Cluster Computing
https://doi.org/10.1007/s10586-024-04466-8

Multicriteria client selection model using class topper optimization based optimal federated learning for healthcare informatics

Mamta Narwaria[1] · Shruti Jaiswal[1]

Received: 9 November 2023 / Revised: 19 February 2024 / Accepted: 26 March 2024

Abstract

Quality of life (QoL) of patients has grown as a result of thoughtful medical care systems where many stakeholders remotely review records. Data privacy is highly at risk due to open communication channels, which also has an impact on how models are trained using centralized servers' acquired data. An emerging idea called federated learning (FL) provides a workable remedy to this problem. There hasn't been a comprehensive or in-depth study of FL in the field of health informatics (HI), in contrast to previous studies that mainly focused on the role of FL in diverse applications. In this proposed approach, a Class Topper Optimization (CTO) based federated learning approach is developed. Clinical data's uploaded by clients are taken as input for this proposed work. Stratified sampling is employed to select clients according to their metadata, preventing contacts with clients that aren't relevant. In this paper, clients are selected based on the CTO approach utilizing a variety of criteria's. The server then receives the newly created parameters from each selected clients, which is then utilized for the training process of the local model. Two different algorithms named as Artificial Neural Network (ANN) and Convolutional Neural Network (CNN) are utilized as a local model to train the homogeneous client. The global model is further improved periodically by utilizing the updates from the locally trained instances. Long Short Term Memory (LSTM) is employed as a global model here. The proposed approach achieves 93% accuracy and 92% precision. Thus, the proposed optimization based client selection approach is the best choice for federated learning.

Keywords Federated learning · Artificial neural network · Class topper optimization · Client selection · Healthcare system

1 Introduction

Internet of Medical Things (IoMT) revolution improved the quality of services while bringing about significant changes in how medical institutions run. To collect healthcare data, IoMT devices that can sense and transmit a person's health updates are widely used [1]. Artificial intelligence then processes this data to produce a variety of healthcare applications, including remote patient monitoring and illness diagnosis. In biological image analysis, for example, Deep Learning (DL) algorithms have shown promise in enabling early diagnosis of acute diseases by managing vast quantities of critical data to advance healthcare efficiency. [2].

✉ Mamta Narwaria
mamta2410@gmail.com

Shruti Jaiswal
dce.shruti@gmail.com

1 Department of CSE&IT, Jaypee Institute of Information Technology, Noida, India

Medical data, diagnostic tools, and models must be exposed to a wide range of cases and data that span all possible anatomies in order for machine learning (ML) based healthcare solutions to uncover more useful trends. [3].The technology, demographics, and acquisition methodology can all considerably skew data from a single source, as is well documented. A model's prediction performance would be biased toward the population if it were trained on data from a single source [4]. This consumes more amount of duration and money toward computing, too. A few of these problems can be reduced by parallelizing model training and training in small batches. The privacy of data may not always be preserved even though this solution tackles computation challenges. Studies using a lot of data gathered from many sources are frequently part of clinical research. Medical data, diagnostic tools, models need to be confronted with a variety of instances and data that represent all potential anatomies. In order for machine learning (ML) based healthcare solutions to uncover more informative trends. [5]. Despite the nuanced nature of health information, ethical and legal worries about security and privacy issues pertaining to healthcare data have significantly increased recently.

A ground-breaking, widely used collaborative AI paradigm called federated learning (FL) has particular potential for smart healthcare while allowing numerous customers, including hospitals, to participate in AI training while respecting data privacy. Therefore, the authors conducted a detailed investigation of FL's potential in smart healthcare [6]. In order to integrate FL in smart healthcare, its foundations and most current advancements are first explored. The authors have provided an up-to-date review of FL's rising applications in key fields of healthcare, such as COVID-19 identification, medical data recording, remote patient monitoring, and biomedical image processing [7]. A recent study revealed potential uses for FL in e-healthcare, intelligent transportation systems, and other IoT applications. FL, for example, has increased the use of e-health services despite a shortage of healthcare data. Instead, medical professionals might train the algorithm locally and then send the parameters to the accumulator to compile the data. FL has presented itself as a viable option for creating cost-effective, cutting-edge healthcare systems while safeguarding privacy [8]. Despite the lack of local data, FL enables the training of AI models by merging local updates from numerous healthcare institutions and smart devices such as IoMT.

On the other hand, because of the swift advancement of AI technology, it has been applied in many other fields, including natural language processing, robotics, the Internet of Things, machine vision, and more [9]. To boost the effectiveness of the healthcare sector, researchers have specifically tried to employ AI to help research and assess innovative treatments. As a result of recent advances in the field, much research has been conducted to investigate FL-based AI-related concerns, including healthcare [10]. The most recent FL innovations, such as resource-aware FL, harmless and dependable FL, the possibility of confidentiality-enhancing FL, stimulus-aware FL, and tailored FL, have received insufficient research [11]. Despite these efforts, to our knowledge no published research provides an in-depth analysis of FL-based artificial intelligence applications in the domain of intelligent healthcare. A detailed taxonomy of FL's use in cutting-edge medical applications is likewise missing from the currently available literature [12]. The authors' analysis of FL-based AI applications in the healthcare sector was prompted by these shortcomings. For use in healthcare applications, this suggested approach proposes class topper optimization-based multi-criteria client selection.

To summarize, the following are the primary achievements of this work:

- Data privacy of healthcare system is enhanced using federated learning through optimal client selection approach.
- Stratified samplingbased client filtering is used to pick homogeneous groups of clients rather than a random sample in order to prevent communicating with irrelevant clients.
- Class Topper Optimization (CTO) methodology is used to choose clients based on a variety of parameters, including CPU, time, resource, memory, and energy.
- This proposed method involves training a local model made up of homogeneous clients on Artificial Neural Networks (ANN) and Convolutional Neural Networks (CNN).
- Long Short Term Memory (LSTM) is employed as a global model, which uses the update from locally trained instances.

The following sections of this paper are structured as follows. The Sect. 2 describes the motivation for current research. Section 3 describes some of the existing work reported on client selection in federated learning. The proposed approach and architecture of the proposed methods are described in Sect. 4. Outputs and performance measures of the proposed framework are explained in Sect. 5. The conclusion part is given in Sect. 6.

2 Motivation of the work

In case of a healthcare system, the sensitivity of the wealth of data created by the devices prevents users from saving their information in a centralized location. If all created data was collected, more sophisticated algorithms may have been modeled. As a substitute to centrally managed systems, which may restrict data from being retained in a central repository due to its privacy and/or quantity, federated learning (FL) is currently an innovative solution tackling both confidentiality and shared learning. It is successful in retaining training data on the devices while exchanging locally calculated, then globally aggregated models across several communication cycles. The selection of clients engaging with the FL method is at present complete/quasi-random. However, due to the variety of the client devices in the Internet-of-Things ecosystem, as well as their limited communication and processing assets the training process may not be completed, resulting in numerous wasted learning rounds that affect the accuracy of the model. To tackle these issues, this current research proposed an optimal client selection model using CTO approach based on resource constraints.

3 Literature review

A technique called federated learning allows for the development of machine learning (ML) models that span datasets spread across several data centres, including hospitals, clinical trial labs, and mobile devices, all while limiting data loss. This essay looks into a few instances and uses of federated learning in the healthcare industry.

Qayyum et al. [13] suggested a developing notion of clustered federated learning (CFL) for an automatic COVID-19 diagnosis. By assessing the potential of intelligent clinical data processing at the edge this makes use of edge computing's medical applications. Promising findings are found on both datasets, yielding results that are equivalent to the central baseline where the specialized models were used. However, this approach has a high computation cost.

A blockchain empowered FL framework for healthcare-based CPSs was suggested by Liu et al. [14]. The task agreement committee, which was made up of the representatives of the hospitals that carry out FL tasks, kept a distributed ledger. To produce reliable blocks, it was suggested to use a protected FL task model training-based consensus procedure. Authenticity of FL model aggregation and effectiveness of offering incentives for FL participants. But this techniques has a high processing time.

Silva et al. [15] presented an open-source frontend structure for federated education with medical industry applications. Several projects now offer federated learning frameworks, but they are geared to certain hardware and modeling methodologies and do not include an easily deployable, production-ready environment. It also visualizes the process aspects involved in the installation of learning models, as well as the software components for clients and the central node. This method made the link between genetics and imaging impossible.

A deep FL framework for healthcare data monitoring and analysis using IoT devices was suggested by Elayan et al. [16]. Additionally, it proposed a FL method that handles the procedure for acquiring local training data. It also includes an experiment to test the suggested framework's ability to identify skin conditions. The massive data collection reveals that the DFL models are capable of protecting data privacy without disclosing it. But, the performance of this approach was low.

Lu et al. [17] developed a FedAP to address domain transitions and subsequently provide individualized models for local clients. FedAP learns the similarities and differences between clients based on group normalization layer statistics while preserving each client's unique traits through separate local batch normalization. Comprehensive tests employing five healthcare criteria reveal that FedAP is more accurate. However, this approach did not reveal client similarities.

Sawsan AbdulRahman et al. [18], developed a multi criteria client selection algorithm in federated learning based on optimal approach in IOT. For every client resources CPU, Energy, memory and time was considered to analysis whether they has the ability to perform FL task. Developed approach was named as FedMCCS and it helps in maximising the number of round and train the model effectively and finally transfer te needed updates.

H. Brendan McMahan et al. [19], developed a federated learning approach for effective communication using decentralized data. As the huge data seems to be sensitive, large in quantity which may be complex logging to data server as well as training using traditional approaches. To overcome these drawbacks federated learning approach in which training data was distributed across mobile network and further leanns a shared model through aggregating updates computed locally. Developed model was termed as VanillaFL.

