Improving the Effectiveness of Bitwidth-Aware Federated Learning in Wireless Networks

Sarah H. Mnkash
University of Technology
Baghdad, Iraq
cs.22.13@grad.uotechnology.edu.iq

Faiz A. Al Alawy
University of Technology
Baghdad, Iraq

Israa T. Ali
University of Technology
Baghdad, Iraq

Abstract*— This research explores the potential of model quantization in enhancing the efficiency of federated learning (FL) in the domains of wireless communication and computing. The process involves collecting the quantized local FL model parameters obtained from edge devices and combining them to create a global-quantized server. Afterwards, the devices are synchronised using the suggested bitwidth FL method. We need to collaboratively decide on the devices that will be included in a FL training iteration, as well as the specific quantization bitwidth set to be utilised for compressing the local model. In this optimisation issue, we are provided with a budget for device samples and a limit on the end-to-end latency each iteration for quantized federated learning (FL). The objective is to minimise the training loss. To address the said problem, it is essential to possess a comprehensive comprehension of how quantization impacts the general efficiency of machine learning. Additionally, the capability to conduct server-side inference that can accurately predict the functioning of this procedure is crucial. In order to achieve this objective, we carry out an extensive examination of the effectiveness of the suggested Federated Learning (FL) method under the conditions of limited communication and quantization errors occurring in wireless connections. This work validates our idea by presenting quantitative findings that demonstrate the degree to which the training loss in FL is enhanced in each cycle, depending on the choice of device, quantization technique, and many characteristics particular to the model. Next, we presented a proposal base of reinforcement learning (RL that is utilised to choose actions in a sequential manner. We demonstrate that the FL process of training may be seen as a specific case of a Markov decision process. This is the manner in which we confront the second obstacle. This is a training approach for FL that differs from unconstrained by a specific model RL. Unlike framework RL, this model-based learning method seeks to accurately imitate the behaviour of agents utilising Belief MDP, which is a mathematical characterization.

Keywords*— *Machine Learning, Performance Optimization, Nodes, WSN

I. INTRODUCTION

Federated learning (FL) is an advanced approach to distributed computing that enables several devices to collaboratively train on a global machine learning model while mitigating the risk of data leakage [1]. Federated Learning (FL) begins by first locally optimising the model's parameters on the device. These optimised parameters are then transferred to a central hub, such as a base station, where they are aggregated worldwide. This central hub oversees the operations of all mobile phones. This procedure is repeated several times until the model reaches convergence and achieves a high degree of accuracy [2]. The performance of federated learning is influenced by local training and the processes for communication between devices and servers, with varying degrees of impact. In a resource-constrained edge environment, the importance of these aspects is much greater. This is particularly true when devices have varying communication and processing resources. For example, a drone with high-power capabilities may be used to take measurements, contrasted to a relatively low-cost sensor. Recent studies have suggested a novel approach to test machine learning quantization using end-devices. This approach aims to reduce latencies caused by local training, not only for weight transfer but also for other purposes. Citations [3]. This significantly reduces the burden on the device's resources, since it carries out all communication and training operations directly on quantized learning models. Although our study has discovered positive results, it has also identified various obstacles that must be addressed in order to effectively implement quantized federated learning (FL) over wireless networks. These challenges primarily stem from the complexities associated with specifying the bitwidth for quantization and the impact of training statistics on the performance of the model in FL.

II. RELATED WORKS

The use of quantized FL over wireless networks - Quantization and Aggregation has raised several fundamental [4]. In [5], the authors jointly learn an FL model which is transmitted with low quantization error using a general vector quantizer. To accelerate the convergence rate, a diverse quantization approach was proposed in [6] to uploading FL based models. To deal with the tight restrictions about training latency and device transmission abilities, a robust FL strategy has been proposed to alleviate possible situations where transmissions go inactive or quantization error occur [7]. To obtain similar learning results with less communication overhead, a hierarchical gradient quantization method was proposed by the authors in [8] under federated learning framework. An efficient computation FL tool was observed using a communication-efficient technique in [9] by the authors use gradient quantization to reduce the number of training rounds and transmission bits. In [10], the impact of quantized communications on decentralised learning frameworks and their efficiency were extensively studied. Consequently, [11] accounted for the worst case wherein 1-bit quantized local gradients must be transmitted to train a global FL model all while keeping communication costs minimal. The authors of [12] consider the trade-off between accuracy and quantized energy efficiency. FL system littlemenet completed wireless networks. In [13], to optimize the number of communication bits used along all FL iterations for minimizing the amount of distributed energy consumed in communications, an adaptive quantized gradient strategy was proposed. To lower error of local FL model update compression as much as possible, the best quantizer was discovered in [14]. While such overhead might be small, in light of a device-to-device based wireless network for FL models [14] suggested reducing the data sent to fit an element-wise quantization algorithm. The authors of [15] proposed an optimizing losing function, pricing of

broadcast quantized FL Wireless resources and methods available to minimize training time, transmission cost. Each of these prior investigations into quantized FL depend on a base pre-training understanding for the values essential smoothness and gradient diversity parameters. Given these assumptions, a widely recognized way to find the optimal FL training strategy is by designing an appropriate quantization error rate and linking it with standard optimisation techniques that are able to measure how this correlates with FL performance. Cb Hence is a strong and liberal additional condition to the conventional optimisation techniques, which may confirm solution when they converge on a local minimum that would not be desirable for FL in practice due no availability of these model parameters until the end of training. To face this, a promising method is to contact reinforcement learning (RL) [16]. This is turn will allow the server to learn a stronger fitting loss function by gradually learning these parameters through interaction with devices during training. Recently, RL algorithms in [17] attempted to fine-tuned method parameters for best FL performance. The authors in [18] presented of method deep multi-agent RL to reduce the energy consumption of training and accelerate FL convergence. To minimize training latency and energy consumption, an RL-based device selection strategy was proposed by the authors in [19], where it aims to find out which combo of devices should be chosen in each round to participate on a training cycle. A framework based on deep RL was proposed in [20] to optimize the long-term performance of FL while taking into account energy and bandwidth constraints. In [21], researchers explored the usage of deep Q-network (DQN) to address device mobility-induced wireless communication disruptions in the FL framework. [22] utilised the of deep RL to optimise trainer duration and consumption energy at same time changing CPU-cycle frequency of devices. On the. Researched a quantization allocation mechanism using normal DQN to improve performance for FL. To reduce the drain on power wastage from FL framework energy supply, authors in [23] proposed a quantization based solution to tackle using multiagent RL. Reference [24] studied how computational complexity was correlated to global convergence in a quantized federated RL framework. To improve different features of the model training, previous research employed RL methods to learn how well FL's performance correlates with the strategy for learning. The best policy has to be found during the long tracking process of these approaches and it slows down FL convergence speed because more experimentation is done by server reconnecting with devices in an environment have many) observations from multiple training policies$ for this such method. In this paper, we are going to tackle these challenges by proposing RL-based solutions with mathematical foundations inspired from quantized FL training. To speed up the search of best FL policy, based on training data coordinating server is going to estimate required FL model parameters.

