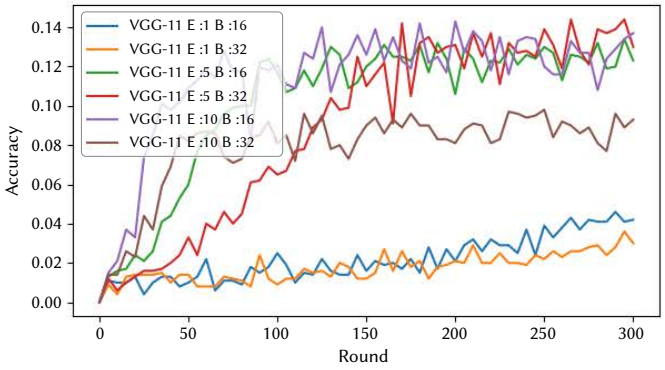
(d) The three models for non-IID settings in Cifar-10 with



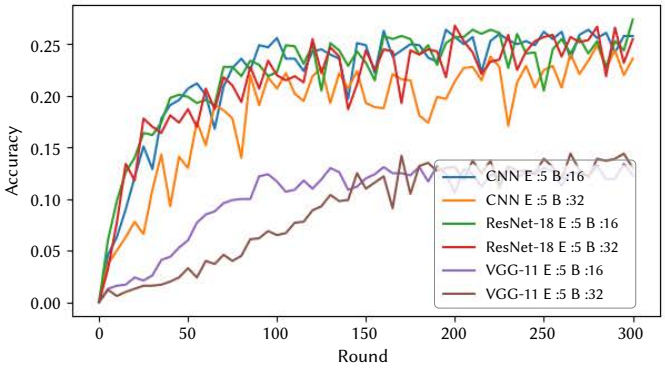
(e) CNN model for non-IID settings in Cifar-100



(f) ResNet-18 model for non-IID settings in Cifar-100



(g) VGG-II model for non-IID settings in Cifar-100



(h) The three models for non-IID settings in Cifar-100 with E=5

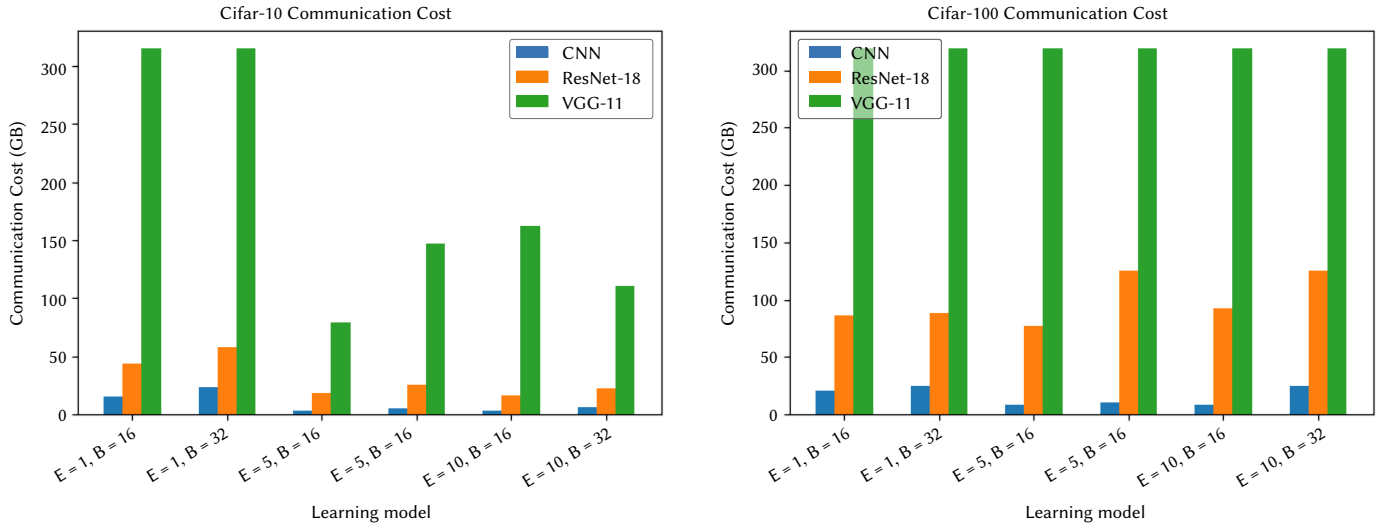
Fig. 4. The test accuracy of the three models for non-IID setting in Cifar-10 and Cifar-100 dataset.