Takayuki Nishio et al. [20], developed a federated learning framework based on client selection for improving communication and to reduce network traffic in mobile edge. Developed federated learning framework was named as Fedcs. This developed federated learning approach solves the problem of client selection by means of framing resources constraints. Training process can be done effectively using this developed Fedcs and further the server get aggregate as many client and finally the performance of ML model get enhanced. Table 1 illustates the existing work on federated learning in healthcare application.

According to the above review numerous approaches are developed for effective federated learning. Numerous substantial obstacles are encountered in those reviewed articles. High computation cost [11], high processing time [12], imaging-genetics relationship was not possible [13], low performance [14] and similarities among the clients were not obtained [15]. In order to overcome these limitations, the proposed approach develops a class topper optimization based client selection approach for optimal federated learning technique.

4 Proposed methodology

The application of dynamic Internet of Medical Things (IoMT) systems with a variety of technologies and machine learning capabilities has been growing significantly in real practice in recent years. MI is crucial to the IoMT system since it helps to balance the burden between delay and energy. The exploitation of the distributed IoMT system's data by conventional learning models for healthcare applications, nevertheless, is still a serious research

Table 1 Existing work on federated learning in healthcare application

Author name	Techniques	Performance	Drawbacks
Qayyum et al. [13]	Clustered federated learning for automated COVID-19 diagnosis	Precision = 86%, recall = 80% and F1 score = 76	Resources heterogeneity is not considered at client level
Liu et al. [14]	Blockchain enabled federated learning for cyber physical system in healthcare	Accuracy = 85%, precision = 85%, F1 score = 88%	In some cases the centeralized FL model may not aggregate the global model
Silva et al. [15]	Decentralized federated learning framework	Accuracy = 90% MSE = 0.2	Hwever these approaches are focused to specified hardware and modelling
Elayan et al. [16]	Deep federated learning for healthcare monitoring usig IOT devices	Accuracy = 97%, precision = 96% recall = 96%	Simultaneous maintain of accuracy, privacy and sustainability is complex
Lu et al. [17]	Adaptive federated learning for improving data privacy in healthcare	Accuracy = 92%, precision = 92% MSE = 0.6	Calculation and updation of similarity among clients is not achieved
Sawsan AbdulRahman et al. [18],	FedMCCS based on resource constraints	Accuracy = 80% Network traffic = 7.6 Kb	Efficiency is not achieved in update of each client
H. Brendan McMahan et al. [19],	Deep network based federated learning for efficient communication	Accuracy = 90%, loss = 1.2	Communication round is not minimized
Takayuki Nishio et al. [20],	FedCS to reduce network traffic in cellular network	Accuracy = 90%	Updating and uploading may fluctuate dynamically

problem. For federated learning, an optimization-based multi-criteria client selection model is developed. Figure 1 presents the suggested approach architecture.

Various clinical data's uploaded by different clients are taken as input for this proposed work. When choosing a homogenous set of customers as opposed to a random assortment, we employ client filtering. Here, stratified sampling is used to choose customers based on their metadata, preventing contacts with clients that aren't relevant. The server then approaches the filtered clients for resource information. Client selection strategies based on class topper optimization (CTO) is used in this case. The global model parameters are distributed to the chosen clients by the server. The local model is then trained using the local data from each selected client, and the newly produced attributes are subsequently shared with the server. In this suggested technique, the homogeneous client is trained using a local model made of Artificial Neural Networks (ANN) and Convolutional Neural Networks (CNN). The global model is then enhanced periodically using the updates from the locally trained instances. As a global model, It uses Long Short Term Memory (LSTM). If inadequate data arrive and an extension is seen, the server terminates the round.

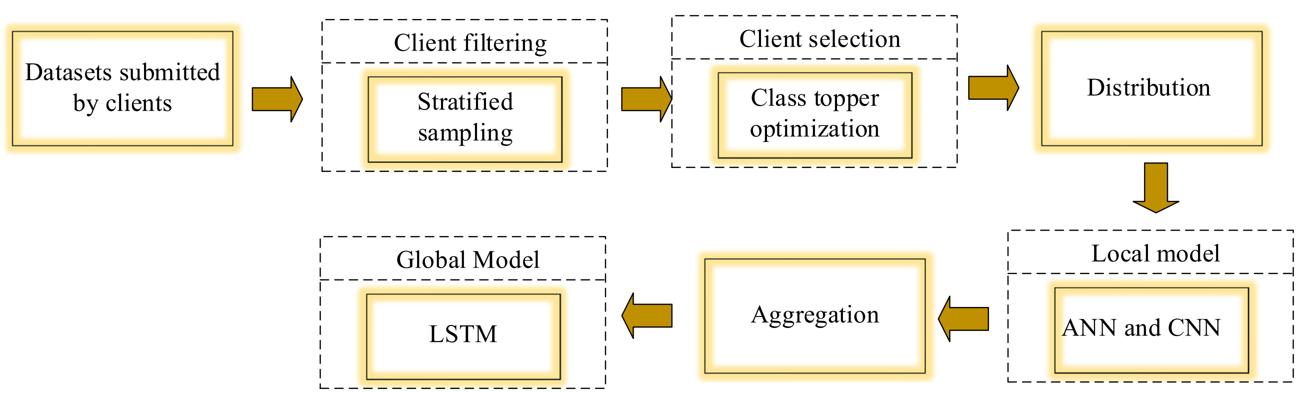


Fig. 1 Proposed CTO based multi criteria client selection FL Approach Workflow

4.1 Stratified sampling for client filtering

A method of random selection known as stratified sampling includes breaking the entire population down into smaller groups and selecting samples at random from each stratum. It is possible to replicate the heterogeneity of the entire population in the sample since the strata are generated based on the population's homogeneous traits. The fundamental principle of stratified sampling is the division of the heterogeneous sample into sampler groups, or strata so that the selection groups are homogeneous in terms of the target qualities of the strata and variable in terms of target characteristics between strata. Following that, observations from each stratum were chosen at random in accordance with how that strata's goal attributes were distributed in the initial data [21].

The server queries these filtered clients for information about their resources (such as how much data they have, historical data from previous training jobs, etc.).

4.1.1 Class topper optimization based client selection

It is the main module in our system. In order to aggregate their updates, it enables choosing the greatest number of capable and available clients every round. The suggested selection algorithm makes use of a variety of factors, including time, CPU, memory, and energy, to achieve the maximizing objective.

4.1.2 Class topper optimization

A meta-heuristic optimization algorithm known as the Class Topper Optimization (CTO) algorithm is developed in opposition to an inspiration source based on human intelligence. The educational endeavour that pupils in a section engage in in order to excel intellectually or take charge of the class is known as the knowledge origin. Each evaluation/examination step is an attempt by the students to do better. The algorithm tracks a student's ascent to excellence in information enhancement. A population-based algorithm is used. The best student in the class can serve as a model for other students. On the worldwide test, one of the students ended up being the class leader, according to that student information enhancement component. The foremost objective is to improve the performance of the class leader. The CTO technique offers two different knowledge enhancements, including knowledge enhancements at the section and student levels. Not all of the pupils in the section might experience that. Only a small number of those kids will actually learn anything from the top students in the group. The opportunity for the pupils to learn from the class leader will be lost [22]. Computationally, a student provides the following knowledge: Each student in each part learns the following with their appropriate ST:

$$L^{T+1} = W \times L^{T} + C \times \varphi_2 \times (ST - CT) \tag{1}$$

$$S^{T+1} = S^{T} + L^{T+1} \tag{2}$$

A section's top student then becomes the section's best student. CTO imparts knowledge to the section topper (ST).

$$L_1^{T+1} = W \times L_1^{T} + C \times \varphi_1 \times (CT - ST) \tag{3}$$

$$ST^{T+1} = ST^{T} + L_1^{T+1} \tag{4}$$

The portion is made better by the student's learning. In the algorithm, some constants like W, C, φ_1, φ_2 will be decided in advance. W is the international eight factor, which maintains the balance between local and worldwide investigation. The word C represents the stochastic acceleration that pushes each student's cause in the direction of the performance index (PI). W is characterized as:

$$W^{E} = W_{max} - \left(\frac{W_{max} - W_{min}}{E_{max}} \times E\right) \tag{5}$$

where, W_{min} is the lowest weight factor and W_{max} is the highest weight factor. Examinations are used to assess students' achievement. The ST and CT are selected for each exam, and the improvement factors for all students are computed. This improvement factor denotes increased solution correctness. The end product is the world's best solution. The rate of convergence and the quality of the response must be balanced. The trade-off between the two determines the number of sections and students.

4.1.3 CTO for the proposed method

Step 1: Initialization

Initialize the set of clients involved in this process. The initialization procedure is given below.

$$X = X_1, X_2, \ldots, X_n \tag{6}$$

Step 2: Fitness function

The fitness function of the proposed method is estimated using the resulting Eq. (7). The fitness is calculated based on three different constraints such as energy, time and budget.

$$\begin{cases} \forall X_{f_{z=1}^{i}} \Sigma Util_{r\in\{CPU,Memory,Energy\}}^{X_{fz}} < Budget_r^{X_{fz}} [CO1] \\ \forall X_{f_{z=1}^{i}} \Sigma \left(T_d^{X_{fz}} + Util_{r=T_{ud}}^{X_{fz}} + T_{ul}^{X_{fz}}\right) < T[CO2] \end{cases} \tag{7}$$

$$Max\ ER_{X_{f_{z=1}^{i}}} = \left[\frac{|X_{fz}.l_A|}{|X_{fz}.l_A| + |X_{fz}.l_N|} \times 100\right][C03] \tag{8}$$

where, $Util^{X_{fz}}_{r\in\{CPU,Memory,Energy\}}$ represents the predicted utilization of the resources r for the client X_{fz}, r represents the CPU, memory or energy. $Budget^{X_{fz}}_{r}$ indicates the resource budge per device type.

