Original research

DIGITAL HEALTH

DIGITAL HEALTH
Volume 9: 1–12
© The Author(s) 2023
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/20552076231203604
journals.sagepub.com/home/dhj
Sage

A lightweight deep learning approach for COVID-19 detection using X-ray images with edge federation

Sohaib Bin Khalid Alvi[1,2], Muhammad Ziad Nayyer[1], Muhammad Hasan Jamal[3], Imran Raza[3], Isabel de la Torre Diez[4], Carmen Lili Rodriguez Velasco[5,6,7], Jose Manuel Brenosa[5,8,9] and Imran Ashraf[10]

Abstract

Objective: This study aims to develop a lightweight convolutional neural network-based edge federated learning architecture for COVID-19 detection using X-ray images, aiming to minimize computational cost, latency, and bandwidth requirements while preserving patient privacy.

Method: The proposed method uses an edge federated learning architecture to optimize task allocation and execution. Unlike in traditional edge networks where requests from fixed nodes are handled by nearby edge devices or remote clouds, the proposed model uses an intelligent broker within the federation to assess member edge cloudlets' parameters, such as resources and hop count, to make optimal decisions for task offloading. This approach enhances performance and privacy by placing tasks in closer proximity to the user. DenseNet is used for model training, with a depth of 60 and 357,482 parameters. This resource-aware distributed approach optimizes computing resource utilization within the edge-federated learning architecture.

Results: The experimental results demonstrate significant improvements in various performance metrics. The proposed method reduces training time by 53.1%, optimizes CPU and memory utilization by 17.5% and 33.6%, and maintains accurate COVID-19 detection capabilities without compromising the F1 score, demonstrating the efficiency and effectiveness of the lightweight convolutional neural network-based edge federated learning architecture.

Conclusion: Existing studies predominantly concentrate on either privacy and accuracy or load balancing and energy optimization, with limited emphasis on training time. The proposed approach offers a comprehensive performance-centric solution that simultaneously addresses privacy, load balancing, and energy optimization while reducing training time, providing a more holistic and balanced solution for optimal system performance.

Keywords

Public health, federated learning, edge computing, deep learning

Submission date: 27 March 2023; Acceptance date: 8 September 2023

[1]Department of Computer Science, GIFT University, Gujranwala, Pakistan
[2]Department of Information Technology, University of the Punjab Gujranwala Campus, Gujranwala, Pakistan
[3]Department of Computer Science, COMSATS University Islamabad, Lahore, Pakistan
[4]Department of Signal Theory and Communications and Telematic Engineering. Unviersity of Valladolid, Valladolid, Spain
[5]Universidad Europea del Atlántico. Isabel Torres 21, Santander, Spain
[6]Universidad Internacional Iberoamericana Campeche, México
[7]Fundación Universitaria Internacional de Colombia Bogotá, Colombia
[8]Universidad Internacional Iberoamericana Arecibo, Puerto Rico, USA
[9]Universidade Internacional do Cuanza. Cuito, Bié, Angola
[10]Department of Information and Communication Engineering, Yeungnam University, Gyeongsan, Korea

Corresponding author:
Imran Ashraf and Isabel de la Torre Diez, Department of Information and Communication Engineering, Yeungnam University, Gyeongsan 38541, Korea.
Emails: imranashraf@ynu.ac.kr; isator@tel.uva.es

Introduction

Artificial intelligence (AI) is revolutionizing several emerging solutions to automate the healthcare sector. The advancements in healthcare automation have attracted many researchers to propose cost-efficient and high accuracy providing solutions such as automatic diagnosis, disease detection, prediction, bioinformatics, etc. Similarly, several solutions have been proposed for COVID-19 detection using X-ray images. However, algorithms used in such solutions often require high computational resources and consume a substantial amount of energy.[1] The existing solutions implement a variety of hybrid deep-learning (DL) algorithms to detect COVID-19 using X-ray images. Promising results are reported in several studies but complex DL algorithms require additional computational resources and the time to train such algorithms is also high.[2–4] The computation scarcity challenge is addressed by placing the DL models and X-ray images on the cloud[6–10] but this results in many concerns such as data privacy, network latency, and additional bandwidth requirements. Additionally, a solution implementing localized model training and sharing the model at a global level without compromising data privacy is the need of the hour.

The proposed work is focused on distributing the computational requirement among several machines available to the user in closer proximity as edge nodes.[11] The concept of the federation in cloud computing is entirely different from the concept of edge federation. The edge federated model is based on a central brokerage system and considers the closer proximity edge devices where the broker also resides at the edge.[12] The broker is responsible for maintaining the information of all the edge nodes. The broker allocates the tasks to edge nodes based on available resources on the edge nodes. These edge nodes in return send the results of the assigned task to the broker for the aggregation of the results.

The proposed approach is divided into a three-tier federated learning (FL) architecture, where the first layer comprises clients that send data to the nearest fixed node in the second layer. After receiving the data in the form of chest X-ray images[13,14] of the patients from the clients and the updated convolutional neural network (CNN) from the broker, the fixed nodes start model training. The extracted features from each client are sent to the edge nodes on the third layer. These edge nodes then aggregate the results using an artificial neural network (ANN) and send the results to the broker to update the aggregated model. However, during feature extraction or model training, if any node faces a resource scarcity issue, the node can request the broker to find the nearest node within the federation to offload the task. To the best of the authors' knowledge, no technique targets resource sharing and load balancing, training time, and privacy altogether, without compromising accuracy. The concept of the edge-based federation has also not been reported in the literature for distributed DL problems. The major contributions of the proposed work are as follows:

- Novel lightweight CNN-based edge FL architecture: The proposed architecture, combining lightweight CNNs with edge FL for COVID-19 detection using chest X-rays, presents a novel approach to address the challenges of cloud deployment. This novel architecture focuses on minimizing computational costs, reducing latency, and optimizing bandwidth requirements, thus offering a unique solution in the field.
- Enhanced privacy preservation in edge FL: The approach introduces an innovative method to preserve patient privacy by enabling multiple clients, such as hospitals, to collaboratively train the model without sharing their local data. This novel privacy-preserving technique ensures data confidentiality while allowing for effective model training, which is particularly important in healthcare settings.