TABLE IV. Communication Rounds and Training Bits Exchanges to Reach a Target Accuracy for IID Data Settings in Cifar-10

Model	**CNN**						**ResNet-18**						**VGG-11**					
Epoch	**1**		**5**		**10**		**1**		**5**		**10**		**1**		**5**		**10**	
Batch	**16**	**32**	**16**	**32**	**16**	**32**	**16**	**32**	**16**	**32**	**16**	**32**	**16**	**32**	**16**	**32**	**16**	**32**
RoA@60	**190**	**285**	**35**	**65**	**40**	**80**	**105**	**140**	**45**	**60**	**40**	**55**	**-**	**-**	**75**	**140**	**155**	**105**
Communication Cost (GB)	**15.63**	**23.44**	**2.87**	**5.34**	**3.29**	**6.58**	**43.70**	**58.27**	**18.73**	**24.97**	**16.65**	**22.89**	**314.62**	**314.62**	**78.65**	**146.82**	**162.55**	**110.11**

TABLE V. Communication Rounds and Training Bits Exchanges to Reach a Target Accuracy for IID Data Settings in Cifar-100

Model	**CNN**						**ResNet-18**						**VGG-11**					
Epoch	**1**		**5**		**10**		**1**		**5**		**10**		**1**		**5**		**10**	
Batch	**16**	**32**	**16**	**32**	**16**	**32**	**16**	**32**	**16**	**32**	**16**	**32**	**16**	**32**	**16**	**32**	**16**	**32**
RoA@30	**240**	**-**	**95**	**120**	**105**	**-**	**205**	**210**	**185**	**-**	**220**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
Communication Cost (GB)	**20.16**	**25.2**	**7.98**	**10.08**	**8.82**	**25.2**	**85.84**	**87.93**	**77.46**	**125.6**	**92.12**	**125.6**	**318.7**	**318.7**	**318.7**	**318.7**	**318.7**	**318.7**



(a) Cifar-10 (b) Cifar-100

Fig. 5. The communication cost for the different models using Cifar-10/Cifar-100 dataset to reach target accuracy (=60%, =30%).

VI. Discussion

In this study, we have assessed the performance of three different models, CNN, ResNet-18, and VGG-11, in image classification tasks under two different settings. Our findings suggest that VGG-11 has a slower start and necessitates more communication rounds to obtain the same accuracy as CNN and ResNet-18. Although VGG-11 eventually reaches a similar performance level to CNN and ResNet-18, it requires more communication rounds, and these rounds are costly as the VGG-11 model size is higher than CNN and ResNet-18. Moreover, VGG-11 did not perform well in the case of non-IID data setting. On the other hand, ResNet-18 performs well and converge quickly, requiring fewer communication rounds than CNN in some cases. However, it is noteworthy that ResNet-18 communication rounds cost more than CNN communication rounds, as the ResNet-18 model requires sending more bits when uploading the model.

The CNN model used in this study is a lighter model compared with ResNet-18 and VGG-11; however, it provides comparable performance compared to the other two models in terms of accuracy obtained in the predefined number of rounds. In some cases, more communication rounds may be required to obtain the same accuracy as ResNet-18; however, these rounds are less costly than ResNet-18 rounds since CNN requires fewer bits for exchanging the model compared to ResNet-18 and VGG-11. When training a model, obtaining high accuracy is necessary and is considered an essential evaluation criterion. However, as FL is a decentralized approach that requires sharing clients' model with the central server, the communication cost is also considered a vital evaluation metric. Therefore, we must consider the number of rounds and the client bits sent to reach these results. When we use the communication costs as an evaluation metric to evaluate the performance of the three models, CNN provides better results than ResNet-18 and VGG-11. The CNN model exchanges fewer bits than ResNet-18 and VGG-11, providing comparable accuracy.

Based on our analysis of the performance of the three models and their associated communication costs, we recommend using a lighter model (such as CNN in [2]) in FL, since federated learning is a learning process that requires sharing locally trained models with a central server for several rounds through the network. Moreover, it is essential to determine the local epoch value since the devices in FL are limited in resources. Setting the local epoch to a high value does not indicate enhancing the model performance in these settings. Therefore, we recommend using an adaptive local epoch that can start with a higher value and decrease after a certain point to avoid overfitting and enhance the global model performance.

Hence, to answer the research question, Do we need a deeper network in a federated learning environment? Our answer is that in FL, it is necessary to consider different factors before choosing the

local model since the clients' devices are limited in resources and data, and the learning process is performed in rounds. We may not need a deeper neural network model since we need to consider not only model accuracy but also communication cost, and it may cost more to share a deeper model during training.