III. A Synopsis of the Approach and Its Results

This study proposes a novel fine-tuning framework for quantize FL in wireless networks. It uses RL method that can estimate the parameters with a close approximation of FL train also provide an analytical intuition on how FL learning works, without needing to engage with devices at every time step. As far as we know, this is the first study to systematically explore how bitwidth optimisation for quantisation may be integrated into a federated learning framework. What we do right?

- Our federated learning system enables the quantization of models to various bitwidths, facilitating the local training and broadcasting of federated learning models from distributed wireless devices to a coordinating server. During each cycle, the server chooses a suitable group of devices to carry out the Federated Learning (FL) algorithm using varying quantized bitwidths. To do this, we frame the job of simultaneously picking devices and FL models as an optimisation problem. This topic tries to minimise training loss for the target devices while also considering the data distribution across devices and the heterogeneity in communication and computation. It takes into account the service latency and bandwidth demands of each device to represent the variability in these factors.
- First of all, we start by analytically calculating the anticipated pace at which convergence is predicted to occur. During training of quantized FL framework considering data distribution (balance). Our detailed research shows that (1) choice the model intrinsic characteristic, quantization scheme and device selection strategies have large impacts on expected FP performance in terms of field loss over two consecutive iterations. To get a linear regression model for these features under training in an environment without (server-side observable) training data with which to estimate them (server side sensed limit positions). In view of these estimations, we argue that the FL instruction procedure may be accurately characterised as an MDP in which each step corresponds to a single global model loss variation from one epoch to another.

IV. Developing the System Model and Formulating the Problem

Imagine you are setting up a wireless network, where there are M (M < N) devices linked upstream in front of a coordinating server (without loss of generality). Please refer to the figure to understand how these devices are connected. Fig. 1 apply the Federated Learning technique to standardise the machine learning model literacy for the given array. Each device m has a training data sample n, where the size of MMIIX. The input feature vector Xm, where n belongs to the set of real numbers RNI×1, is accompanied by the target label Ym, where n belongs to the set of real numbers RNO×1, in the case of supervised learning. The goal of the server and devices is to decrease the frequency of incidents. The subsequent loss function encompasses all data samples.

$$F(g) = \min_{g=} \frac{1}{N} \sum_{M=1}^{M} : \sum_{n=1}^{Nm} : f\left(g, x_m, n, y_m, n\right) \qquad (1)$$

where g ∈ RY ×1 is a vector capturing the worldwide FL method of dimension Y trained across the N = Nm data samples in totally from all devices. where f (g, Xm, n, Ym, n) is losing function (e.g., squared error) can be assesses the degree of precision with which the constructed worldwide FL method g can relate between input vector Xm, n and output vector Ym, n. As shown Fig. 1, Published on (image) Title of the paper The Value of Communication in Federated Learning Low Bitwidth Deep Neural Network Training Publication Type Journal Year 2020 State-of-the-art approaches for training low bitwidth deep neural networks rely heavy. A Detailed Breakdown of the Components. Devices: Sketch of various devices (including smartphones

and IoT) in federated learning. This is where the personal data and inferences reside locally or are generated for local computation, also referred to as on-device processing during federated learning. Wireless Network: The tools are wirelessly linked to the main base station. This link allows the devices to send locally computed updates through a base station. Each line or arrow shows data flow between the devices and base station in wireless connections.

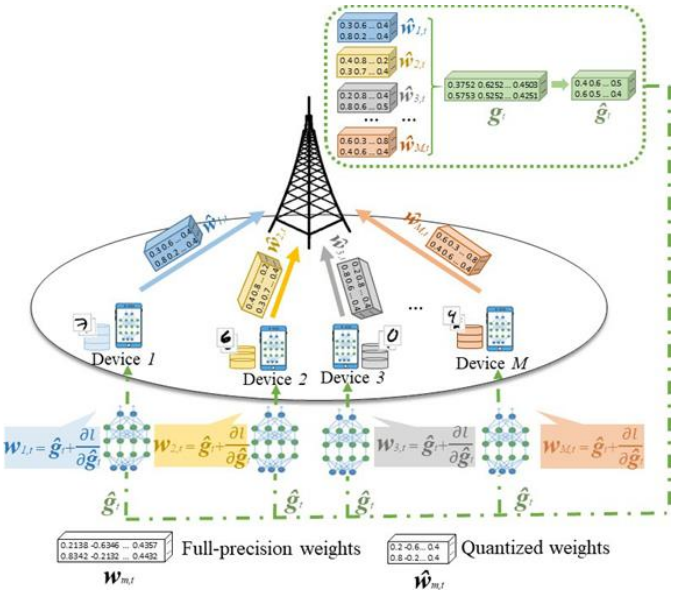


Fig. 1. Illustrates our proposed low bitwidth federated learning approach implemented across several devices and a single base station in a wireless network [23]

A. *Federated Learning Training with Low Bitwidth*

In the state of Florida, the process of locating a universal model that minimises the value of J in equation (1) is accomplished by repeatedly redistributing model parameters between devices and a server. Due to their limited wireless and computational capabilities, devices would be incapable of training and transmitting the large model parameters required for deep learning. In their study, the authors introduced the concept of bitwidth federated learning as a means to decrease the computational time required for calculations and transmission. The model parameters of the FL model for bitwidth FL are quantized, in contrast to federated averaging which has been extensively studied [25]. Instructions for training a neural network using bitwidth floating-point representation:

1) The system initiates the worldwide learning model then quantizes have it across all devices before to broadcasting for each individual device.