The first requirement is a strict budget for utilizing resources on each type of device while preventing dropouts. A model's download, update, and upload must not go beyond the specified threshold T, which is the second consideration. The third need is choosing the required number of customers depending on the occasion's rate.

Step 3: Updation

The process gets updated until the best solution is obtained. Equation (3) and Eq. (4) are used to update the fitness function.

Step 4: Termination

The entire client selection process gets terminated after the best solution is obtained.

The server evaluates client replies to choose the most qualified group to take part in the upcoming training sessions.

4.2 CNN and ANN-based Local model

Each chosen client then uses unique local data to train the local model before sharing the newly created parameters with the server. In this suggested technique, a homogenous client comprising an integer and picture dataset is trained using a local model that syndicates an artificial neural network (ANN) with a convolutional neural network (CNN).

4.2.1 Artificial neural network

ANN is used in this paper, which classifies the given image as normal and disease. Networks of artificial neurons resemble the network of neurons in a biological brain. Each synthetic neuron has the ability to signal other neurons. This signal, which is typically a real number, may be processed. Each neuron's output signal is calculated as a nonlinear function of the inputs. An input layer and an output layer make up the core of an ANN's design, with one or more hidden layers between them perhaps included to increase model precision.

Figure 2 shows the ANN's basic architecture. In general, input layers will look for more straightforward patterns, whereas output layers will hunt for more intricate relationships. Every neuron will carry out a weighted sum, W_s, as determined by:

$$W_s = \sum_{i=1}^{n} w_i X_i \tag{9}$$

where w_i is the weights, n is the number of inputs to be processed, and X_i are the signals from neighbouring neurons. The weight values will be optimized by ANN as it learns. Weighted sum, or W_s. ANN will employ a function for activation. One of the most used methods for categorizing photographs is the "relu" activation function, which is defined as:

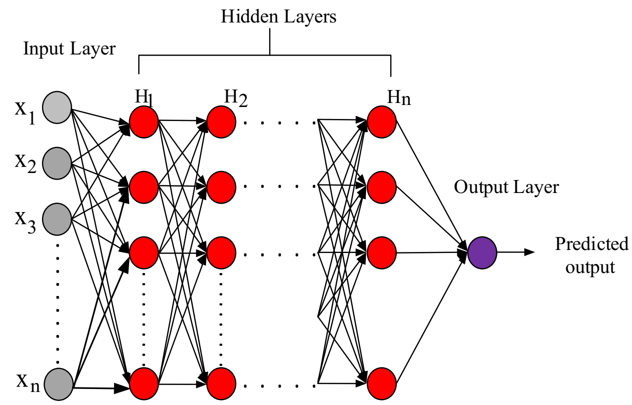


Fig. 2 ANN's basic architecture

$$y = \max(W_s, 0) \tag{10}$$

Which will result in either 0 if W_s has a negative value or the weighted sum itself is a positive integer. The loss function must then be applied to all of the network's weights using a back-propagation method as a subsequent step. A loss function is often computed using the discrepancy between the predicted and actual output values. A loss function is well-illustrated by the mean squared error.

$$C = \frac{1}{2}\sum_{j=2}^{n} (y_j - \overline{y}_j)^2 \tag{11}$$

where, $\overline{y}_j$ is the jth outcome's predicted value. The sparse_categorical_crossentropy loss function is typically used for classification problems with numerous classes and single classes indicated by numbers. The Keras documentation contains additional details on the definition and application of this loss function, among others. The next step after computing the loss function is to determine its minimum and then optimize the weight values. The weights in the ANN are updated using optimization algorithms based on the gradient of the loss function [23]. The Adam optimizer is employed in the present research.

ANN employ initial weight values that are close to zero. The input row of data is the initial row that is processed by the network. The network's prediction is compared to the actual outcome, and during optimization, the weight values are updated using the cost function. Following that, this process is repeated for all data or, in certain situations, just a selection of data. When the training process is complete for all of the observations, an epoch is said to be done. To enhance the accuracy of the forecasts, the entire procedure can then be repeated for different epochs.

4.2.2 ResNext50 based convolutional neural network

Within ResNexr50 architeture, in each non-identity portion of residual block there exist a squeeze and excitation (SE) block. In addition to residual block identity and convolution block is present in five discrete section. Three process of conversion is present in every Identity block and signle convolution block constitue three convolution layers. Using SE block the feature map is generated using the input data. Also, this block acts as a computational unit. Generally CNN is made up of numerous pooling and convolution layer and it also consist of single or more linked layer at each end. Every convolutional layer possess three stages which includes pooling, convolution and nonlinear activation. Feature map is initially created using each convolution later and further it is then sent to the subsequent layer [24]. Mathematical expression for convolution layer is given in Eq. (12)

$$F^n(X) = pooling(F^n(F^{n-1}(X) * w^n + B^n) \quad (12)$$

B^n is used to denote the bias and w^n is used to illustrate the kernel of the layer and feature map for n^{th} layer of convolution is given by $F^n(X)$. In this framework, features are retrieved and classified using CNN (ResNext-50). It is compatible with a variety of CNN designs and legacy networks. The computational cost increases because the SE block arrives preceding summation. ResNext-50 is able to obtain a greater degree of reliability over ResNet-50 owing to it. Figure 3 reveals ResNext-50's overall architecture.

ResNeXt placed second in the Image Net Large Scale Visual Recognition Challenge (ILSVRC), a competition that tests algorithms to improve accuracy on a given data set for picture categorization. Each non-identity segment of a residual block uses the squeeze and excitation (SE) block according with its design. Table 1 illustrates the ResNext-50 using a 32 × 4d template.

It is organized into five major sections, which include blocks for convolution and identity. A single convolution block consists of three layers of convolution and three phases of conversion for each ID. The convolution layer (CONV) uses filters to perform convolution operations on the input data to examine it in terms of its dimensions. The stride and filter size are two convolutional layer hyperparameters. The result is known as a feature map or an activation map. In models of neural networks that estimate a multinomial distribution of probabilities, the fully linked layer (FC) employs a compacted input, which means that each input is tied to each neuron. These models' output layers activate using the softmax function.

The chosen clients update the shared model with their local data and submit the updated model parameters to the server. In order to create an improved model, the server averages the changed parameters.

4.3 LSTM based global model

It is in charge of producing a much enhanced model update utilizing a federated average function. The long short term memory is used as the global model in the suggested method.

4.3.1 LSTM

Recurrent neural networks (RNNs) of the LSTM variety enable the system to preserve long-term links between data from a number of previous time steps at a given moment. It has the shape of a series of repeatedly occurring neural network modules, each of which has three control gates the forget gate, the input gate, and the output gate. Each gate consists of a pointwise operation and a layer of a sigmoid neural network. A percentage of the input data that should

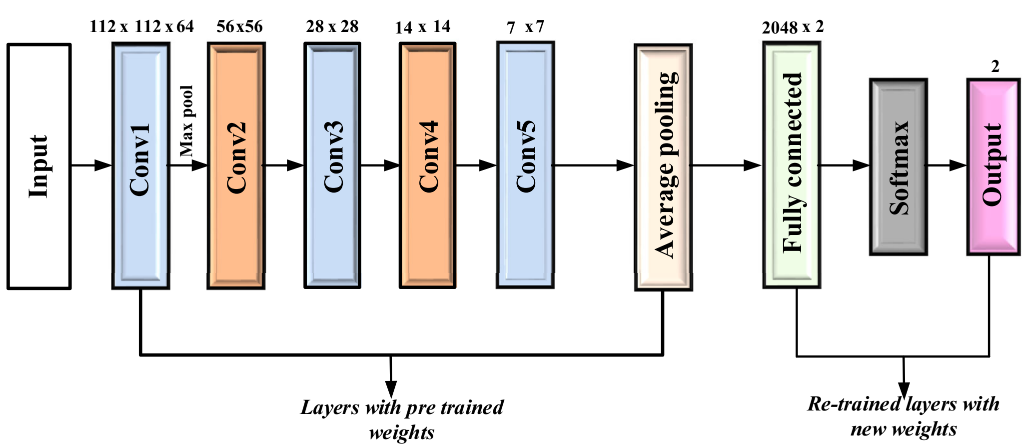


Fig. 3 The pre-trained architecture of ResNeXt-50

be allowed through is represented by the sigmoid layers' output numbers in the range [0. 1]. For time series data, the LSTM receives a succession of input vectors $x = (x_1, x_2, \ldots, x_t, \ldots\}$, where $x_t \in \mathbb{R}^m$ corresponds to an m-dimensional vector of readings for m variables at time occurrence t [25]. Figure 4 shows the Basic LSTM architecture.