Related work

Multiple studies on FL have been reported in the literature that focus on the utilization of resources, security, privacy, and energy efficiency. The dispersed FL (DFL) proposed in Khan et al.[15] minimizes the FL cost by using an integer linear optimization approach for DFL and dividing the problem into two sub-problems. The first problem is resource allocation and association, while the second problem is the relaxation of association and allocation problems, which are converted into convex optimization problems. The proposed algorithm works iteratively to resolve the variable of association and compute the second variable of resource allocation. The algorithm runs iterations till the optimization of resources is complete. While keeping in view resource utilization, privacy is also a big issue that is resolved using privacy-enhanced FL (PEFL) in Hao et al.[16] The privacy enhancement using FL is initiated to allow the participants to share models without sharing their local data. Despite that the parameters must still be shared, posing a risk to privacy in different industries such as robotics, auto-driving, navigation, etc. As a solution, this study introduces the factor of encryption for sharing the parameters. The data is trained on the local machine after which the encrypted parameters are shared with other participants using a public–private key pair.

The application of FL in internet of things (IoT)-based applications is gaining popularity recently. The scheme proposed in Khan et al.[17] uses the Stackelberg game based on an incentive model for active participation in the model training by the members. The study uses edge networks for the implementation of a federation. The impact of dependent and identically distributed device data is reduced by assigning higher weights to devices with lower capacities to increase the efficiency of the algorithm. An energy optimization scheme, based on fifth generation plus (5G+) technology is proposed in Shi et al.[18] that employs software and hardware-based solutions at the edge networks. High energy consumption is the main obstacle to 5G+ technology. The scheme works by examining different energy consumption models for graphical processing units and wireless transmission. The energy-efficient techniques such as weight quantization, pruning, and gradient sparsification are used for the selection of an optimized model.

FL-based camouflage learning is proposed in Sigg et al.[19] that can be implemented using multiple deep learning (DL) approaches to secure privacy. The proposed approach avoids the sharing of models and data to preserve privacy. The scheme is simulated using camouflage learning on an IoT application that is equipped with the ability to sense light, temperature, and humidity.[20] The data is scaled based on the maximum and minimum sensed values. These values are averaged over 60 s intervals and sent to the coordinator to train a logistic regression model with a learning rate of 0.8 to achieve maximum accuracy.

A training algorithm named FedGKT (Group Knowledge Transfer) is proposed in He et al.,[21] which suggests that scaling up the CNN can achieve better accuracy. When FL produces the load on the edge nodes, the FedGKT minimizes the load of CNN from the edge devices and transfers this load toward the consolidated servers by converting it into a single framework. The model is trained based on residual network (ResNet)[22] variant ResNet-56 and ResNet-110 based on different datasets (CIFAR-10),[23] CIFAR-100, and CINIC-10[24]). The study claims to take 9 to 17 times less computation power as compared to FedAvg[25] and also needs 54 to 105 times lesser parameters than in an edge CNN.

In the medical field, X-rays and computed tomography (CT) scans are easy and less costly resources to extract patient information for diagnosis. FL is also applied in the medical field to look for various anomalies such as abnormal growth in the lungs.[26] The authors proposed a decentralized approach to improve privacy that consists of two models. The first model detects the occurrence of nodules, while the second model confirms its presence. Experimental results show that the proposed approach achieves a higher accuracy as compared to traditional models. Privacy is a big concern in using medical images of patients. A novel approach is proposed in Grama et al.[27] that is robust and has higher accuracy. It also detects and blocks malicious nodes that send bad model updates. The proposed approaches thus help in the reduction of computational and communication burdens and privacy concerns. The trials are run with two healthcare datasets for the diagnosis and classification of the data. From the two datasets, each dataset is trained using three different FL approaches to avoid any malicious activity. Experimental results show that the best robustness is achieved by the Byzantine-robust aggregation keeping the differential privacy intact.

Secure multiparty computation (SMC) is conventionally used in FL, which is susceptible to differential privacy and achieves lower accuracy against the data distributed among different parties. A novel approach is proposed in Truex et al.[28] to gain equilibrium in SMC and differential privacy. The hybrid of SMC and differential privacy results in the reduction of noise injection (adding noise artificially to the ANN input data during the training process) as multiple parties carry out the training process. A variety of DL algorithms is trained using the given technique providing better results in terms of accuracy and privacy. For warfighting, techniques are developed to carry out activities such as contesting, breaching, breaking, and exploiting opponents under the multi-domain operations (MDO) Joint Force. When AI techniques are implemented several challenges arise such as data infection, noise, and partial or full errors that can benefit the opponent. FL is a technique that does not require any sharing of data and trains the model in a collaborative environment. The collaborative environment built on Tactical Edge[29] and FL is implemented in MDO using six different AI strategies addressing the above-mentioned challenges. Table 1 shows a comparative analysis of the discussed techniques considering privacy, accuracy, training time, load balancing, and energy.

The literature review highlights that most of the existing studies focus on privacy and accuracy, and only a few studies focus on load balancing and energy optimization while no study targets training time which is the focus of this study.

Edge federation-based lightweight DL approach

Figure 1 shows the architecture of the proposed edge federation-based DL approach. In edge federation, clients receive services from the nearest edge devices in the federation without worrying about the delay in data transfer. The broker in the federation manages all the edge federation operations such as the management of resources, optimal placement, model sharing, and migration decisions. Other responsibilities of the broker include receiving and sharing resource information with all the edge nodes in the federation to carry out decision-making. In case of resource scarcity at an edge node, a request is forwarded to the broker to find a node in the federation with adequate resources. The broker evaluates the optimal edge concerning the proximity and available resources. If somehow the broker is not working, any edge node can use the local information matrix previously pushed by the broker thus eliminating the chances of failure.[1] Next, the task is offloaded from the connected node to the optimal edge node. The task can be offloaded in the form of workload, code, or virtual machine. In this study, we are considering the task bundled as a virtual machine. After that, the client and the fixed node communicate for the training of the

Table 1. Comparative overview of the discussed works.