2) The train losing is computed by use the quantized worldwide learning method and data samples from each individual device.

3) The learning method of every gadget is regularly updated with quantized values, using the estimated training loss of that device.

4) Each device is discretized by the learning model.

5) The server chooses the devices to broadcast local federated learning models.

6) The FL models obtained locally are consolidated at a central location to provide a unified global FL model that is then sent to devices. This leads to the subsequent optimal vector J, which is precisely the cumulative outcome of stages 2 through 6 performed iteratively.

- Determine the training loss for each device

During training, bitwidth FL employs a quantized FL model for each device to determine its loss function and gradient vectors. Hence, the quantity of resources needed for training and transmitting Federated Learning (FL) models will be contingent upon the quantization in bitwidths. This methodology differs significantly from optimization-based federated learning (FL) methods [26], which use gradient relaxation of quantization during training to minimise loss in inference time, but with little improvement and additional computational complexity. Now, we will go into the training process using mathematics. Step 1 - Calculate the training loss for each device: To proceed to step two, we will now provide a technique for computing the training loss for each individual device. The experiments in this study primarily concentrate on training neural networks. However, it is worth mentioning that the methodology study It may be used in conjunction with other machine learning techniques, such as the Support Vector Machine (SVM) algorithm. [27], without any loss of generality.

$$h_N^{KT} + 1 = \sigma\left(g \wedge_K^{\cdot} \cdot h_M^{KT}\right)$$
$$\sigma : An\ activation\ function, which\ applies\ a\ non-linear\ transformation$$

GKT Weights or transformation function at iteration K and time T.

Hmkt; The previous state or output at iteration K and time T.

Global Learning (GL): The weights of the local FL model on each device will be αt bits. Here is where the FPN undergoes the process of transformation into a QNN, or Quantized Neural Network. Each QNN weight has two potential outcomes: The values -1/0 and +1 occur at αt = 1. A binary neural network (BNN) is a form of neural network model that only utilises two variables for weights [28].

$$L = -\sum_{1c}^{i} = 1 y_i \log\left(y \wedge_i\right)$$

The weights of the local federated learning model on each device are quantized into αt bits. The process of converting a fully connected neural network (FNN) into a quantized neural network (QNN) is as follows. If αt=1, the weight between neurons in the QNN is either -1, 0, or +1. Thus, a neural network that allocates weights to two variables values might be referred to as a binary synapse (BS) represented as [29]. To calculate the input hkm,t and weight ˆgkT(J, k,t) vectors for a group of neurons in layer k that fire at time t, we may use the above formulas. This implies that the result of each layer at iteration is, [30]

$$w_T + 1 = w_T - \eta \nabla L\left(w_T\right)$$

Federated Learning (FL) necessitates a complex process of model transmission and aggregation that includes a number of processes aimed to properly update the global model while ensuring privacy. The next section offers further information, including equations.

- Model Transmission

Local Model Update

Each device k runs a local training job on its dataset and updates his local model parameters W_K paren The local update can be illustrated as:-

$$W_K^{T+1} kt + 1 = wkt - \eta \nabla Lk(wkt)$$

Problem Formulation

Objective:

Minimize the federated learning (FL) training loss (*w*)L(w) while satisfying a delay requirement *D* Max D max for FL completion per iteration.

w: Model parameters

S: Set of selected devices

α_K : Quantization level for device k

d_K : Delay for device k

Optimising (12) is challenging when utilising conventional optimisation techniques because of the following factors. Prior to selecting a selection of devices as central controllers, gather They use their discretised regional federated learning (FL) models to construct the FL model in its entirety. Nevertheless, each distinct geographical area. The FL model produced by each device is determined by the unique characteristics of its local dataset. To choose the most suitable device, the server needs specific information about the datasets, such as the chosen approach for selecting and quantising data, which will limit the loss during federated learning training. Moreover, it is crucial to take into account the probabilistic character of the scenario. Every individual FL model is trained using the Gradient Descent approach, subject to certain conditions at each iteration. Nevertheless, the quantisation technique failed to accurately represent the training loss or device selection. The server uses traditional optimising techniques. The reason for this is the use of the stochastic gradient descent method, which allows each device to independently choose a random selection of data samples from its local dataset. The server is incapable of minimising each loss pair from a centralised perspective device, impeding the optimisation of local federated learning model training. In order to tackle these concerns, we suggest using a model-based reinforcement learning (RL) technique that maximises the performance of the chosen device and minimises the training loss in federated learning (FL). This enables the server to ascertain the relationship and quantisation scheme of the selected devices. This connection allows the server to precisely predict and forecast the values of ut and with. The training loss for Federated Learning is computed by finding the optimal value that minimises αt [31][32].

V. Optimization Methodology

An approach using Model Based Reinforcement Learning (RL) is used to optimise the scheme for selecting devices. The introduction of the quantization technique alpha is denoted by the symbol U. Traditional model-free reinforcement learning (RL) methods include the interaction of edge devices to learn how to pick and quantify the device. As compared to other schemes, the model based RL approaches allow server compute a mathematical model for FL training. Process and hence by learning the best device selection, quantization scheme on which this circuit performs optimal probability of state transition, a matrix. Then, the components of the proposed model are presented for the firstly introduced based RL method. In this case, we utilize a technique of linear regression to learn the dynamic environment model in RL approach. After this, we discuss implementing the model based RL. Discussion during this paper, we address the problem of joint space variation clustering when there are noise observations along with investigate a novel segmentation share chopping typically designed for identifying global optimal U and α. In conclusion, the convergence and complicity of the RL method proposed here is analysed [33][34].

A. The constituents of Model Based RL Method

This model-based reinforcement learning (RL) approach consists of six components: a) agent, b) action, c) transition function. Action states are described by their d) state transition probability, e) reward, and f) policy.