The LSTM module functions as follows given the new data x_t in state t. In order to determine whether outdated information should be forgotten, it first outputs a number between [0, 1], say f_t with

$$f_t = \sigma_1\left(W_f.[h_{t-1}, x_t] + b_f\right) \tag{13}$$

where W_f and b_f are the weight matrices and the bias of the gate that forgets, respectively, and h_{t-1} is the output in state *t-1*.Furtherlythe x_t is then processed before being stored in the cell state. In order to update in the new cell state C_t, the value i_t is calculated in the input gate in addition to a vector of candidate values $\widetilde{C}_t$ that is produced by a tanh layer inspite of time instant.

$$i_t = \sigma_2(W_i.[h_{t-1}, x_t] + b_i) \tag{14}$$

$$\widetilde{C}_t = tanh\left(W_c\left[h_{t-1,}x_t\right] + b_c\right) \tag{15}$$

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t \tag{16}$$

where the weight matrices along with biases of the input gate as well as memory cell states, respectively, are (W_i, b_i) and (W_c, b_c) respectively. The output gate is the last one and it is specified by

$$o_t = \sigma_3(W_o.[h_{t-1}, x_t] + b_o) \tag{17}$$

$$h_t = o_t * tanh(C_t) \tag{18}$$

where W_o and b_o represent the weight matrix along with the bias of the output gate, respectively, determine a portion of the outputted cell state. Additionally, many authors have recommended alternative LSTM variations. Popular LSTM versions were directly compared and found to be almost identical, with a few being more effective than others, but only for a limited number of problems.

Numerous systems could crash due to the devices' limited resources. In order to make up for the device dropout, handle 30% of the chosen customers who are not responding in each round while using the settlement approach. The FL round is regarded as being abandoned if over thirty percent of the information is not received. The aggregate is successfully completed in all other respects. The previous procedures are then iterated over till the desired model efficiency is attained. The Pseudo code of the proposed approach is given in Algorithm 1.

Algorithm 1: Pseudocode for Proposed Federated Learning

```
Input: Clinical data's submitted by clients
# Client Filtering
{
Stratified sampling based on metadata
}
# Client Selection
{
Class Topper Optimization
{
Update clients based on multi-criteria using Eqn. (3) and (4)
}
}
# Local model training
{
Artificial Neural Network for integer data
Convolutional Neural Network for Image Data
}
# Global model
{
Long Short Term Memory
}
```

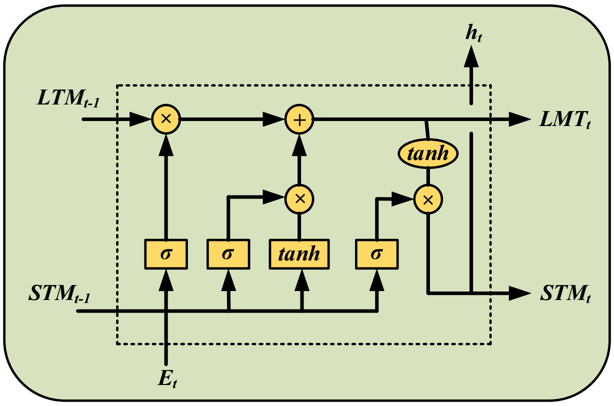


Fig. 4 Basic LSTM architecture

Table 2 ResNext-50 along with a 32 × 4d template

Stage	Output	ResNeXt-50 (32 × 4d)
Conv1	112 × 112	7 × 7, 64, stride 2
		3 × 3 max pool, stride 2
Conv5	7 × 7	$\begin{bmatrix} 1\times 1, & 1024 \\ 3\times 3, & 1024, C=32 \\ 1\times 1, & 2048 \end{bmatrix} \times 3$
Conv4	14 × 14	$\begin{bmatrix} 1\times 1, & 512 \\ 3\times 3, & 512, C=32 \\ 1\times 1, & 1024 \end{bmatrix} \times 6$
Conv3	28 × 28	$\begin{bmatrix} 1\times 1, & 256 \\ 3\times 3, & 256, C=32 \\ 1\times 1, & 512 \end{bmatrix} \times 4$
Conv2	56 × 56	$\begin{bmatrix} 1\times 1, & 128 \\ 3\times 3, & 128, C=32 \\ 1\times 1, & 128 \end{bmatrix} \times 3$
Conv1	112 × 112	7 × 7, 64, stride 2
		3 × 3 max pool, stride 2
	1 × 1	Global average pool 1000-d fc, softmax

Table 3 Simulation parameters of ANN

Parameters	Range
optimizer	adam
activation	relu

Table 4 Simulation parameters of CNN

Parameters	Range
Optimizer	SGD
Activation	Relu
Maximum epoch	50

Table 5 Simulation parameters of LSTM

Parameters	Range
Optimizer	Adam
Epochs	5
Batch_size	2

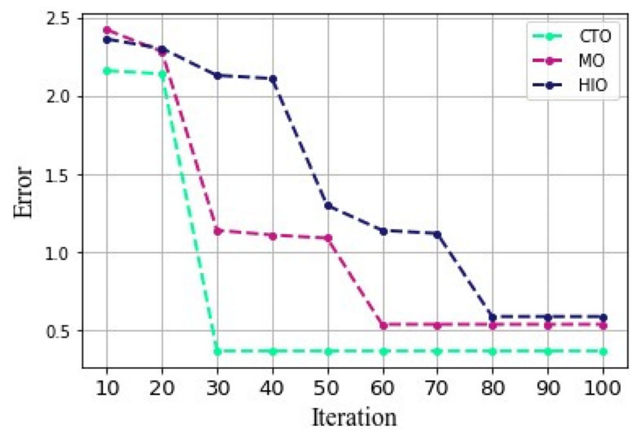


Fig. 5 Convergence plot for CTO optimization approach

5 Result and discussion

Healthcare efficiency, service outcomes, and human well-being could all be improved by means of AI approaches to improve or support healthcare applications. Deep learning and data-driven machine learning have recently achieved success across a wide range of business sectors, including healthcare. However, constructing a deep learning model typically necessitates an extensive set of data specimens, which is not always feasible with sensitive health information. Federated learning is suggested as a new ML paradigm to do this, and it has the potential to produce an international ML model even though it has direct access to each contributor's private data, which may include user, device, or medical facility information. Different healthcare datasets are used as input in the suggested method.

Simulation parameters used for the local model named as ANN and CNN are given in Tables 2 and 3. Likewise, the modelling parameters of a global model called LSTM is represented in Tables 4 and 5.

The input data submitted by users are filtered using stratified sampling to avoid communications with irrelevant clients. Then, the clients are selected using multiple criteria such as time, CPU, resource, memory and energy using Class topper optimization (CTO). A few currently used optimization approaches are compared with the performance of the suggested optimization algorithm.

Convergence plot of proposed and existing optimizations are depicted in Fig. 5. Proposed class topper optimization is compared with the current algorithms called human inspired optimization and mother optimization algorithms. Typically a convergence plot is produced through iteration with regard to the fitness function. The fitness function is represented as an error. Based on this

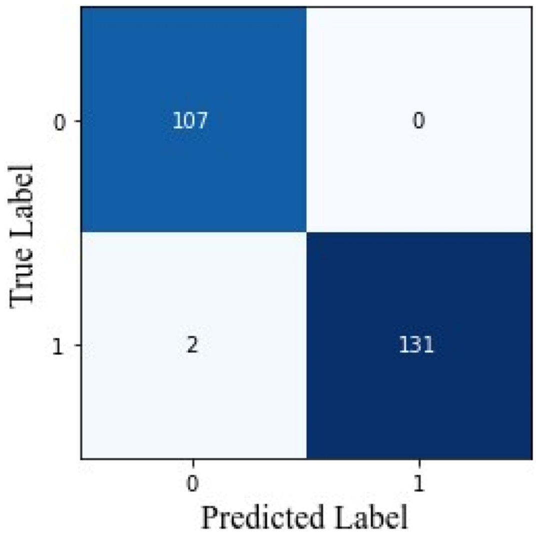


Fig. 6 Confusion matrix plot

data, the CTO method outperforms other optimization algorithms since it converges at early iteration. Each chosen client trains a local model, such as an ANN and CNN, using local data after client selection and then communicates the newly created features with the server. The global model known as LSTM is then enhanced by aggregating modifications from locally trained instances on a regular basis.

The confusion matrix plot for the overall dataset is given in Fig. 6. Two different classes, such as disease affected and non-affected are considered here. It is obtained using predicted and true label. To explain how effectively a categorization algorithm system works, a confusion matrix is used. This shows and summarises how well the categorization algorithm performed. This graphic indicates that just a small number of samples from each class are incorrectly predicted, resulting in minimum error and