VII. Conclusion and Future Work

Federated learning is a collaborative learning approach where the clients and server exchange training models for several rounds through the network to reach a predefined target. The choice of the learning model affects the model performance and the communication costs. Researchers have been faced with the decision of which learning model to choose when using FL for image classification tasks; several studies chose a deeper neural network model, such as VGG and ResNet, to evaluate their proposed work, while others chose a light model, such as CNN. In this study, we aimed to answer the question, "Do we need a deeper network in a federated learning environment?" Since FL is a decentralized approach, the model weight must be considered along with the model performance when choosing a neural network model since the model will be shared during training through the network. To answer this question, we conducted an empirical study investigating the impact of using three different neural networks in a FL environment (CNN, VGG-11, and ResNet-18). Our study evaluates the three models under different data settings (IID and non-IID) using two datasets (Cifar-10 and Cifar-100). We showed the performance of these models with varying numbers of local epochs and batch sizes. The results indicate that using CNN provides comparable results compared to the other models, with less communication cost; however, in some cases, it may require more rounds to reach the predefined target, but the communication cost (GB) is less than the other two models, making it a more practical choice for FL applications where communication efficiency is critical. We observed significant performance degradation for all models under non-IID settings compared to IID settings, highlighting the importance of addressing data heterogeneity in FL. Furthermore, our analysis revealed that using a 5-epoch configuration with a batch size 16 resulted in the best performance across all three models compared to other configurations. Our study provided valuable insights into the trade-offs between model accuracy and communication efficiency, suggesting that CNN offers a balanced approach by maintaining high performance while minimizing communication costs. Our findings indicate that training a model using a CNN model requires fewer network resources to train the FL model to reach accuracy similar to that obtained using deeper models.

For the future research direction, we aim there is a need to analyze the performance of FL on client device energy consumption and computational resources using different models and investigate if they are applicable on the client devices that are limited in resources. Also, investigate the effect of applying different compression method with deep neural networks. Furthermore, investigating the effects of introducing an adaptive local epoch size. By initially setting a higher epoch size that ultimately decreases after a certain point, to enhance the model's accuracy while simultaneously reducing costs.

VIII. Funding Declaration

1: The authors would like to acknowledge the support received from the Saudi Data and AI Authority (SDAIA) and King Fahd University of Petroleum and Minerals (KFUPM) under SDAIA-KFUPM Joint Research Center for Artificial Intelligence Grant no. JRC-AI-RFP-12.

2: This work has been partially supported by the project PCI2022-134990-2 (MARTINI) of the CHISTERA IV Cofund 2021 program; by MCIN/AEI/10.13039/501100011033/ and European Union NextGenerationEU/PRTR for XAI-Disinfodemics (PLEC 2021-007681) grant, by European Comission under IBERIFIER Plus - Iberian Digital Media Observatory (DIGITAL-2023-DEPLOY- 04-EDMO-HUBS 101158511); and by EMIF managed by the Calouste Gulbenkian Foundation, in the project MuseAI.

References

[1] Z. Yang, M. Chen, K.-K. Wong, H. V. Poor and S. Cui, "Federated learning for 6G: Applications, challenges, and opportunities," *Engineering*, vol. 8, pp. 33-41, 2022.

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, Fort Lauderdale, FL, USA, 2017.

[3] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li and H. V. Poor, "Federated learning for internet of things: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 23, pp. 1622-1658, 2021.

[4] X. Ma, J. Zhu, Z. Lin, S. Chen and Y. Qin, "A state-of-the-art survey on solving non-IID data in Federated Learning," *Future Generation Computer Systems*, vol. 135, pp. 244-258, 2022.

[5] C. Carrascosa, F. Enguix, M. Rebollo and J. Rincon, "Consensus-based learning for MAS: definition, implementation and integration in IVEs," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 8, pp. 21-32, 2023.

[6] M. Aledhari, R. Razzak, R. M. Parizi and F. Saeed, "Federated learning: A survey on enabling technologies, protocols, and applications," *IEEE Access*, vol. 8, pp. 140699-140725, 2020.

[7] T. Li, A. K. Sahu, A. Talwalkar and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE signal processing magazine*, vol. 37, pp. 50-60, 2020.