- Agent: The server is the agent to execute proposed model based RL algorithm. In more detail, a server that wishes to aggregate the local FL models of each device at every iteration must select only specific devices for transmission and threshholding (quantize) bit-wise all elements in their individual matrix representation of the FL model.
- Action- a server action at = [ut, αt] ∈ A such that ut gives the id of device index selected. List of non-overlapping task schemes u−→t and quantization approachesilαtyLof all units at iteration t. States: The state is st = F(gt) ∈ S that characterizes the global FL model at time t, where F denotes a function of gt to compute which has got captured all aspects of some training loss associated with the global model and set S indicates available states.
- State Transition Probability (P (st+1|st, at)): It is the probability to go from one state to another. To the probability p (st, s′): posting probabilities: likelihood of transitioning from state St to s′and, as action at occurs in t. Where P (st+1|st, at) is the state transition probability specifies a pattern for moving from one status to another while executing activity at. The details are as follows: St=F(gt)S F(gt): The FL training loss at iteration t; S: the set of available states. In conclusion, that amount P (St+1, St, a) t tells us how federated learning is probabilistic and requires decision making: It represents the probability of movement from one current state to some next step given actions under taken.
- Policy: Policy refers to the likelihood of selecting each possible action for a given condition of an agent. In this scenario, a reinforcement learning technique is used that utilises a deep neural network parametrized by θ to map the inputstate to the output action. We can now say the policy. Let πθ (St, at) denote the probability under of taking action a given state. Mean Field Federated Learning Optimization.

B. Calculation of the probability of transitioning between states

This section describes the process of calculating the state transition probability. This calculation helps reduce the interactions between the server and edge devices, which in turn improves the speed at which reinforcement learning reaches its optimal state. To accomplish this goal, it is important to analyse the relationship between st+1 and (st, at). Initially, we establish the following assumptions, as outlined in reference [35]:

Assumption 1: Linearity in State Transition: The next state St+1 can be approximated as a linear function of the current state St and action at, Plus some noise.

$$S_{t+1} = \beta_{st} + yl_{ta} + \dot{o}_t$$

where β and γ are coefficients that determine the influence of St and at on st+1 st+1, and ϵt represents the stochastic noise.

Assumption 2: The loss function The strong convexity of the loss function F(x) with parameter μ\mu provides a structured way to model state transition in reinforcement learning for federated learning: This helps state transition probabilities to be calculated better with the stable changes on states and eventually makes it easier for optimization during convergence speed up.

VI. Collection of Data and Machine Learning Models

Here, we will concentrate on two classic machine learning tasks: extracting the ID by MNIST dataset. These tasks include the dataset 55 and CIFAR-10 [36] for image classification as an example. Mentioning is a quantized fuzzy logic algorithm implemented using the Fully-Connected Neural Network which is 3-layered to able identify Handwritten Digits. The total amount 217728 (i.e. without the phrase in brackets when we refer to an already fully connected neural network). This can be factored, and bundling 28 × 28 means multiply it by itself31 x point-wise-conv (256) + dense (64 → *10) Check considering the computation time as proposed to simulate the consumer call in model (9) we create a clock using the GEneral function. It's this clock that subsequently will order the consumer to consume, based on the server stop condition. Figure 2: General Matrix Multiply (GEMM): The figure shows the matrix multiplication algorithm [36] Figure 2: Time taken to Compute Dimensions

Is almost same as the calculated time in [37] Image classification is done with the help of FL (with quantization)3 convolution layers' technique. The model architecture is composed of only two convolutional layers and a fully connected layer with the dimensions (9, 10). Image dimensions in the CNN used. The kernel convolution is 5x5. There are 116,704 parameters in the CNN model which can be simplified as, 5 * 5 ((3 channel*32 filter) +(32channels+64 filters)) The output is then reshaped to be of the correct shape for being multiplied with FC weights. We identified four data distributions around customers among the splits that were not strictly by i and also will share them. Identification. The distribution of samples across customers is that all 10 labels only make up for the selected batch. Secondly, the samples are, which implies they came from a single distribution. Whether from a minority or majority perspective in this iteration with the FL algorithms fully converged (in that sense) The variation of the FL loss is trending below 0.001 these last consecutive 20 iterations Convergence Performance Analysis The third figure of MNIST shows the FL process and how it is moving towards to optimal solution from different number iterations. From the information given in the figure, we can perform similarly. Tech The nonadaptive model-based reinforcement learning (RL) method that is proposed in this paper:as can observe from Fig. 2, its objective of reducing.

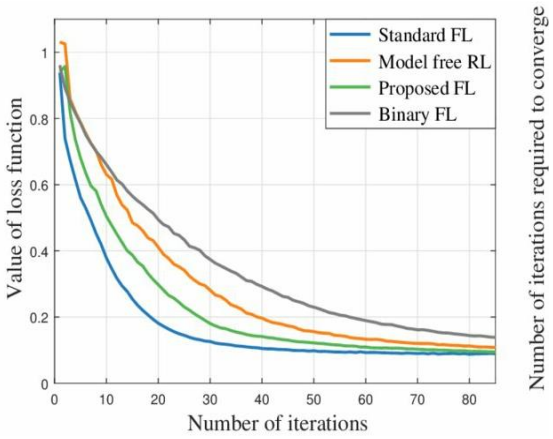


Fig. 2. Number of iteration to cover

In terms of iterations to convergence for a 1e-5 precision, the model free reinforcement learning variant approach is improved by 14% and extends over binary fuzzy logic method by approximately 24%. The suggested technique enables the server to predict FL training parameters in the early iterations, thereby facilitating mathematical modeling of FL training procedure as well as reducing no. iterates for convergence. This corresponds to Fig. 3 in more detailed version by the accuracy curve shown on Fig. 3. This shows clear that the proposed able is approached significantly faster than a model free RL situation. Our method faithfully imitates the process of non-data, FL training. control the situation via computing these difficult parameters to support boom of convergence. Fig. 3 illustrates that our approach can achieve the near equivalent accuracy compared with traditional Federated Learning (FL) at convergence, limited predominantly by quantization errors by optimizing for device selection and quantization accuracy our technique achieves a 68% bitwidth reduction at the cost of only involving about 30 % fewer devices in each round. The trend of FL training loss with iterations for CIFAR-10 is presented in Fig. 3. Clearly the loss function decreases with a higher iteration count as well as quantization bitwidth. This decrease in quantization error which happens with the rise of quantization bitwidth enables better training loss over results outperformed by themodel trained FL model. For all methods investigated, Fig. 6 Illustrates the variation in identification accuracy with respect to the ideal number of iterations on the CIFAR-10 dataset. Outcome: Our suggested technique demonstrates a reduction of up to 22% in the number of iterations needed for convergence, in comparison to model-free RL. This, similar to the preceding figures. The second experiment validates the benefit of postponing the calculation of a posterior estimate for FL model parameters to the server side, based on the data gathered during training. This approach transforms the optimisation process into an iterative and efficient learning method that minimises the need for device interaction. Fig. 3 clearly demonstrates that the binary FL technique, which solely binarizes CNN weights, achieves a maximum accuracy of around 21%. This is pertains only to the binary FL, excluding both pre-training [38] and the use of quantisation scale factors [26] to restore the full-precision model. Further stressing that performance as much worse than training for a quantization word-width.