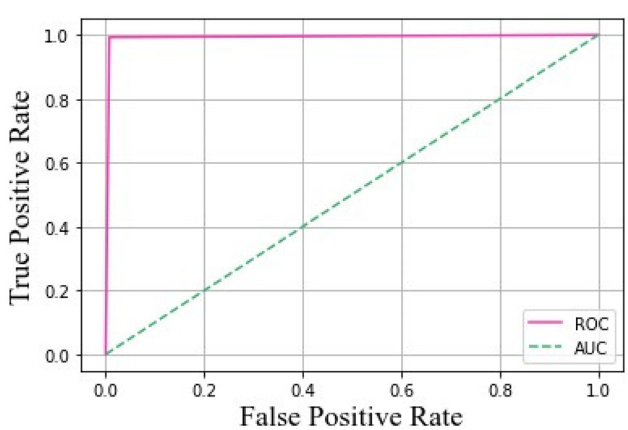


Fig. 7 AUC and ROC plot

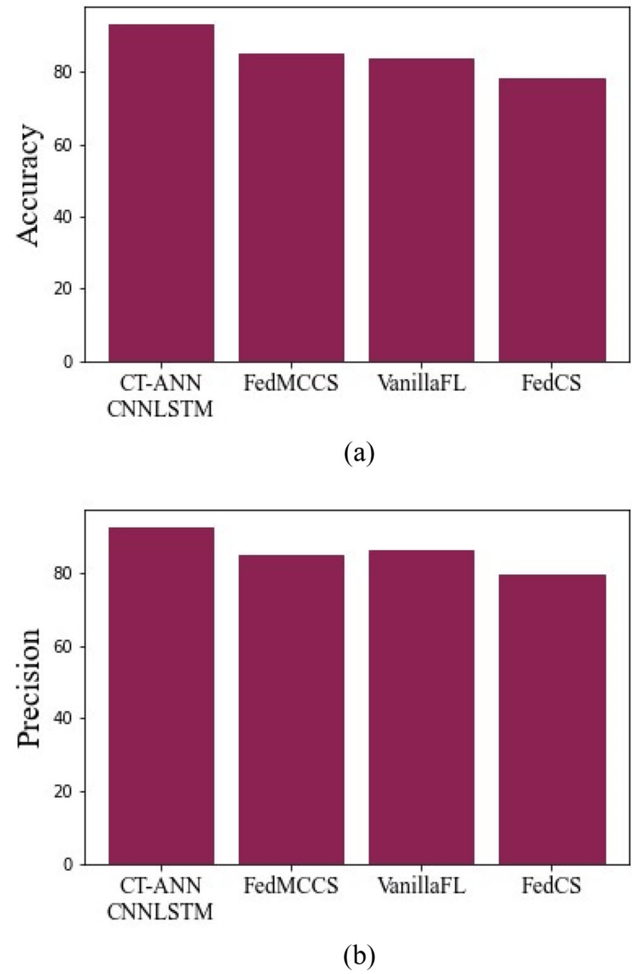


Fig. 8 Comparison of **a** accuracy and **b** precision

higher accuracy when utilizing the proposed federated technique.

The proposed federated learning algorithms AUC and ROC plot are depicted in Fig. 7. The Area Under Curve (AUC)-Receiver Operating Curve (ROC) is an effectiveness measure for categorization difficulties with varied threshold levels. It illustrates the model's capacity for class based differentiation.

The proposed Class Topper Optimization based federated learning approach is compared with some existing federated algorithms called FedMCC, VanillaFL and FedCS approaches. Some of the metrics utilized for assessing proposed and existing federated learning algorithms include accuracy, precision, training time, and testing time. The graphical representation of these comparison is given below.

Figure 8 shows a comparison of proposed and existing techniques' accuracy metrics. (a). 93% accuracy rate is attained using proposedCT-ANN-CNN-LSTM. However, the accuracy of current algorithms like FedMCC has a value of 85%, VanillaFL has a value of 83%, and FedCS is

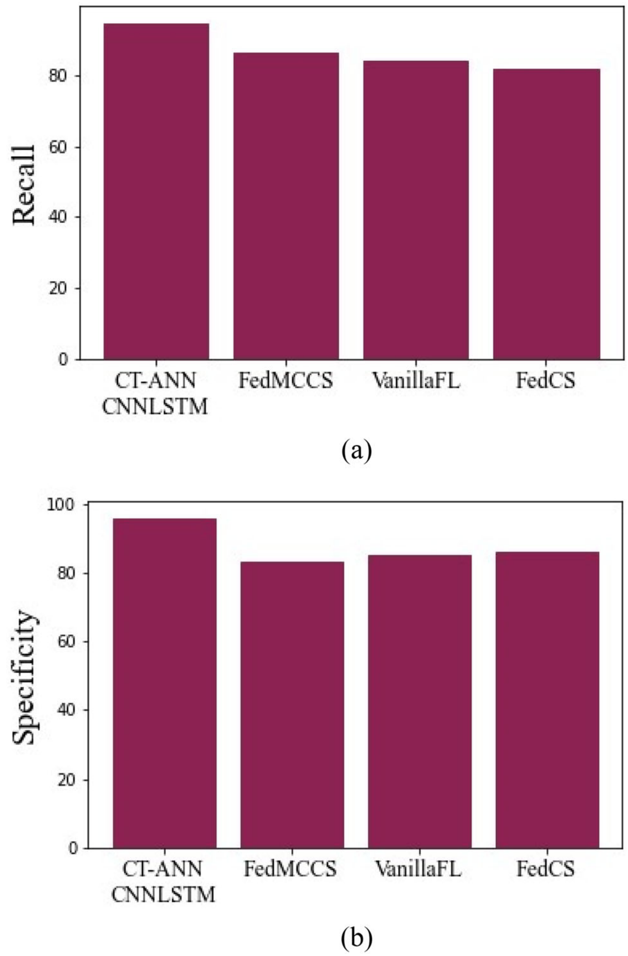


Fig. 9 **a** Recall, **b** specificity metric evaluation

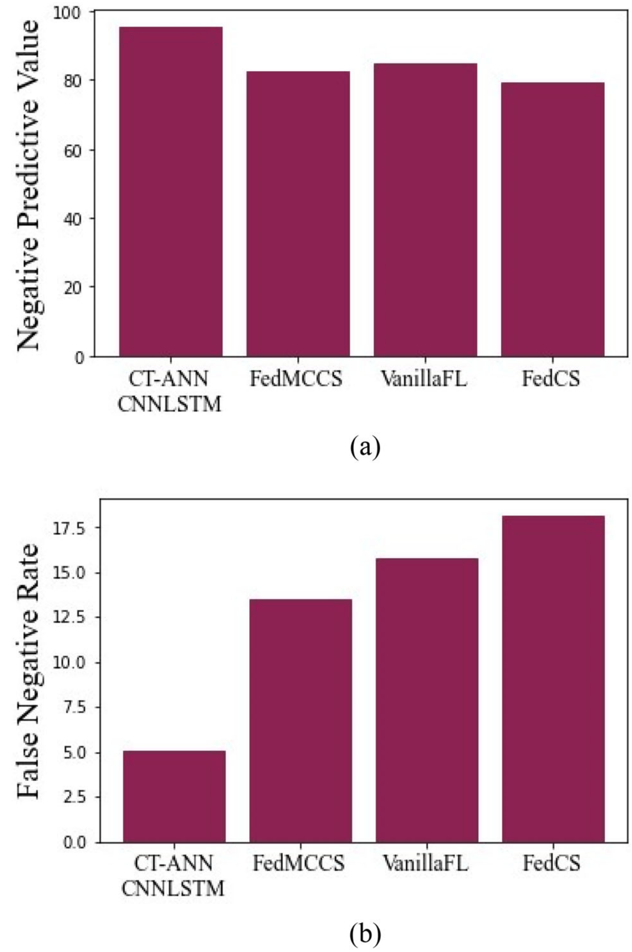


Fig. 10 Analysis of **a** NPV, **b** FNR

78%, correspondingly. When compared to the suggested procedure, these are low. The precision metric comparisons of the proposed and existing strategies is shown in Fig. 8b. The suggested CT-ANNCNNLSTM holds the value of 92%, existing FedMCC has the value of 84%, VanillaFL holds the value of 86%, and FedCS algorithms' precision value results in 79%, respectively. In comparison to other approaches, it can be observed that CT-ANNCNNLSTM has a high precision value.

Figure 9a shows how recall measures were used to compare the proposed CT-ANNCNNLSTM and existing approaches. The recall value acquired using the proposed strategy was 94%, which is greater than the recall value achieved through previous methods. Because the recall values of current approaches like FedMCC, VanillaFL, and FedCS are 86%, 84%, and 81%, respectively. A similar comparison of specificity measurements is shown in Fig. 9b. The specificity value of the proposed CT-ANNCNNLSTM is 98%. However, the specificity values for current methods for FedMCC, VanillaFL, and FedCS are 96%, 83%, 85%, and 86%, respectively. The proposed method has been shown to perform superior in regards to specificity and recall.

A comparison of proposed and existing algorithms in terms of Negative Predictive Value is illustrated in Fig. 10a. 95%, 82%, 84% and 79% are the NPV produced by CT-ANNCNNLSTM, FedMCC, VanillaFL and FedCS classifiers. False Negative Rate analysis is explained in Fig. 10 (b). FNR values obtained are 5%, 13%, 15% and 18% for CT-ANNCNNLSTM, FedMCC, VanillaFL and FedCS respectively. Among these algorithms proposed approach has a low FNR.

Figure 11a provides an assessment of the False Positive Rate. A variety of classifiers, including CT-ANNCNNLSTM, FedMCC, VanillaFL, and FedCS, were used to analyze the FPR statistic; their respective FPR values were 3%, 16%, 14%, and 13%. This indicates that the proposed hybrid classifier produces low FPR. Similarly, Fig. 11b represents the F1_Score metric comparison. 93%, 85.5%, 85%, and 80% are the F1_Score value obtained by CT-ANNCNNLSTM, FedMCC, VanillaFL and FedCS.