Literature	Privacy	Accuracy	Training time	Load balancing	Energy
DFL[15]	✓	–	–	✓	–
PEFL[16]	✓	✓	–	–	–
IoT-based[17]	✓	✓	–	–	–
5G+[18]	–	–	–	–	✓
Camouflage[19]	✓	–	–	–	–
FedGKT[21]	–	✓	–	–	–
CT Scan[26]	✓	–	–	–	–
Medical Images[27]	✓	–	–	–	–
SMC[28]	✓	✓	–	–	–
MDO[29]	✓	✓	–	–	–
Proposed model	✓	✓	✓	✓	–

DFL: dispersed federated learning; PEFL: privacy-enhanced federated learning; IoT: internet of things; 5G: fifth generation; CT: computed tomography; SMC: secure multiparty computation; MDO: multi-domain operations.

model. The model lending task is requested from the broker which is responsible for the optimal DL model placement. This process is independent of the platform, programmer, or underlying infrastructure. If in the worst case, the required resources are not available in the federation, the request is forwarded to the cloud.

As shown in Figure 1, clients P, Q, and R are connected to fixed devices A, B, and C, respectively, which are then connected to access points M and N, respectively. These access points are connected to the edge devices in closer proximity. Model training for the classification of COVID-19 chest X-rays starts at edge devices A, B, and C. Due to the scarcity of resources, both devices request resources from the broker. The broker assigns the task to an optimal edge device K based on latency and computation resource availability. Edge device K lends the model against the specific request to the fixed devices A, B, and C from vicinity Y to vicinity X using edge device J. Both devices start to train the model and after the first round, the weights are returned to the edge device K using the same route as edge device J. Edge device K aggregates the weights and sends these aggregated weights to the fixed device A, B, and C for round two. Similarly, multiple rounds are executed and in the end, the results sent by the edge device K to the fixed device A, B, and C are considered the final results. The same fixed devices A, B, and C are used for detection purposes against the request sent by the clients P, Q, and R.

Problem formulation

Let $RR = \{r_1, r_2, r_3, \ldots, r_n\}$ be the required resource set, $AR = \{a_1, a_2, a_3, \ldots, a_n\}$ be the available resource set, and $ARR = \{ar_1, ar_2, ar_3, \ldots, ar_n\}$ be the additional required resources not available on the immediate edge node, where immediate edge node represents the node with which the user is currently connected. There may be three cases:

(i) The immediate edge node has the required resources available,
(ii) The immediate edge node is unable to fulfill the requirements, and resources from other neighboring edge nodes are borrowed using edge federation, and
(iii) Neither the immediate edge node nor the federation possesses adequate resources to fulfill the requirement and hence the request is forwarded to the cloud.

The last case is considered the worst-case scenario. All the notations used in the system setup are shown in Table 2.

The condition for the eligible node is as follows:

$$f(x) = \begin{cases} 1, & \text{if } RR[r_i] < AR[a_i] \forall i = 1, 2, 3, \ldots, n \\ 0, & \text{Otherwise} \end{cases} \tag{1}$$

Figure 2 shows the workflow diagram of the proposed approach. In case the function returns 1, the first round is executed, and the eligible node is found and added to the

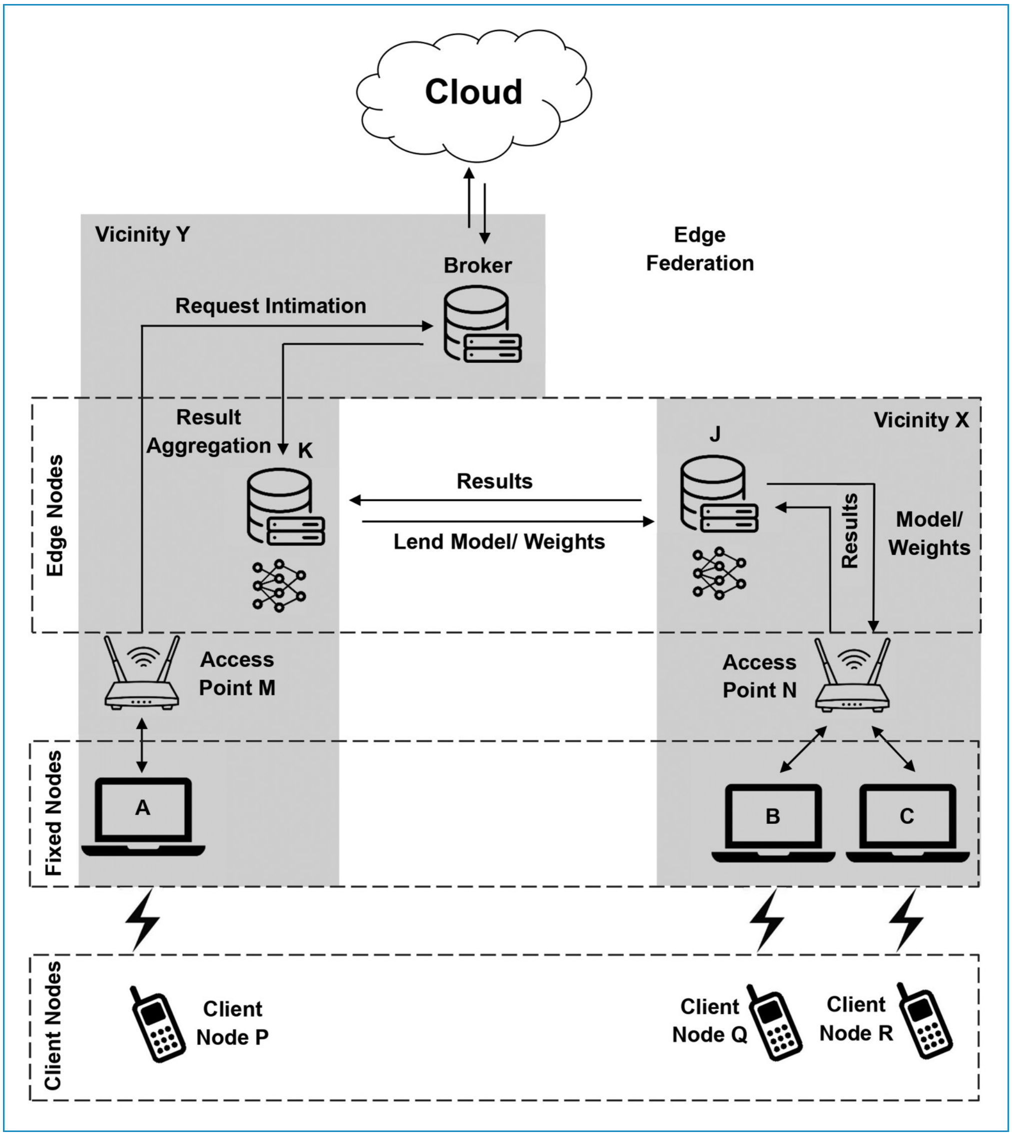

Figure 1. Edge federation-based lightweight deep learning (DL) approach.