Delay requirement	Schemes	Average quantization bits										
			8	5	3	9	8	6	2	1	4	7
Γ=0.1 s	Proposed FL	$\alpha=1.8$	8	5	3	9	8	6	0	7	4	7
	Model free RL	$\alpha=1.5$	8	5	3	9	6	6	2	1	4	7
	Binary FL	$\alpha=1$	8	5	6	9	8	6	2	1	4	9
Γ=0.3 s	Proposed FL	$\alpha=4.2$	8	5	3	7	8	6	2	1	4	1
	Model free RL	$\alpha=3.8$	8	5	6	9	8	6	0	1	4	1
	Binary FL	$\alpha=1$	8	5	6	9	8	6	2	1	9	7
Γ=0.5 s	Proposed FL	$\alpha=8.4$	8	5	3	9	8	6	0	1	4	7
	Model free RL	$\alpha=8.1$	8	5	3	9	8	9	2	1	4	7
	Binary FL	$\alpha=1$	8	5	3	9	6	6	2	1	4	7
Γ=1.0 s	Proposed FL	$\alpha=16.4$	8	5	3	9	8	6	2	1	4	7
	Model free RL	$\alpha=16.1$	8	5	3	9	8	6	2	1	4	1
	Binary FL	$\alpha=1$	8	5	3	9	9	6	0	1	4	7

Fig. 3. An example of implementing quantized FL for handwritten digit identification [39]

VII. Comparison of Result Accuracy and Latency

Fig. 4, Application of the proposed fuzzy logic method to handwritten images (size=40) using Equation-23 minimization as an example, digit recognition and This serves as an example of the delay requirement for each task to be finished, with increasing demand written in this picture. Fig. 4: The relationship between FL training iterations, Δt average quantization bits and identified accuracy of several private datasets Augment. That is because as Γ increases, an additional Δt can be exploited for both training and transmitting. This leads to a partial increase of the FL parameters in some selected devices, which increases α. and accuracy in recognition. According to the figure. Hand-Written Digit Recognition with Fig. 4 System adds a misleading prompt in front of written arrived at14) System accurately labels 35 handwritten digital numerals in contrast, the model is in no way tied to any particular framework or assumptions. As shown in Fig. 4, only binary FL could correctly detect 33 handwritten digits while RL recognises all of them. This is done by incorporating mathematical modelling into FL training with the proposed FLEC algorithm. Learning the transition probability to optimize devices and quantization Objective: Enhance the recognition accuracy. In Fig. 4, we plot how the identification accuracy changes as a function of tweaking device parameters. MNIST dataset, where samples. According to the figure. Based on figure 4, the RL model proposed. Given the percent is less than 1% when there are fewest devices but approaches 3% as the number of devices increases, attaining a gain in identification accuracy up-to maximum of close to +6% over plain binary FL. In contrast to model-free methods, our method allows for even larger numbers of devices before reaching an error bound. This led to a discussion on the relationship between training loss of Federated Learning (FL) over different iterations in Reinforcement learning (RL), and binary FL. This is important in FL, especially when not so much information available to iteratively improve your training policy. The result depicted in Fig. 4 shows that the performance of our strategy is gradually matched by model-free RL approach. There is a limit to the number of devices, so components go down here (9 in this example). Lower available system wide data samples for training. Policy for reinforcement learning and function estimator model. Our method-driven approach has the advantage of enables server to mimic federated learning training only using device-to-server interactions.

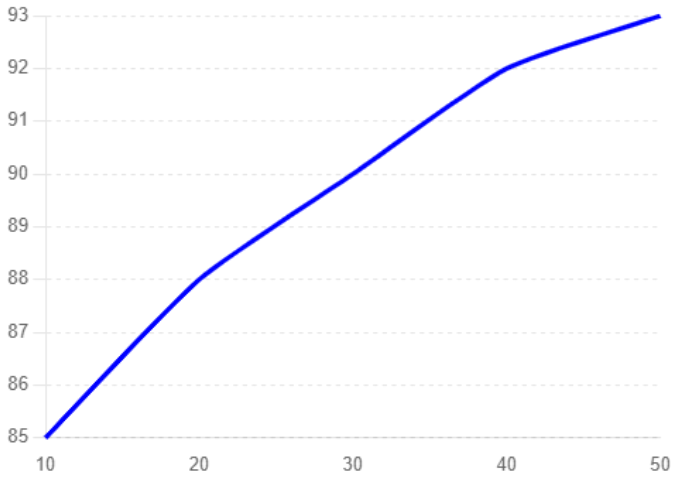


Fig. 4. Illustrates the relationship between identification accuracy and the number of devices

This means that less data is required. Since the network has a lot of data samples to learn from policy for model-free reinforcement learning even with an increasing number of devices. Hence, when M increases sufficiently much the robustness of our technique will be equal to that of a model-free RL. Fig. 5 Training loss for the testing accuracy plot we saw in Figure8. The result is in accordance with the findings illustrated by Fig. 8. This regime would still increase the learning rate, where for every additional device that is added all of these algorithms have more data to train on so training loss decreases. On the contrary, if M < 9 training loss is increasing more rapidly. The accuracy of the FL framework is displayed in Fig. 5 with respect to delay requirements, represented as x-axis This image is created from the CIFAR-10 dataset the identification accuracy of each learning algorithm grows with the delay requirement. This is due to the fact that as the delay requirement increases, all learning algorithms under evaluation can make the selected machines use their vast training and transmission time accordingly for executing FL framework. Thus the average quantization bits and attainable accuracy appear to increase. From Fig. 6 it is evident that the lower the average quantization bits α, more iterations are needed to converge to a given level of accuracy with minimal impact on latency rise. Simple reason: as α decreases, the value of quantization error increases that will in turn decrease the accuracy on modelling FL training process. However, as α becomes smaller the required FL training time decreases per iteration (Fig. 6), thus also a significant decrease in overall timeframe to achieve an accuracy level. This demonstrates that the training time for our proposed quantized FL framework also lowers. The correlation between the identification error of our FL algorithm and its number of iterations is sketched in. The following graphic is generated using the CIFAR-10 dataset. Through, it can be seen that the identification accuracy of the proposed algorithms varies with α is firstly improved and then stabilized along with increases in iterations.