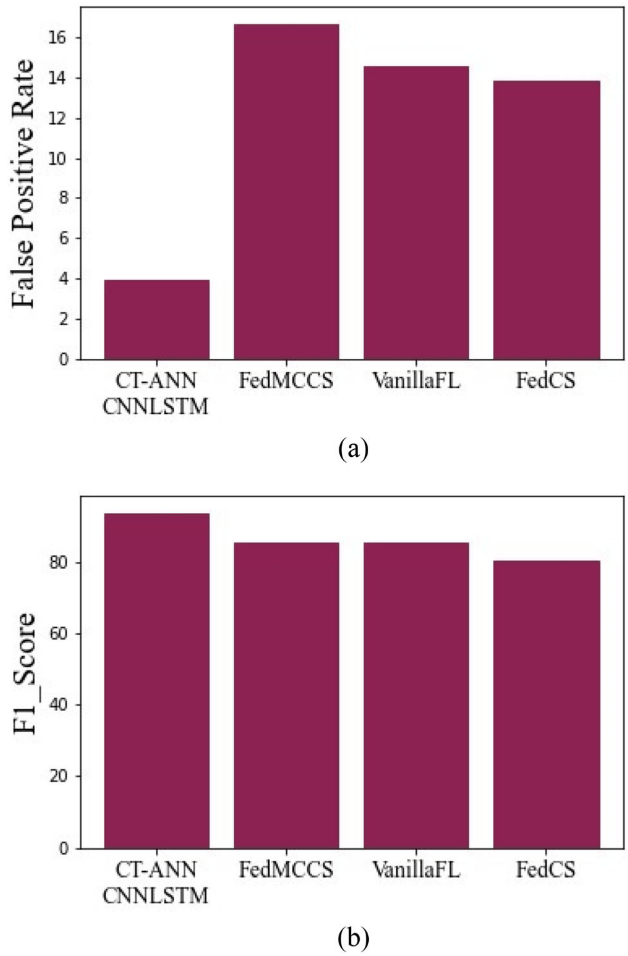


Fig. 11 **a** FPR, **b** F1_Score examination

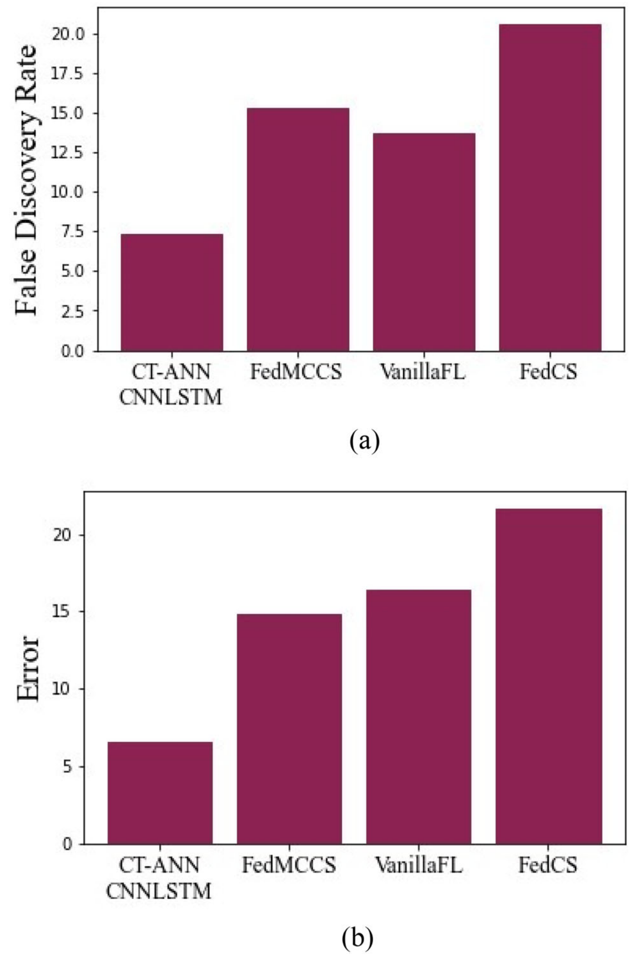


Fig. 12 Comparison of **a** FDR, **b** error

Figure 12a illustrates the False Discovery Rate (FDR) examination of proposed and existing algorithms. FDR values produced by FedMCC, VanillaFL and FedCS classifiers using are 7%, 15%, 13% and 20%. The proposed CT-ANNCNNLSTM approach has better performance. Figure 12b displays an evaluation utilizing error metrics between suggested and current strategies. Existing approaches CT-ANNCNNLSTM, FedMCC, VanillaFL and FedCS give error values of 6%, 14%, 16%, and 21%, respectively. But, the proposed model has 3% error value.

Figure 13a shows the Mathews Correlation Coefficient (MCC) evaluation. MCC metrics gained by the suggested model is 91%. But the MCC value of the existing FedMCC, VanillaFL and FedCS techniques are 88%, 85% and 78%, respectively. It demonstrates how the suggested strategy outperforms current procedures. Kappa metric evaluation of suggested and current approaches is depicted in Fig. 13b. The Kappa value of the suggested model is 89%. But the existing techniques, such as FedMCC, VanillaFL and FedCS has kappa value are 87%, 83% and 78%, respectively.

Figure 14a compares suggested and current techniques using the False Omission Rate (FOR) statistic. The FOR rate attained by the suggested CT-ANN-CNN-LSTM is 5%. However, current algorithms like FedMCC, VanillaFL, and FedCS have respective FORs of 18%, 16%, and 21%. When compared to the suggested procedure, these are high. Similar to Fig. 14a, b compares proposed and existing strategies using the MK metric. The MK values proposed by CT-ANNCNNLSTM and the existing FedMCC, VanillaFL, and FedCS algorithms, respectively, are 91%, 83%, 81%, and 76%. Comparing CT-ANNCNNLSTM to other techniques, it can be observed that it has MK value.

Evaluation of the proposed CT-ANNCNNLSTM and existing approaches utilizing FM measures is depicted in Fig. 15a. The recall rate attained by the suggested strategy, 91%, is greater than the FM rate attained by previous strategies. Considering that the FM values of current approaches like FedMCC, VanillaFL, and FedCS are respectively 88%, 79%, and 83%. Figure 15b shows a comparison of measures using the Adjusted Rand Score (ARS). An ARS value of 86% is assigned to the proposed

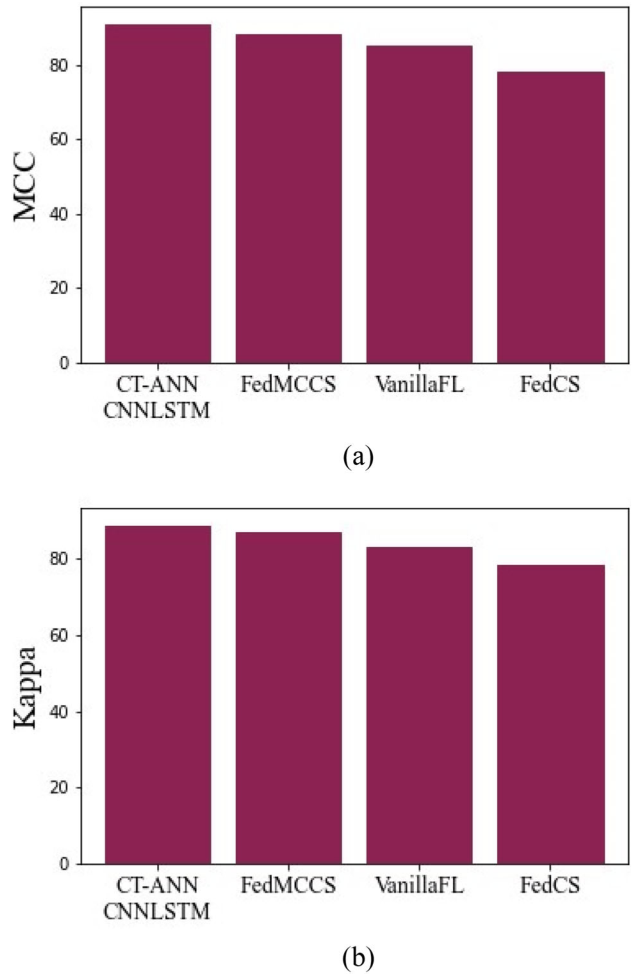


Fig. 13 **a** MCC, **b** Kappa metric evaluation

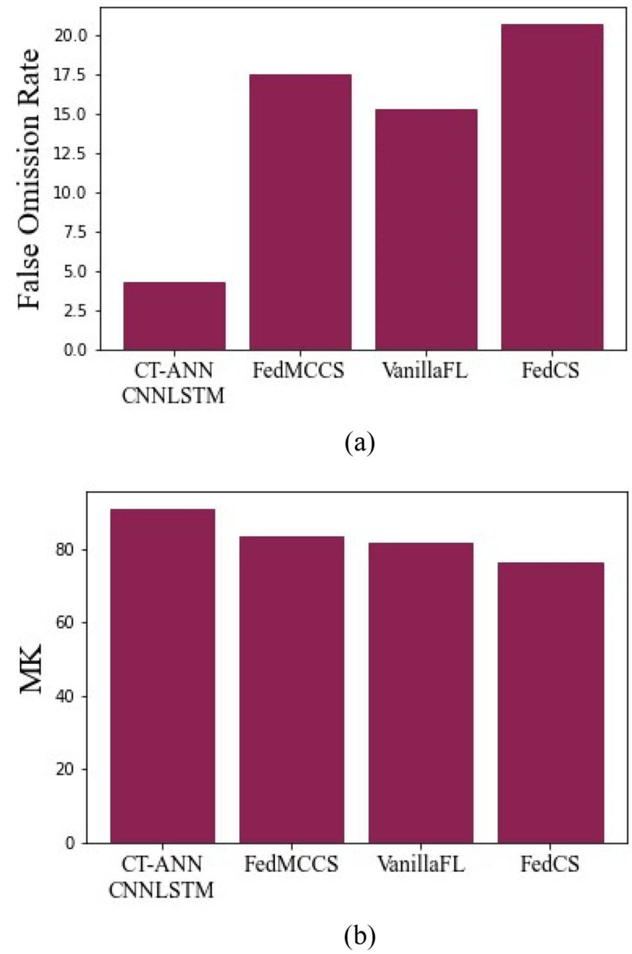


Fig. 14 Analysis of **a** FOR, **b** MK statistic

CT-ANNCNNLSTM. However, for FedMCC, VanillaFL, and FedCS, respectively, the ARS values are 80%, 77%, and 71%. It has been demonstrated that the suggested approach performs better in terms of FM and ARS.