table of eligible nodes. However, if the function returns 0 the neighboring edge node is consulted, and the same condition is checked again until the requirements are fulfilled.

Resource aware distributed approach

In a traditional edge network, the request of the fixed node is resolved by the connected nearby edge device, or it is sent to the remote cloud for execution. In the proposed edge federation model, the broker within the federation makes the optimal decision for task offloading to provide the optimal edge node. This decision is based on the current parameters of the member edge cloudlets received at the broker such as available resources and hop count since the objective is to place the task in closer proximity to the user, as shown in Algorithm ??. The main objective of using federated deep learning (FDL) is to train the model without transferring data from the fixed node, which helps in achieving a higher scale of privacy. The fixed nodes contain the data; however, share a small number of weights obtained from the training of CNN. The weights are averaged at edge nodes and used to train the model. Later on, the optimal edge node referred by the broker starts communicating with the requesting node. The DL model is shared between the fixed devices where the data is stored. Algorithm ?? provides details of the client and server-side processing.

Experimental setup

Figure 3(a) to (c) illustrates the three use cases considered in this study. Edge devices receive the DL model from the broker to train on the data available at the edge devices. Fixed client nodes use the trained model for the classification of images.

This study uses three machines and one access point to conduct the experiments. This study uses a testbed for the experiments. The specifications of the machines used for experimentation are given in Table 3. Machine 1 is used for the first case. For the second use case, Machine 1 is used as a client while Machine 2 is used as the server. For the third use case, Machine 1 is used as the first client edge node, Machine 2 is used as the second client edge node while Machine 3 is used as the broker. The access point used for the communication between the client and the server is a TP-link model TL-WR840N V6.20 and has a 300 Mbps speed.

Table 2. Table of notations.

Notation	Description
C_i	Clients
E_i	Edge device
S_D	Pending decision list
R_i	Resources
a_i	Available resources
I_m	Images
R_o	Rounds
W_i	Weights
A_r	Aggregated weights
S_r	Server

All experiments are repeated five times and the best results are reported in this paper as there was insignificant difference across runs. For training the CNN model, two rounds are executed, each of 10 epochs using a publicly available dataset[13,14] containing 13,808 chest X-ray images, each of size 291 × 291 pixels. The dataset comprises 10,192 normal X-ray images and 3616 X-ray images of COVID-19-positive cases. The raw images were red–green–blue,, which consumes more processing power and memory, therefore, before training, the images were preprocessed by removing the edges and converting them to grayscale, as the grayscale images were sufficient for our experimentation and it consumes less processing power and memory. For the third use case, the images are split equally between the two clients' edge nodes. For CNN, we have used DenseNet with a depth of 60 and a total of 357,482 parameters. The total memory used by the CNN model is 1.30 GB. DenseNet is used in this study because it gives smoother decision boundaries and performs well when training data is insufficient which suits our scenario where the data on individual fixed nodes is of small size.

Results

We evaluate the performance and efficacy of the proposed edge-based distributed DL approach using accuracy, precision, recall, and F1 score parameters, the equations of which are shown in equations (2) to (5). To evaluate the resource utilization and load balancing of the proposed approach, we consider execution time, central processing unit (CPU) utilization, and memory utilization.

$$\text{Accuracy} = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \tag{2}$$

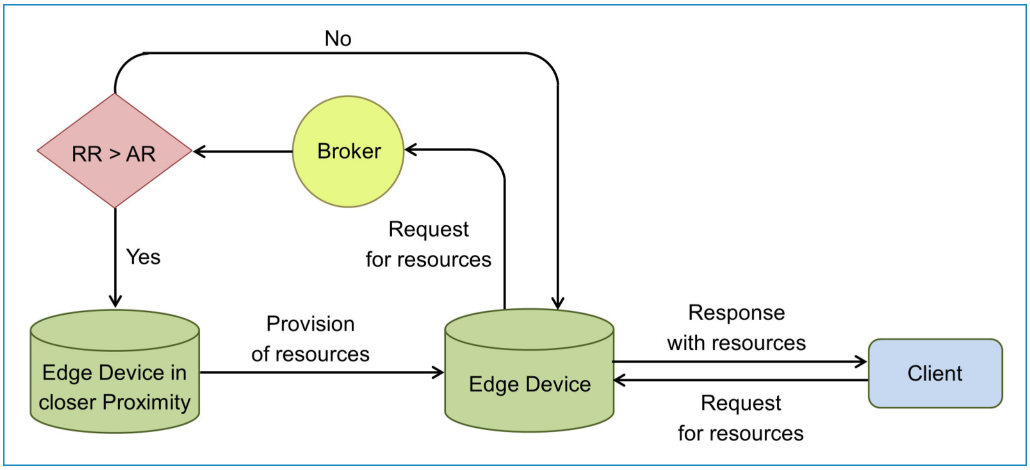

Figure 2. Communication flow between nodes.

$$\text{Precision} = \frac{T_P}{T_P + F_P} \tag{3}$$

$$\text{Recall} = \frac{T_P}{T_P + F_N} \tag{4}$$

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5}$$

In the proposed model, the request is forwarded to the broker where the resource information of all edge nodes is already available. As the request matrix is delivered to