[8] S. AbdulRahman, H. Tout, H. Ould-Slimane, A. Mourad, C. Talhi and M. Guizani, "A survey on federated learning: The journey from centralized to distributed on-site learning and beyond," *IEEE Internet of Things Journal*, vol. 8, pp. 5476-5497, 2020.

[9] C. Briggs, Z. Fan and P. Andras, "A review of privacy-preserving federated learning for the Internet-of-Things," *Federated Learning Systems: Towards Next-Generation AI*, pp. 21-50, 2021.

[10] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie and R. Pedarsani, "Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization," in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, Online, 2020.

[11] A. Khan, M. ten Thij and A. Wilbik, "Communication-Efficient Vertical Federated Learning," *Algorithms*, vol. 15, p. 273, 2022.

[12] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li and Y. Gao, "A survey on federated learning," *Knowledge-Based Systems*, vol. 216, p. 106775, 2021.

[13] B. Yu, W. Mao, Y. Lv, C. Zhang and Y. Xie, "A survey on federated learning in data mining," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 12, p. e1443, 2022.

[14] L. Li, Y. Fan, M. Tse and K.-Y. Lin, "A review of applications in federated learning," *Computers & Industrial Engineering*, vol. 149, p. 106854, 2020.

[15] Z. Lian, J. Cao, Y. Zuo, W. Liu and Z. Zhu, "AGQFL: Communication-efficient Federated Learning via Automatic Gradient Quantization in Edge Heterogeneous Systems," in *2021 IEEE 39th International Conference on Computer Design (ICCD)*, Storrs, CT, USA, 2021.

[16] J. Xu, W. Du, Y. Jin, W. He and R. Cheng, "Ternary Compression for Communication-Efficient Federated Learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, p. 1162–1176, 2022.

[17] D. Rothchild, A. Panda, E. Ullah, N. Ivkin, I. Stoica, V. Braverman, J. Gonzalez and R. Arora, "Fetchsgd: Communication-efficient federated learning with sketching," in *Proceedings of the 37th International Conference on Machine Learning*, Virtual, 2020.

[18] Y. Zhou, Q. Ye and J. Lv, "Communication-efficient federated learning with compensated overlap-fedavg," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, pp. 192-205, 2021.

[19] Z. Qu, S. Guo, H. Wang, B. Ye, Y. Wang, A. Y. Zomaya and B. Tang,

"Partial Synchronization to Accelerate Federated Learning Over Relay-Assisted Edge Networks," *IEEE Transactions on Mobile Computing,* vol. 21, pp. 4502-4516, 2021.

[20] B. Alotaibi, F. A. Khan and S. Mahmood, "Communication Efficiency and Non-Independent and Identically Distributed Data Challenge in Federated Learning: A Systematic Mapping Study," *Applied Sciences,* vol. 14, p. 2720, 2024.

[21] J. Zhong, Y. Wu, W. Ma, S. Deng and H. Zhou, "Optimizing Multi-Objective Federated Learning on Non-IID Data with Improved NSGA-III and Hierarchical Clustering," *Symmetry,* vol. 14, p. 1070, 2022.

[22] X. Wu, X. Yao and C.-L. Wang, "FedSCR: Structure-based communication reduction for federated learning," *IEEE Transactions on Parallel and Distributed Systems,* vol. 32, pp. 1565-1577, 2020.

[23] L. Gao, H. Fu, L. Li, Y. Chen, M. Xu and C.-Z. Xu, "Feddc: Federated learning with non-iid data via local drift decoupling and correction," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, New Orleans, LA, USA, 2022.

[24] Z. Lian, W. Liu, J. Cao, Z. Zhu and X. Zhou, "FedNorm: An Efficient Federated Learning Framework with Dual Heterogeneity Coexistence on Edge Intelligence Systems," in *2022 IEEE 40th International Conference on Computer Design (ICCD)*, Olympic Valley, CA, USA , 2022.

[25] Y. Gong, Y. Li and N. M. Freris, "FedADMM: A robust federated deep learning framework with adaptivity to system heterogeneity," in *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, Kuala Lumpur, Malaysia, 2022.

[26] S. Zhou, Y. Huo, S. Bao, B. Landman and A. Gokhale, "FedACA: An Adaptive Communication-Efficient Asynchronous Framework for Federated Learning," in *2022 IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS)*, CA, USA , 2022.

[27] X. Li, Z. Qu, B. Tang and Z. Lu, "Fedlga: Toward system-heterogeneity of federated learning via local gradient approximation," *IEEE Transactions on Cybernetics,* vol. 54, pp. 401 - 414, 2023.