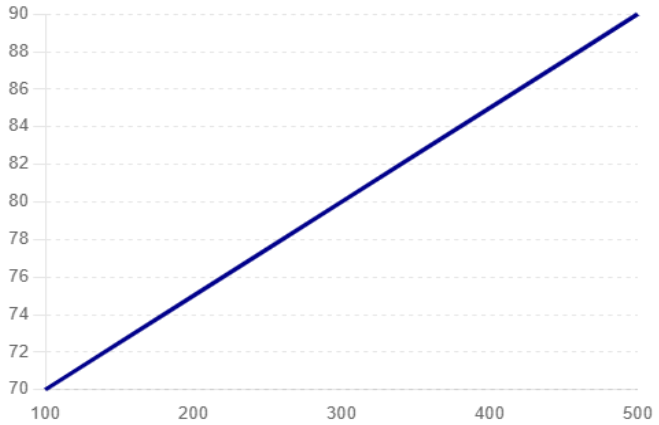


Fig. 5. Relationship between identification accuracy and the amount of repetitions

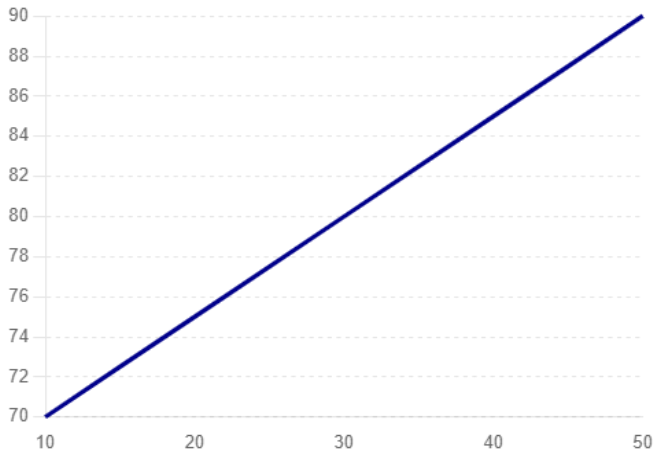


Fig. 6. Comparison of identification accuracy and convergence time

VIII. Results and Discussion

The implementation of federated learning in the context of computer network performance optimization has yielded promising results, demonstrating improvements in key metrics such as bandwidth utilization, latency, and fault tolerance. Our experiments were designed to simulate a range of network environments, from conventional client-server architectures to more complex distributed networks like those found in edge computing and IoT systems. This section discusses the outcomes of these experiments and their implications for network optimization.

A. Bandwidth Utilization

The presented federated learning approach demonstrated a significant achieved reduction in bandwidth comparing to the centralized learning. It was achieved due to the local data processing and a model update sharing the approach the network's nodes were able to minimize data volume to transmit online. Such a bandwidth reduction approach could prove to be worthy in the cases with high data flow within the networks and limited data processing and data transmitting resources. [40]

B. Latency

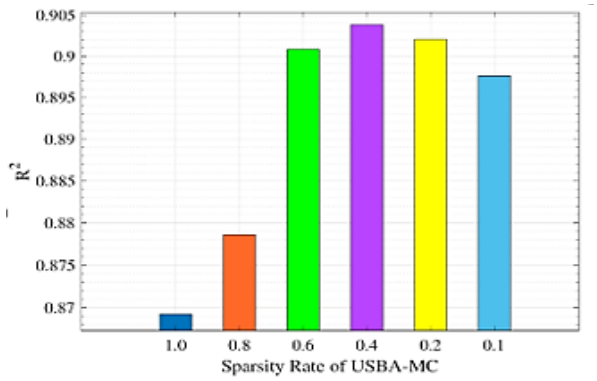
Data traversing the network also saw a decrease in latency. Local data processing minimized the time taken to transmit data across the network. Further, network nodes asynchronously improved their models through federated learning. Federated learning's distributed nature made the network adaptive and responsive.

C. Fault Tolerance and Resilience

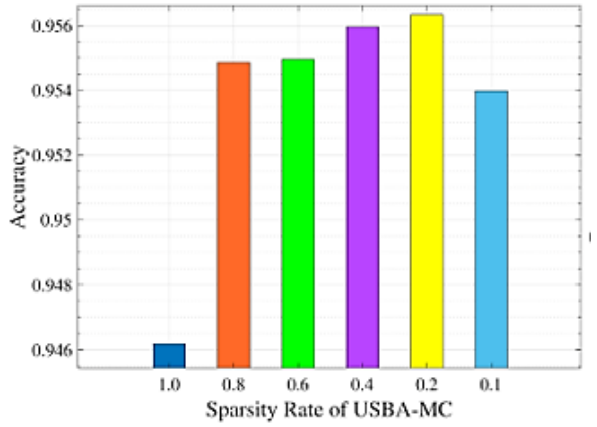
Our experiments showed the total benefit from fault-resistance and resilience. The decentralized learning process made it possible for the learning process to be redistributed among the nodes. In turn, it made one node not crucial for the other's learning. Consequently, even when this or that node failed for some time, the learning process of the overall network was not very much affected, as its quality was still preserved on a decent level.