Algorithm 1. Broker.

```
Input: List of clients, list of edge devices, available resources at edges
Output: Required resources to client
Begin:
for each $E_i$ request broker checks the status do
    Check status $S_D$
    if $S_D$ = "DecisionPending" then
        for each edge $E_i$ in edge list do
            for each resource $a_i$ and $r_i$ in available and require resources list do
                if $a_i \geq r_i$ then
                    Push edge $E_i$ in the eligible cloudlet list
                end if
            end for
        end for
        for each cloudlet in the eligible cloudlet list do
            Calculate $h$
        end for
    end if
    E_o = E_i
end for
return $E_o$, $h(E_o)$
End:
```

Algorithm 2. Client--server federation model.

```
Input: Raw images of X-rays
Output: Trained model and updated weights.
Begin: Client side
Preprocessing on the images ($I_m$)
Perform statistical operations
for each client ($C_i$) do
    for each round ($R_o$) do
        send a request for the weights ($W_i$)
    end for
end for
Train the model
for each client ($C_i$) do
    for each round ($R_o$) do
        send results to the server ($S_r$)
    end for
end for
Receive aggregated results ($A_r$)
End:
Begin: Server side
for each client do
    if $R_o = 1$ then
        allot weights to requesting client
        receive and aggregate results ($A_r$)
    else if $R_o > 1 \& R_o \leq n$ then
        allot aggregated weights to requesting client
        receive and aggregate results
    end if
    save aggregated results
end for
return results to client
End:
```

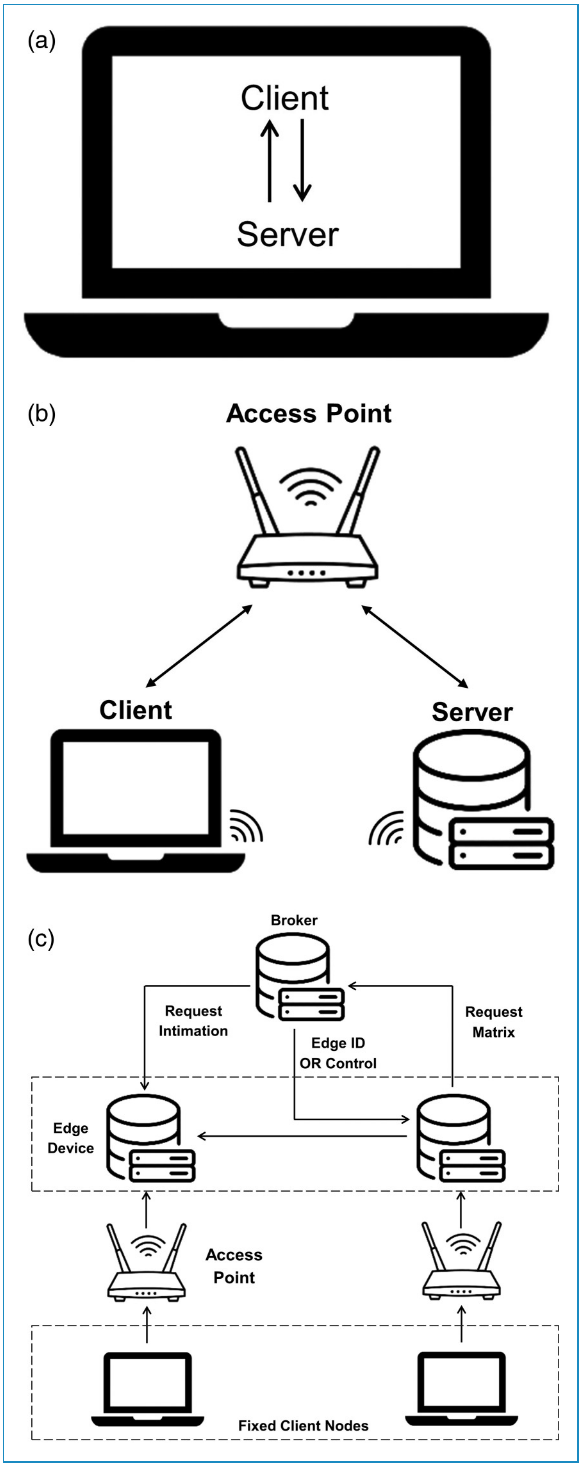

Figure 3. Use cases for the experimental setup. (a) First use case: Client and server are both executed on the same machine; (b) Second use case: Client and server are executed on two separate machines; (c) Third use case: Edge federation comprising two fixed client nodes, two edge devices, and a broker.

the broker, the broker finds one machine in closer proximity with the same model and shares the required resources with the requesting machine. The images remain on the fixed nodes, but the memory and processor resources are shared by edge nodes. The memory and CPU utilization by the broker is negligible as it only receives the request from fixed nodes *A* and *B* and lends the model and weights for model training.

For the first use case, the client and server reside on the same computer. CNN is applied to the actual environment of FDL. The split ratio of data is 80% and 20% for train and test sets, respectively. Table 4 shows the comparative results obtained for all three use cases. For the first use case, the achieved accuracy, precision, recall, and F1 scores are 86.3%, 92.7%, 74.8%, and 82.7%, respectively. For the second use case, the achieved accuracy, precision, recall, and F1 score are 89.6%, 90.5%, 78.3%, and 83.9%, respectively. For the third use case, scores for accuracy, precision, recall, and F1 are 87.8%, 89.9%, 76.3%, and 82.5%, respectively. The model training time for the first use case is 17 min and 10 s. The model training time for the second use case is 15 min and 30 s and the model training time for the third case is 8 min and 3 s. The training time for the third use case is 53.1% and 48.1% less than the first and second use case, respectively. It is observed that our approach significantly reduces the training time without compromising the F1 score.

Figure 4 shows the confusion matrix obtained for the three use cases considered in this study. In the third use case, the data is distributed between two different fixed nodes *A* and *B*. Figure 5 shows the confusion matrix obtained from the fixed node *A* and fixed node *B* for the third use case. The calculated accuracy, precision, recall, and F1 score for fixed node *A* are 89.1%, 93.9%, 73.1%, and 82.2%, respectively. Similarly, for fixed node *B*, the accuracy, precision, recall, and F1 score are 86.6%, 86.9%, 79.6%, and 82.9%, respectively.