A comparison of proposed and existing algorithms in terms of Adjusted Mutual Info-score (AMIS) is illustrated in Fig. 16a. 85%, 81%, 73% and 76% are the AMIS produced by CT-ANNCNNLSTM, FedMCC, VanillaFL and FedCS classifiers. Figure 16b shows the training time comparison. The period taken for the training process using CT-ANNCNNLSTM is 7.7 s. It is lower than other existing FedMCC, VanillaFL and FedCS approaches, which has a training time of 9.3 s, 10.7 s and 9.7 s. It demonstrates that the suggested strategy performs better.

Testing time analysis of proposed and existing algorithm is depicted in Fig. 17a. The testing times for CT-ANNCNNLSTMm is 0.6 s, FedMCC results in 0.8 s, VanillaFL holds the value of 0.7 s, and FedCS is 0.9 s. It demonstrates that the proposed approach requires less time for testing than existing approaches. It takes 8.3 s for CT-ANNCNNLSTM, 10.1 s for FedMCC, VanillaFL holds the value of 11.5 s, and 10.6 s for FedCS to execute, which is shown in Fig. 17, which depicts the execution of the time analysis of the proposed and current techniques.

Evaluation of proposed and current techniques in terms of training error is depicted in Fig. 18a. It is taken by increasing the epoch number. When the number of epochs increases, the training error get decreased. Likewise, Fig. 18b shows the testing error examination of proposed and existing algorithms. It also decreased with an increased number of nodes.

Training accuracy analysis is illustrated in Fig. 19a. It is taken by changing the number of epochs. When the number of epochs increases, the training accuracy is also get increased. Likewise, Fig. 19b depicts the validation

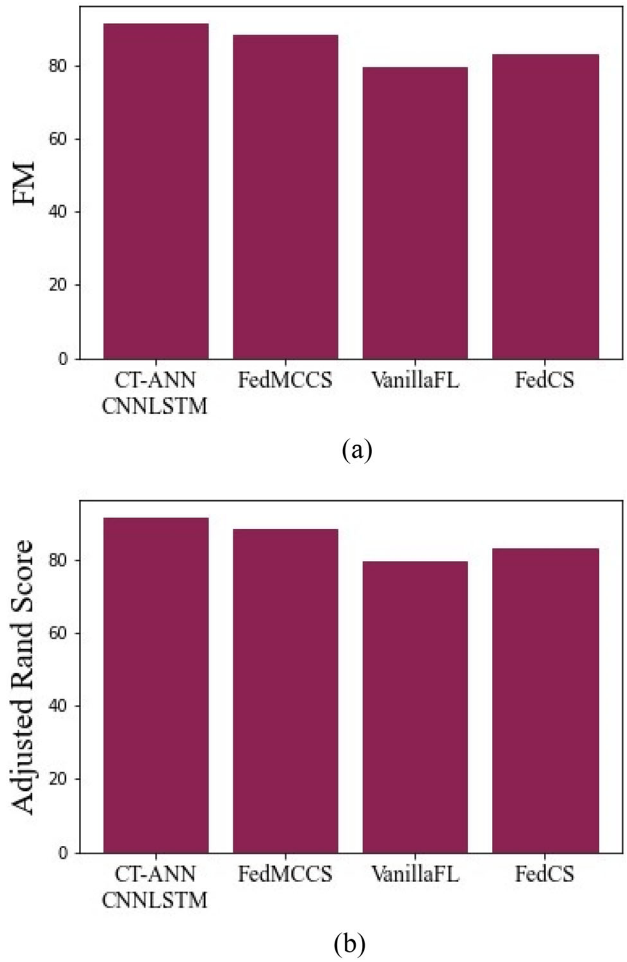


Fig. 15 **a** FM, **b** ARS examination

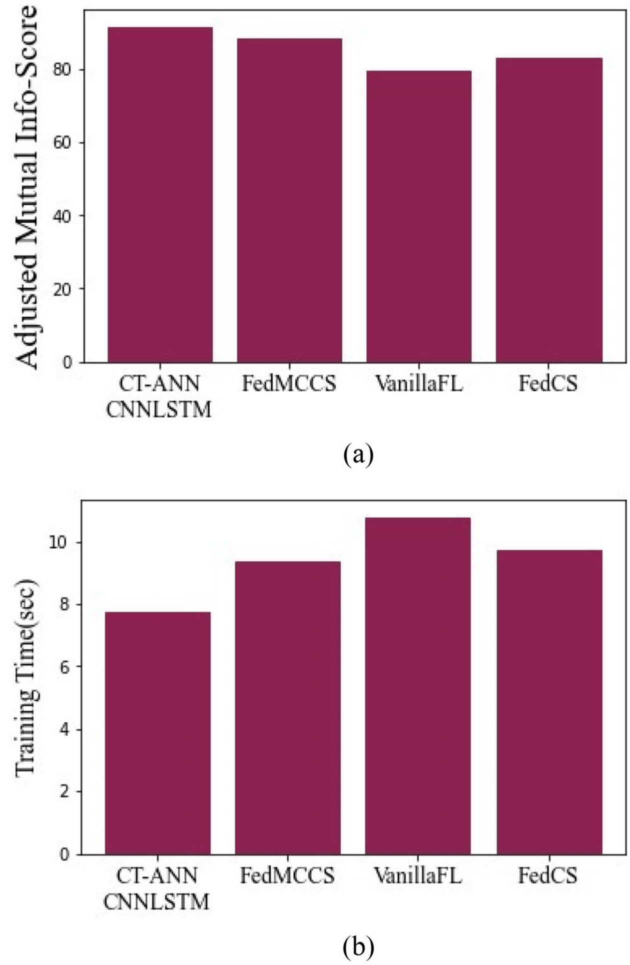


Fig. 16 Comparison of **a** AMIS and **b** training time

accuracy comparison of proposed and existing algorithms. Validation accuracy get increased when the number of epochs increased.

Figure 20 illustrates the preicision recall curve fior proposed model. The precision-recall curve illustrates the relationship among precision versus recalls at different tiers. It is commonly employed in circumstances where classes are severely inconsistent. An ideal model will have great precision and recall and will produce a large number of results that are all properly predicted, whereas a baseline model will have extremely low precision. However the proposed model has high preicision as well as high recall which shows effective performance of the proposed model. Network traffic calculated for proposed and existing federated learning framework is given in Fig. 21. Network traffic achieved for proposed CT-ANNCNN LSTM is 3.8 Kb and existing model such as FedMCCS, VanillaFL and FedCS attains network traffic of 7.6 kb, 5.7 kb and 4.6 kb. Table 6 illustrates the comparison between proposed and existing state-of-the-art methods for federated learning.

6 Conclusion

The incorporation of multiple technologies, such as the Internet of Things (IoT), has led to a substantial development in intelligent applications in dispersed healthcare and medical systems in recent years. A sizable amount of personally identifiable information frequently needs to be safeguarded and kept private in the context of healthcare and clinical studies. Hospitals and researchers will frequently need to keep certain data on hand, which might cause information silos and problems with collaboration. Furthermore, heterogeneous federated learning offers data transmission, replica placement, throughput reduction, resource management, and network load reduction to improve the accuracy, consistency, and service level agreement (SLA) components of public health and medical systems. The major components of federated learning offer a safe and secure architecture to enhance the privacy and effectiveness of heterogeneous information in public health records and medical systems. Federated learning is a platform for distributed intelligence that supports quality of