[28] Z. Chai, A. Ali, S. Zawad, S. Truex, A. Anwar, N. Baracaldo, Y. Zhou, H. Ludwig, F. Yan and Y. Cheng, "Tifl: A tier-based federated learning system," in *Proceedings of the 29th International Symposium on High-Performance Parallel and Distributed Computing*, Stockholm, 2020.

[29] Q. Zeng, Y. Du, K. Huang and K. K. Leung, "Energy-efficient radio resource allocation for federated edge learning," in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, Dublin, Ireland, 2020.

[30] N. M. Jebreel, J. Domingo-Ferrer, D. S´anchez and A. Blanco-Justicia, "LFighter: Defending against the label-flipping attack in federated learning," *Neural Networks,* vol. 170, pp. 111-126, 2024.

[31] H. Zhang, J. Jia, J. Chen, L. Lin and D. Wu, "A3fl: Adversarially adaptive backdoor attacks to federated learning," in *Advances in Neural Information Processing Systems*, New Orleans, LA, USA, 2024.

[32] S. K. Lo, Q. Lu, C. Wang, H.-Y. Paik and L. Zhu, "A systematic literature review on federated machine learning: From a software engineering perspective," *ACM Computing Surveys (CSUR),* vol. 54, pp. 1-39, 2021.

[33] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE,* vol. 86, pp. 2278-2324, 1998.

[34] B. Ghimire and D. B. Rawat, "Recent advances on federated learning for cybersecurity and cybersecurity for federated learning for internet of things," *IEEE Internet of Things Journal,* vol. 9, pp. 8229-8249, 2022.

[35] Z. Lu, H. Pan, Y. Dai, X. Si and Y. Zhang, "Federated Learning With Non-IID Data: A Survey," *IEEE Internet of Things Journal,* pp. 1-1, 2024.

[36] C. Janiesch, P. Zschech and K. Heinrich, "Machine learning and deep learning," *Electronic Markets,* vol. 31, pp. 685-695, 2021.

[37] A. Mathew, P. Amudha and S. Sivakumari, "Deep Learning Techniques: An Overview," *Advanced Machine Learning Technologies and Applications: Proceedings of AMLTA 2020,* pp. 599-608, 2021.

[38] Z. Li, F. Liu, W. Yang, S. Peng and J. Zhou, "A survey of convolutional neural networks: analysis, applications, and prospects," *IEEE transactions on neural networks and learning systems,* vol. 33, pp. 6999 - 7019, 2021.

[39] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning," *Nature,* vol. 521, pp. 436-444, 2015.

[40] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, L. Wang, G. Wang, J. Cai and T. Chen, "Recent advances in convolutional neural networks," *Pattern recognition,* vol. 77, pp. 354-377, 2018.

[41] N. K. Chauhan and K. Singh, "A review on conventional machine learning vs deep learning," in *2018 International conference on computing, power and communication technologies (GUCON)*, Greater Noida, India, 2018.

[42] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556,* 2014.

[43] A. S. Rao, T. Nguyen, M. Palaniswami and T. Ngo, "Vision-based automated crack detection using convolutional neural networks for condition assessment of infrastructure," *Structural Health Monitoring,* vol. 20, pp. 2124-2142, 2021.

[44] W. Wang, Y. Yang, X. Wang, W. Wang and J. Li, "Development of convolutional neural network and its application in image classification: a survey," *Optical Engineering,* vol. 58, pp. 040901-040901, 2019.

[45] M. Pak and S. Kim, "A review of deep learning in image recognition," in *2017 4th international conference on computer applications and information processing technology (CAIPT)*, Kuta Bali, Indonesia, 2017.

[46] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016.

[47] M. Shafiq and Z. Gu, "Deep Residual Learning for Image Recognition: A Survey," *Applied Sciences,* vol. 12, p. 8972, 2022.

[48] A. Krizhevsky, G. Hinton and others, "Learning multiple layers of features from tiny images," University of Tront, Toronto, ON, Canada, 2009.

[49] K. Hsieh, A. Phanishayee, O. Mutlu and P. Gibbons, "The Non-IID Data Quagmire of Decentralized Machine Learning," in *Proceedings of the 37th International Conference on Machine Learning,* Virtual, 2020.

Basmah Alotaibi

Basmah Alotaibi received the B.Sc. degree in Computer Science from Imam Muhammad ibn Saud Islamic University (IMSIU) and the M.Sc. degree from the Department of Computer Science, King Saud University, Riyadh, Saudi Arabia. She is currently a Ph.D. student in Information and Computer Science, at King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia. Her research interest includes Federated learning, IoT, fog computing, cloud computing, and network security.