Figure 6 shows the CPU and memory utilization of all three use cases. The CPU utilization for the first use case, where only a single machine is used, is 90% on average, the memory utilization is 5.5 GB on average, and the training time is 17 min and 10 s. The CPU utilization for the second use case is 87.5% on average, the memory utilization is 4.75 GB on average, and the training time is reduced to 15 min and 30 s. As compared to the first use case, a decrease of 2.5% is observed in terms of CPU utilization and a decrease of 13.6% is observed in terms of memory utilization.

The CPU utilization for the proposed third use case is 72.5% on average, the memory utilization is 3.65 GB on average, and the training time is reduced to 8 min and 3 s. In terms of CPU utilization, a decrease of 17.5% and 15% is observed for use case three as compared to the first and second use case, respectively. In terms of memory utilization, a decrease of 33.6% and 23.1% is observed for use case three as compared to the first and second use case, respectively. Since the processing at the server is not playing a significant role in the model training, the results of the server side are ignored.

Table 3. Specification of devices used for experimentation.

	Model	RAM/speed	HDD	Processor
Machine 1	HP Pro book 450	8 GB	512 GB	2.20 GHz
Machine 2	Lenovo Think pad	8 GB	128 GB (SSD)	2.10 GHz
Machine 3	Dell Inspiron	4 GB	512 GB	1.80 GHz
Access point	TL-WR840N V6.20	300 Mbps	–	–

Table 4. Comparison of three use cases.

	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)	Training time
First use case	86.3	92.7	74.8	82.7	17 min and 10 s
Second use case	89.6	90.5	78.3	83.9	15 min and 30 s
Third use case	87.8	89.9	76.3	82.5	8 min and 3 s

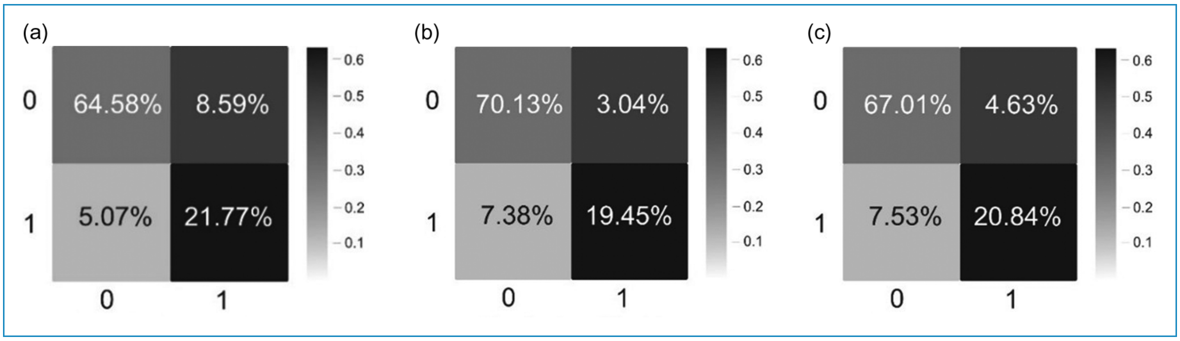

Figure 4. Confusion matrices of the three use cases: (a) first use case; (b) second use case; (c) third use case.

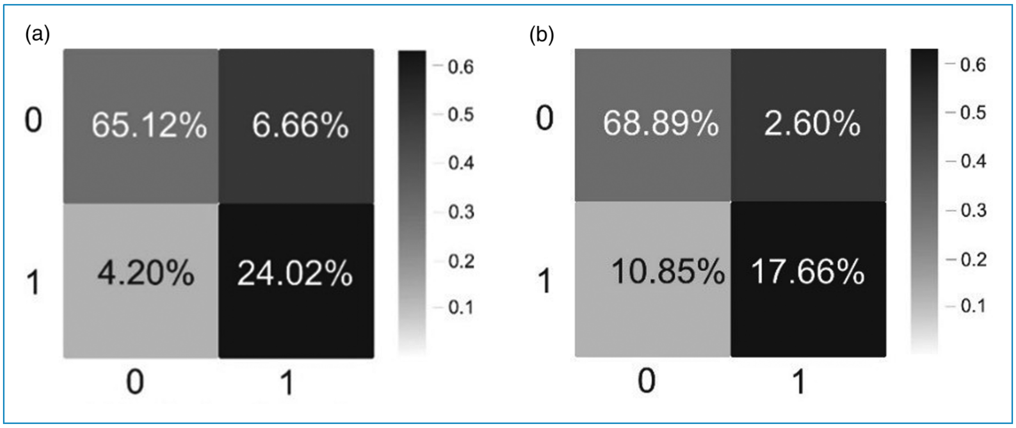

Figure 5. Confusion matrices of nodes in the third use case: (a) fixed node A and (b) fixed node B.