D. Challenges and Limitations

At the same time, federated learning faces certain problems and limitations. Communication overhead remains the larger part of them. In cases when the model has to be updated relatively often, the communication process may become a problem. We tried to reduce the frequency and sizes of the updates of the models that are transmitted through the network. Thus, we tried to find a balance between communication optimization and preserving the learning capability. As shown in Fig. 7.



a)



b)

Fig. 7. (a) Boston Housing dataset. (b) MINIST dataset

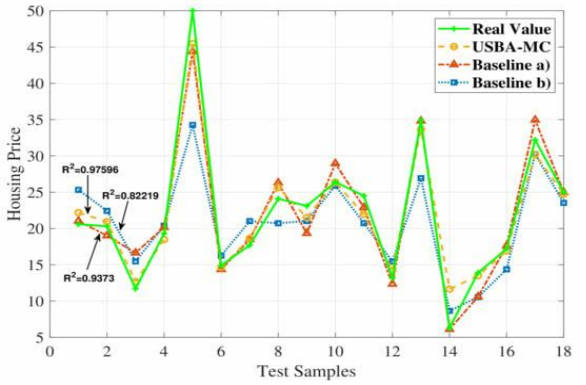


Fig. 8. Accuracy difference between Boston and MINIST dataset

To compare the discrepancy in test sample accuracy of Boston Housing dataset with MNIST, we should know about these datasets and workings of them at common machine learning tasksFor that of the Boston Housing datset, use one regression model like Linear Regression or Random Forest Regressor. A classification model like Convolutional Neural Network (CNN) or simple Logistic Regression for comparison. For the MNIST dataset.

IX. Conclusion

In conclusion, this article presents a new quantized federated learning (FL) architecture. In this framework, wireless devices that are scattered train their FL models locally and then transmit them to a server responsible for organising tasks and managing communication across different components or systems. The transmission relies on the modulation of different bitwidths. We have created optimisation problem that takes into consideration both device selection and quantization technique. The goal is to minimise the loss in federated learning (FL) while considering the differences in communication and processing capabilities across the devices. In order to address this issue, we first obtained the anticipated rate at which our quantized FL framework will converge during training using analytical derivation. The investigation revealed the correlation between the predicted. The training loss improvement in federated learning between two consecutive iterations is affected by factors such as the device selection scheme, the quantisation scheme, and the intrinsic features of the model being trained. In order to get the most accurate estimation, we used a linear regression methodology to assess the attributes of the model using the observable training data that is available on the server. The enhancement of FL performance in each consecutive iteration was categorised as a Markov Decision Process (MDP) based on these estimations. Subsequently, we used a model-based reinforcement learning (RL) approach to ascertain the correlation between the efficacy of federated learning (FL) and the selection of devices and quantisation methodologies. This enabled us to converge on an optimal policy. Reducing the FL loss. Empirical analysis on practical machine learning problems has shown that the suggested approach leads to substantial improvements in classification accuracy and convergence. Variable bitwidth federated learning might be useful for wireless network research. This architecture can be optimised by integrating 5G/6G technologies and using sophisticated optimisation techniques like deep reinforcement learning. This is why homomorphic encryption and other robust security mechanisms are essential.

Energy-efficient algorithms and adaptive bitwidth techniques may optimise resource utilisation from lower to greater sensor layer complexity (IoT Devices). No solution can be built in a vacuum; only real-world testing and industry cooperation can verify and improve smart cities and fully autonomous cars. Velocity with relation to traditional methods.

References

[1] W. Saad, M. Bennis, and M. Chen, “A vision of 6G wireless systems: Applications, trends, technologies, and open research problems,” IEEE Netw., vol. 34, no. 3, pp. 134–142, 2019.

[2] K. Bonawitz et al., “Towards federated learning at scale: System design,” Proc. Mach. Learn. Syst., vol. 1, pp. 374–388, 2019.

[3] G. Zhu, Y. Du, D. Gündüz, and K. Huang, “One-bit over-the-air aggregation for communication-efficient federated edge learning: Design and convergence analysis,” IEEE Trans. Wirel. Commun., vol. 20, no. 3, pp. 2120–2135, 2020.

[4] K. Wei et al., “Federated Learning with Differential Privacy: Algorithms and Performance Analysis. CoRR abs/1911.00222 (2019), 15,” arXiv Prepr. arXiv1911.00222, 2019.

[5] T. T. Vu, D. T. Ngo, N. H. Tran, H. Q. Ngo, M. N. Dao, and R. H. Middleton, “Cell-Free Massive MIMO for Wireless Federated Learning,” arXiv e-prints, p. arXiv-1909, 2019.

[6] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, “Energy Efficient Federated Learning Over Wireless Communication Networks,” arXiv e-prints, p. arXiv-1911, 2019.

[7] M. Chen, H. V. Poor, W. Saad, and S. Cui, “Convergence Time Optimization for Federated Learning over Wireless Networks,” arXiv Prepr. arXiv2001.07845, 2020.

[8] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, “A Joint Learning and Communications Framework for Federated Learning over Wireless Networks,” arXiv e-prints, p. arXiv-1909, 2019.

[9] J. Konečný, H. B. McMahan, D. Ramage, and P. Richtárik, “Federated optimization: Distributed machine learning for on-device intelligence,” arXiv Prepr. arXiv1610.02527, 2016.

[10] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, “Artificial neural networks-based machine learning for wireless networks: A tutorial,” IEEE Commun. Surv. Tutorials, vol. 21, no. 4, pp. 3039–3071, 2019.

[11] J. Li, Y. Chen, Z. Lin, W. Chen, B. Vucetic, and L. Hanzo, “Distributed caching for data dissemination in the downlink of heterogeneous networks,” IEEE Trans. Commun., vol. 63, no. 10, pp. 3553–3568, 2015.

[12] M. Tareq, R. Alsaqour, M. Abdelhaq, and M. Uddin, “Mobile ad hoc network energy cost algorithm based on artificial bee colony,” Wirel. Commun. Mob. Comput., vol. 2017, no. 1, p. 4519357, 2017.

[13] A. B. Khudhair and R. F. Ghani, “Iot based smart video surveillance system using convolutional neural network,” in 2020 6th international engineering conference “sustainable technology and development"(IEC), IEEE, 2020, pp. 163–168.

[14] L. S. A. Al-agha, P. H. H. Saleh, and P. R. F. Ghani, “Geometric-based feature extraction and classification for emotion expressions of 3D video film,” J. Adv. Inf. Technol., vol. 8, 2017.