Fakhri Alam Khan

Fakhri Alam Khan is currently serving as an Associate Professor with the Department of Information and Computer Science at King Fahd University of Petroleum and Minerals. He is also a 'Research Fellow' with the Saudi Data and AI Authority (SDAIA) under the SDAIA-KFUPM Joint Research Center for Artificial Intelligence. He received his PhD in computer science from the University of Vienna, Austria, in 2010 and completed a post-doctorate from the Vienna University of Technology in 2017. He has published several research articles in various reputed peer-reviewed internationally recognized journals and has supervised numerous M.S. and Ph.D. students. His research interests include the IoT, data analytics, data provenance, distributed systems, machine learning, multimedia technologies, and nature-inspired metaheuristic algorithms.

Yousef Qawqzeh

Yousef Qawqzeh received the PhD. degree in systems engineering from UKM University, Bangi, Kuala Lumpur, Malaysia, in 2011, where he is currently working as an associate professor in the college of information technology, Fujairah University. He is currently working on several projects in the fields of machine learning, data science, and bioinformatics. He has several publications in international journal and conferences. His research interest includes the early prediction of cardiovascular diseases using the photoplethysmography technique, the development of computer-aided diagnosis systems for early diagnosis of breast cancer using artificial intelligence and machine learning techniques, and the detection and prediction of high-risk diabetics using machine learning and artificial intelligence techniques.

Gwanggil Jeon

Gwanggil Jeon received the B.S., M.S., and Ph.D. (summa cum laude) degrees from the Department of Electronics and Computer Engineering, Hanyang University, Seoul, Korea, in 2003, 2005, and 2008, respectively. From 2009.09 to 2011.08, he was with the School of Information Technology and Engineering, University of Ottawa, Ottawa, ON, Canada, as a Post-Doctoral Fellow. From 2011.09 to 2012.02, he was with the Graduate School of Science and Technology, Niigata University, Niigata, Japan, as an Assistant Professor. From 2014.12 to 2015.02 and 2015.06 to 2015.07, he was a Visiting Scholar at Centre de Mathématiques et Leurs Applications (CMLA), École Normale Supérieure Paris-Saclay (ENS-Cachan), France. From 2019 to 2020, he was a Prestigious Visiting Professor at Dipartimento di Informatica, Università degli Studi di Milano Statale, Italy. From 2019 to 2020 and 2023 to 2024, he was a Visiting Professor at Faculdade de Ciência da Computação, Universidade Federal de Uberlândia, Brasil. He is currently a professor at Incheon National University, Incheon. He was a general chair of IEEE SITIS 2023, and served as a workshop chairs in numerous conferences. Dr. Jeon is an Associate Editor of IEEE Transactions on Circuits and Systems for Video Technology (TCSVT), Elsevier Sustainable Cities and Society, IEEE Access, Springer Real-Time Image Processing, Journal of System Architecture, and Wiley Expert Systems. Dr. Jeon was a recipient of the IEEE Chester Sall Award in 2007, ACM's Distinguished Speaker in 2022, the ETRI Journal Paper Award in 2008, and Industry-Academic Merit Award by Ministry of SMEs and Startups of Korea Minister in 2020.

David Camacho

David Camacho is full professor at Computer Systems Engineering Department of Universidad Politécnica de Madrid (UPM), and the head of the Applied Intelligence and Data Analysis research group (AIDA: https://aida.etsisi.uam.es) at UPM. He holds a Ph.D. in Computer Science from Universidad Carlos III de Madrid in 2001 with honors (best thesis award in Computer Science). He has published more than 300 journals, books, and conference papers. His research interests include Machine Learning (Clustering/Deep Learning), Computational Intelligence (Evolutionary Computation, Swarm Intelligence), Social Network Analysis, Fake News and Disinformation Analysis. He has participated/led more than 50 research projects (Spanish and European: H2020, DG Justice, ISFP, and Erasmus+), related to the design and application of artificial intelligence methods for data mining and optimization for problems emerging in industrial scenarios, aeronautics, aerospace engineering, cybercrime/cyber intelligence, social networks applications, or video games among others. He serves as Editor in Chief of Wiley's Expert Systems from 2023, and sits on the Editorial Board of several journals including Information Fusion, IEEE Transactions on Emerging Topics in Computational Intelligence (IEEE TETCI), Human-centric Computing and Information Sciences (HCIS), and Cognitive Computation among others.