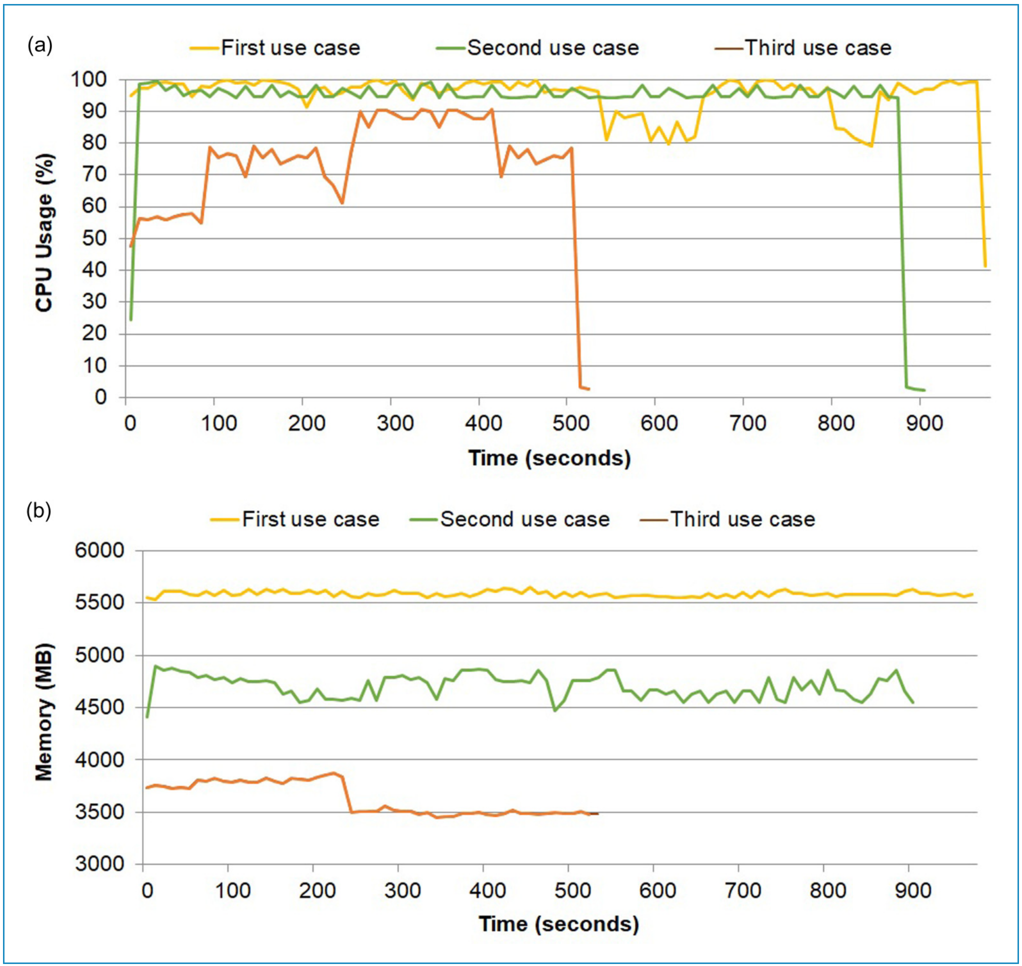

Figure 6. Central processing unit (CPU) and memory utilization for the three use cases while model training: (a) CPU utilization and (b) memory utilization.

In the first use case, the feature extraction and model training is performed on a single machine, which takes more time due to resource constraints. In the second use case, the task of model training and feature extraction is divided between the server and client, respectively, which helps reduce the memory and CPU utilization. However, with the proposed distributed approach, as in the third use case, an edge federation consisting of multiple systems is deployed, which substantially reduces the CPU and memory utilization as well as model training time as compared to the first and second use cases. Additionally, privacy is preserved as the images remain on the fixed nodes in the proposed model.

Discussion

COVID-19 detection using DL methods with X-ray images has been widely studied in the literature. However, these methods often involve complex algorithms that consume significant computational resources and require extensive training time. Moreover, the conventional FL models used for this purpose are centralized in nature and rely on a single heavy-duty machine to execute the complex DL algorithms. This poses challenges when dealing with X-ray data belonging to different patients, which is distributed across various machines in hospitals and medical facilities. The transfer of such confidential data to a single machine not only raises privacy concerns, but also results in significant time delays and encounters limitations related to bandwidth and latency. These limitations underscore the necessity for a distributed approach that eliminates the reliance on a centralized heavy-duty machine, preserves privacy, and reduces dependence on the Internet.

In this research, an edge federation-based lightweight DL approach is proposed that successfully addresses the limitations of conventional FL approaches and achieves the desired objectives. The primary challenge in developing the proposed edge federation approach was to design a distributed lightweight DL model that is well-suited to the environment. Additionally, it was crucial to overcome the limitations of computational resources, bandwidth, and latency while ensuring privacy and accuracy.

The proposed edge federation leverages the normal systems already present in different medical facilities and incorporates them into the federation. This enables the direct transfer of images, whether from patients or diagnostic facilities to fixed nodes within the hospital. Since the primary objective of these images is the analysis by doctors, they naturally need to reach the doctor's system and subsequently become part of the edge nodes as patient data. The proposed model acquires the required model from the broker and performs training on these images to reach a conclusion. Importantly, as these images never leave the intended premises, privacy is effectively maintained. Moreover, the use of edge nodes in closer proximity mitigates the challenges associated with limited bandwidth and latency. Additionally, different edge nodes collaborate and share resources to address computational challenges.

The results of the proposed work indicate its success in achieving the set objectives. However, it is important to acknowledge the limitations of this study. Firstly, there is limited existing research available for comparison, as edge federation for resource sharing in a distributed manner is a relatively new concept. Further studies and comparisons are necessary to establish benchmarks and evaluate the performance of the proposed approach. Additionally, other challenges, such as security and ownership, remain unexplored within this emerging paradigm. Since each edge node falls under different administration compared to the cloud, it is important to address these challenges to ensure the robustness and viability of the proposed edge federation approach.

In conclusion, this research demonstrates the effectiveness of an edge federation-based lightweight DL approach for COVID-19 detection using X-ray images. By addressing the limitations of conventional FL approaches and adopting a distributed framework, this approach offers advantages such as reduced reliance on a centralized heavy-duty machine, preserved privacy, and improved utilization of limited resources. Nonetheless, further research is needed to compare and validate the results, as well as to explore additional challenges associated with security and ownership in this new paradigm.

Conclusion and future directions

This study addresses the challenges of privacy, increased model training time, and the need for higher computational resources in DL algorithms. It introduces a novel lightweight CNN-based edge FL architecture, offering an alternative to traditional cloud-based solutions. The proposed approach employs a load-balancing mechanism that distributes compute-intensive tasks among edge federation nodes, ensuring privacy, reducing model training time, and managing the computational load on edge nodes. The experiments conducted in three phases demonstrate the effectiveness of the approach, showcasing reduced model training time, improved load balancing, and privacy preservation. By participating in an edge federation, healthcare providers can benefit from better model training, data sharing, and computational resources while upholding privacy. Future work can explore additional parameters such as energy and carbon dissipation, adopt hybrid models combining CNN and graphical neural networks for enhanced accuracy and performance, and emphasize trust management in edge federations to expand the federation's membership.