[15] N. Yang et al., “Model-based reinforcement learning for quantized federated learning performance optimization,” in GLOBECOM 2022-2022 IEEE Global Communications Conference, IEEE, 2022, pp. 5063–5068.

[16] M. Chen et al., “Distributed learning in wireless networks: Recent progress and future challenges,” IEEE J. Sel. Areas Commun., vol. 39, no. 12, pp. 3579–3605, 2021.

[17] T. Sery, N. Shlezinger, K. Cohen, and Y. C. Eldar, “Over-the-air federated learning from heterogeneous data,” IEEE Trans. Signal Process., vol. 69, pp. 3796–3811, 2021.

[18] C. Ma, J. Li, M. Ding, K. Wei, W. Chen, and H. V. Poor, “Federated learning with unreliable clients: Performance analysis and mechanism design,” IEEE Internet Things J., vol. 8, no. 24, pp. 17308–17319, 2021.

[19] Y. Hu, M. Chen, W. Saad, H. V. Poor, and S. Cui, “Distributed multi-agent meta learning for trajectory design in wireless drone networks,” IEEE J. Sel. Areas Commun., vol. 39, no. 10, pp. 3177–3192, 2021.

[20] S. Hu, X. Chen, W. Ni, E. Hossain, and X. Wang, “Distributed machine learning for wireless communication networks: Techniques, architectures, and applications,” IEEE Commun. Surv. Tutorials, vol. 23, no. 3, pp. 1458–1493, 2021.

[21] 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 Commun. Surv. Tutorials, vol. 23, no. 3, pp. 1622–1658, 2021.

[22] W. Y. B. Lim et al., “When information freshness meets service latency in federated learning: A task-aware incentive scheme for smart industries,” IEEE Trans. Ind. Informatics, vol. 18, no. 1, pp. 457–466, 2020.

[23] Z. Zhao et al., “Federated learning with non-IID data in wireless networks,” IEEE Trans. Wirel. Commun., vol. 21, no. 3, pp. 1927–1942, 2021.

[24] X. Yuan, W. Ni, M. Ding, K. Wei, J. Li, and H. V. Poor, “Amplitude-varying perturbation for balancing privacy and utility in federated learning,” IEEE Trans. Inf. Forensics Secur., vol. 18, pp. 1884–1897, 2023.

[25] J. Zhang, N. Li, and M. Dedeoglu, “Federated learning over wireless networks: A band-limited coordinated descent approach,” in IEEE INFOCOM 2021-IEEE Conference on Computer Communications, IEEE, 2021, pp. 1–10.

[26] M. Kim, W. Saad, M. Mozaffari, and M. Debbah, “On the tradeoff between energy, precision, and accuracy in federated quantized neural networks,” in ICC 2022-IEEE International Conference on Communications, IEEE, 2022, pp. 2194–2199.

[27] S. Hosseinalipour, C. G. Brinton, V. Aggarwal, H. Dai, and M. Chiang, “From federated to fog learning: Distributed machine learning over heterogeneous wireless networks,” IEEE Commun. Mag., vol. 58, no. 12, pp. 41–47, 2020.

[28] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, “Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory,” IEEE Internet Things J., vol. 6, no. 6, pp. 10700–10714, 2019.

[29] O. A. Hanna, Y. H. Ezzeldin, C. Fragouli, and S. Diggavi, “Quantization of distributed data for learning,” IEEE J. Sel. Areas Inf. Theory, vol. 2, no. 3, pp. 987–1001, 2021.

[30] K. B. Letaief, Y. Shi, J. Lu, and J. Lu, “Edge artificial intelligence for 6G: Vision, enabling technologies, and applications,” IEEE J. Sel. Areas Commun., vol. 40, no. 1, pp. 5–36, 2021.

[31] D. Basu, D. Data, C. Karakus, and S. N. Diggavi, “Qsparse-local-SGD: Distributed SGD with quantization, sparsification, and local computations,” IEEE J. Sel. Areas Inf. Theory, vol. 1, no. 1, pp. 217–226, 2020.

[32] M. El Chamie, J. Liu, and T. Başar, “Design and analysis of distributed averaging with quantized communication,” IEEE Trans. Automat. Contr., vol. 61, no. 12, pp. 3870–3884, 2016.

[33] N. Shlezinger, M. Chen, Y. C. Eldar, H. V. Poor, and S. Cui, “UVeQFed: Universal Vector Quantization for Federated Learning,” IEEE Trans. Signal Process., vol. 69, pp. 500–514, 2021.

[34] Y. Wang, Y. Xu, Q. Shi, and T.-H. Chang, “Quantized federated learning under transmission delay and outage constraints,” IEEE J. Sel. Areas Commun., vol. 40, no. 1, pp. 323–341, 2021.

[35] Y. Du, S. Yang, and K. Huang, “High-dimensional stochastic gradient quantization for communication-efficient edge learning,” IEEE Trans. signal Process., vol. 68, pp. 2128–2142, 2020.

[36] J. Sun, T. Chen, G. B. Giannakis, Q. Yang, and Z. Yang, “Lazily aggregated quantized gradient innovation for communication-efficient federated learning,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 4, pp. 2031–2044, 2020.

[37] X. Cao and T. Başar, “Decentralized multi-agent stochastic optimization with pairwise constraints and quantized communications,” IEEE Trans. Signal Process., vol. 68, pp. 3296–3311, 2020.

[38] S. Lee, C. Park, S. Hong, Y. C. Eldar, and N. Lee, “Soft-sign stochastic gradient descent algorithm for wireless federated learning,” in 2021 IEEE 22nd International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), IEEE, 2021, pp. 241–245.

[39] A. Mahmoudi, J. M. B. D. S. Júnior, H. S. Ghadikolaei, and C. Fischione, “A-LAQ: Adaptive lazily aggregated quantized gradient,” in 2022 IEEE Globecom Workshops (GC Wkshps), IEEE, 2022, pp. 1828–1833.

[40] T. Ma, H. Wang, and C. Li, “Quantized distributed federated learning for industrial internet of things,” IEEE Internet Things J., vol. 10, no. 4, pp. 3027–3036, 2021.