Acknowledgement: Not applicable.

Contributorship: Not applicable.

Declaration of conflicting interests: The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Ethical approval: Not applicable.

Funding: The authors disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research was supported by the European University of the Atlantic.

Guarantor: Not applicable.

ORCID iDs: Muhammad Hasan Jamal https://orcid.org/0000-0002-0114-0887
Imran Ashraf https://orcid.org/0000-0002-8271-6496

References

1. Inam M and Nayyer MZ. Energy-aware load balancing in a cloudlet federation. *Eng Proc* 2021; 12: 27.
2. Chola C, Mallikarjuna P, Muaad AY, et al. A hybrid deep learning approach for covid-19 diagnosis via ct and x-ray medical images. *Comput Sci Math Forum* 2022; 2.
3. Kaya Y, Yiner Z, Kaya M, et al. A new approach to COVID-19 detection from X-ray images using angle transformation with GoogleNet and LSTM. *Meas Sci Technol* 2022; 33: 124011.
4. Yılmaz A. Diagnosing covid-19 from x-ray images with using multi-channel cnn architecture. *J Fac Eng Archit Gazi Univ* 2021; 36: 1761–1774.
5. Nair R, Alhudhaif A, Koundal D, et al. Deep learning-based COVID-19 detection system using pulmonary ct scans. *Turk J Electr Eng Comput Sci* 2021; 29: 2716–2727.
6. Jain R, Gupta M, Taneja S, et al. Deep learning based detection and analysis of COVID-19 on chest X-ray images. *Appl Intell* 2021; 51: 1690–1700.
7. Li Z, Xu X, Cao X, et al. Integrated CNN and federated learning for COVID-19 detection on chest X-ray images. *IEEE/*

ACM Trans Comput Biol Bioinform 2022: 1–11. DOI: 10.1109/TCBB.2022.3184319.

8. Feki I, Ammar S, Kessentini Y, et al. Federated learning for COVID-19 screening from chest X-ray images. *Appl Soft Comput* 2021; 106: 107330.
9. Naz S, Phan KT and Chen YPP. A comprehensive review of federated learning for COVID-19 detection. *Int J Intell Syst* 2022; 37: 2371–2392.
10. Nguyen DC, Pham QV, Pathirana PN, et al. Federated learning for smart healthcare: A survey. *ACM Comput Surv* 2022; 55.
11. LeCun Y, Bengio Y et al. *Convolutional networks for images, speech, and time series*. In: *The handbook of brain theory and neural networks*. Cambridge, MA; MIT Press, 1998, 255–258.
12. Nayyer MZ, Raza I and Hussain SA. Cfro: Cloudlet federation for resource optimization. *IEEE Access* 2020; 8: 106234.
13. Chowdhury MEH, Rahman T, Khandakar A, et al. Can AI help in screening viral and COVID-19 pneumonia? *IEEE Access* 2020; 8: 132665.
14. Rahman T, Khandakar A, Qiblawey Y, et al. Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images. *Comput Biol Med* 2021; 132: 104319.
15. Khan LU, Alsenwi M, Yaqoob I, et al. Resource optimized federated learning-enabled cognitive internet of things for smart industries. *IEEE Access* 2020; 8: 168854.
16. Hao M, Li H, Luo X, et al. Efficient and privacy-enhanced federated learning for industrial artificial intelligence. *IEEE Trans Ind Inform* 2019; 16: 6532–6542.
17. Khan LU, Pandey SR, Tran NH, et al. Federated learning for edge networks: Resource optimization and incentive mechanism. *IEEE Commun Mag* 2020; 58: 88–93.
18. Shi D, Li L, Chen R, et al. Towards energy efficient federated learning over 5G+ mobile devices. *IEEE Wirel Commun* 2022; 29: 44–51.
19. Sigg S, Ma J et al. Camouflage learning. In: *IEEE international conference on pervasive computing and communications workshops and other affiliated events (PerCom Workshops)*, Kassel, Germany, 22–26 March 2021, pp. 724–729. IEEE.
20. Candanedo LM and Feldheim V. Accurate occupancy detection of an office room from light, temperature, humidity and CO_2 measurements using statistical learning models. *Energy Build* 2016; 112: 28–39.
21. He C, Annavaram M and Avestimehr S. Group knowledge transfer: Federated learning of large cnns at the edge. *Adv Neural Inf Process Syst* 2020; 33: 14068–14080.
22. He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, Las Vegas, NV, USA, 27–30 June 2016, pp. 770–778. New York, NY, USA: IEEE.
23. Krizhevsky A, Hinton G et al. *Learning multiple layers of features from tiny images*. Technical Report, Department of Computer Science, University of Toronto, ON, Canada, 2009.
24. McMahan B, Moore E, Ramage D, et al. Communication-efficient learning of deep networks from decentralized data. In: *Artificial intelligence and statistics*. Ft. Lauderdale, FL, USA: PMLR, April 20–22, 2017, pp. 1273–1282.
25. Zheng Q, Chen S, Long Q, et al. Federated f-differential privacy. In: *International conference on artificial intelligence and statistics*, April 13, 2021–April 15, 2021, pp. 2251–2259. San Diego, CA, USA: PMLR.
26. Baheti P, Sikka M, Arya K, et al. Federated learning on distributed medical records for detection of lung nodules. In: *VISIGRAPP (4: VISAPP)*, Valletta, Malta: 27–29 February 2020, pp. 445–451. Setúbal, Portugal: Science and Technology Publications.
27. Grama M, Musat M, Muñoz-González L, et al. Robust aggregation for adaptive privacy preserving federated learning in healthcare. *arXiv preprint arXiv:2009.08294*, 2020.
28. Truex S, Baracaldo N, Anwar A, et al. A hybrid approach to privacy-preserving federated learning. In: *Proceedings of the 12th ACM workshop on artificial intelligence and security*, London, UK, 15 November 2019, pp. 1–11. New York, NY, USA: ACM.
29. Zhang J, Chen C, Li B, et al. A practical data-free approach to one-shot federated learning with heterogeneity. *arXiv preprint arXiv:2112.12371*, 2021.