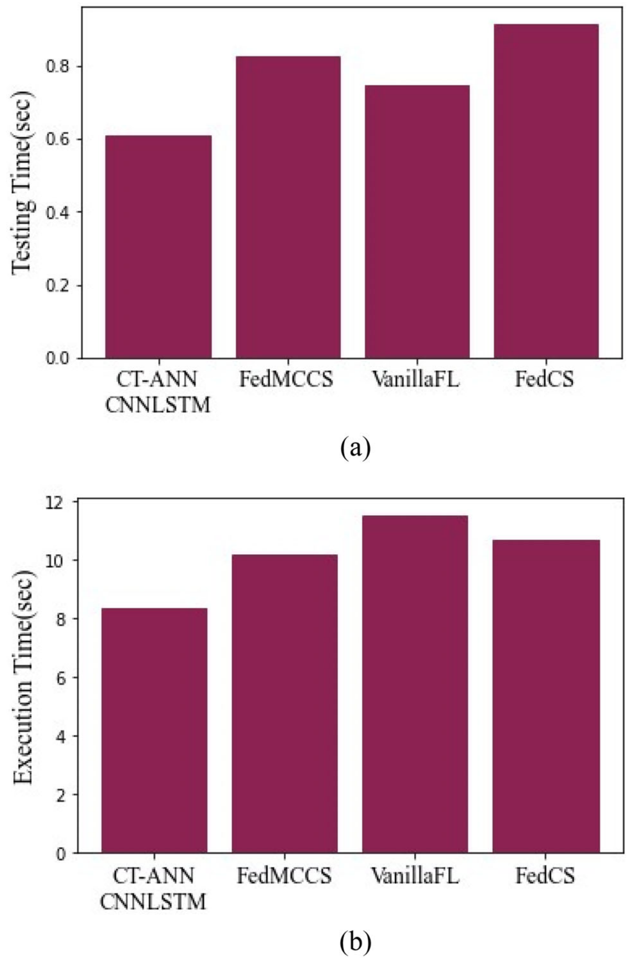


Fig. 17 **a** Testing time, **b** execution time evaluation

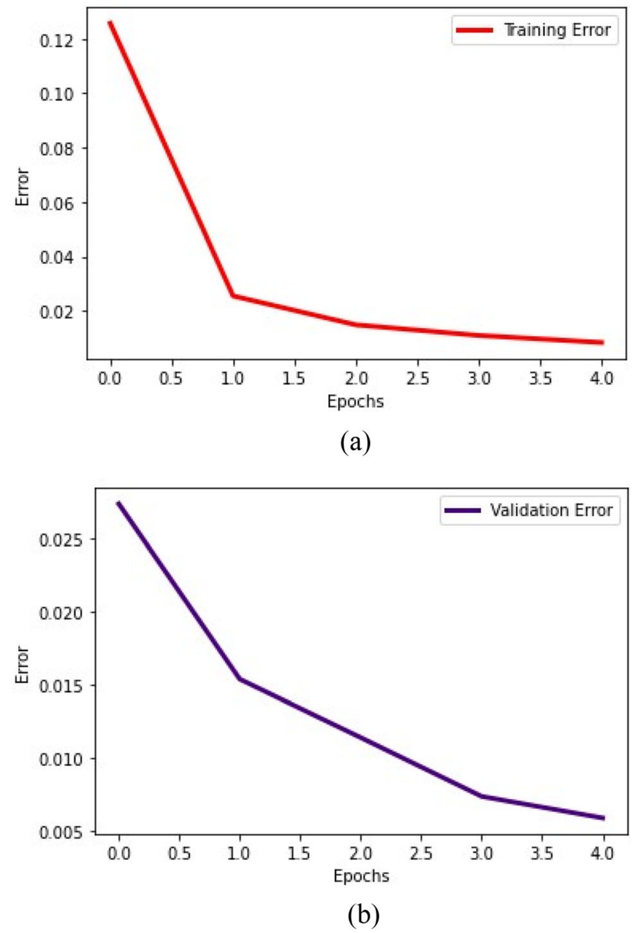
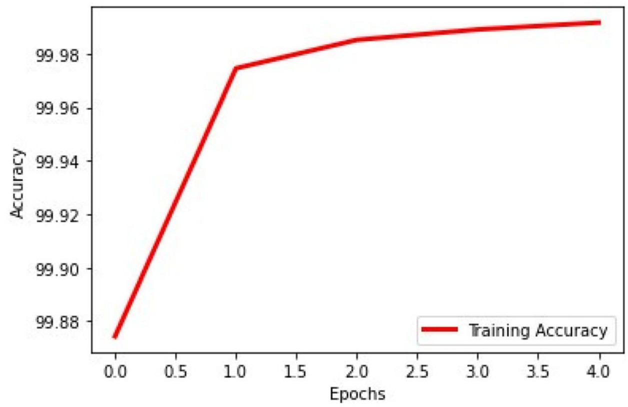


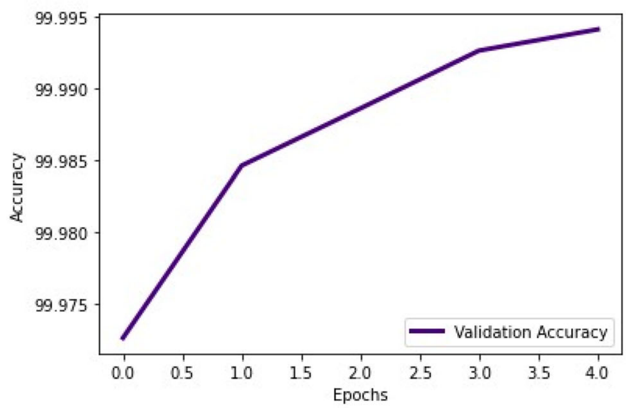
Fig. 18 **a** Training and **b** testing error analysis

service (QoS), network availability, and user experience while enhancing the connectivity of intelligent systems with improved network capacity. Due to the typical fragmentation and privacy of medical data, it might be difficult to get reliable results across populations. Complex machine-learning models can now be distributed trained because of recent advancements in federated learning (FL).

Consequently, FL has developed into a busy research area, with a focus on the decentralized processing of medical data at the network's edge to address privacy and security issues. This study designs an ideal federated learning system based on Class Topper Optimization (CTO). Various clinical data's uploaded by different clients were taken as input for this proposed work. To choose a homogeneous set of customers rather than a random assortment, client filtering was used. The stratified sampling was used to choose clients based on their metadata, preventing contacts with clients who weren't appropriate. The server then asks the filtered clients for resource information. Class topper optimization (CTO) based client selection algorithms were employed here. The global model variables are distributed to the chosen clients by the server. Each chosen client then uses its own local data to train the local model before sharing the newly created parameters to the server. In this proposed method, the homogeneous client was trained using a local model made of Artificial Neural Networks (ANN) and Convolutional Neural Networks (CNN). The global model is then enhanced periodically using the updates from the locally trained instances. As a general model, Long Short Term Memory (LSTM) was employed. The server terminates the round if not adequate updates have been submitted as well as a delay was recorded. To assess performance, the results of the suggested strategy are contrasted with those of the existing methods. The proposed CT based federated learning approach achieved 93% accuracy, 92% precision,



(a)



(b)

Fig. 19 Examination of **a** training and **b** testing accuracy

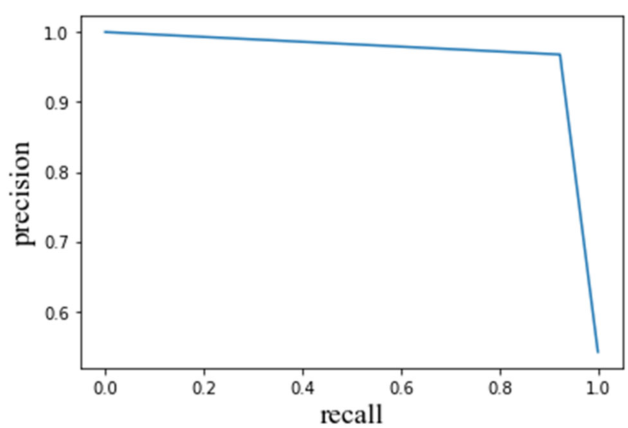


Fig. 20 Precision recall curve for proposed model

94% recall, specificity of 96%, execution time of 8.3 s, 7.7 s of training time and 0.6 s of testing time. The graphical representation of these performances shows that the proposed approach produces better results.

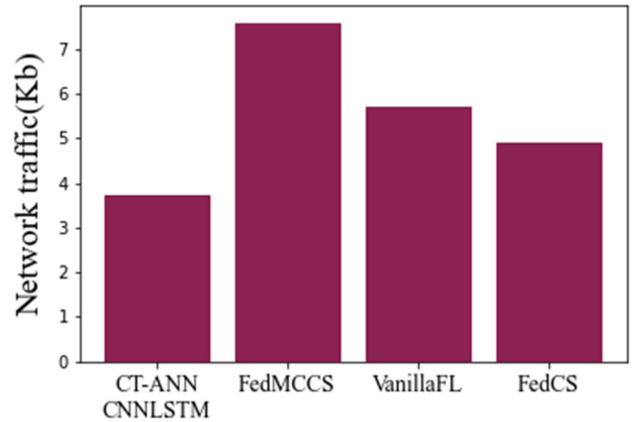


Fig. 21 Network traffic for proposed and existing model

Table 6 Comparison between proposed and existing state-of-the-art methods

Metrics	FedHealth [26]	FedMD [27]	FedNAS [28]	Proposed
Accuracy	90	76	91	93
Precision	89	78	88	92
Recall	89	74	91	94
F1 score	89	76	91	93

Author contributions The corresponding author claims the major contribution of the paper including formulation, analysis and editing. The co-author provides guidance to verify the analysis result and manuscript editing.

Funding The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Availability of data and material Not applicable.

Code availability Not applicable.

Declarations

Conflict of Interest The authors declared that they have no conflicts of interest to this work. We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

Compliance with ethical standards This article is a completely original work of its authors; it has not been published before and will not be sent to other publications until the journal's editorial board decides not to accept it for publication.

References

1. Grama, M., Musat, M., Muñoz-González, L., Passerat-Palmbach, J., Rueckert, D., Alansary, A.: Robust aggregation for adaptive privacy preserving federated learning in healthcare. arXiv:2009.08294. (2020)

2. Rahman, A., Hossain, M.S., Muhammad, G., Kundu, D., Debnath, T., Rahman, M., Khan, M.S.I., Tiwari, P., Band, S.S.: Federated learning-based AI approaches in smart healthcare: concepts, taxonomies, challenges and open issues. Clust. Comput. **26**(4), 2271–2311 (2023)
3. Kumar, Y., Singla, R.: Federated learning systems for healthcare: perspective and recent progress. Feder. Learn. Syst. Towards Next-Gener. AI. 141–156 (2021)
4. Arikumar, K.S., Prathiba, S.B., Alazab, M., Gadekallu, T.R., Pandya, S., Khan, J.M., Moorthy, R.S.: FL-PMI: federated learning-based person movement identification through wearable devices in smart healthcare systems. Sensors. **22**(4), 1377 (2022)
5. Terrail, J.O.D., Ayed, S.S., Cyffers, E., Grimberg, F., He, C., Loeb, R., Mangold, P., Marchand, T., Marfoq, O., Mushtaq, E., Muzellec, B.: FlambyDatasets and benchmarks for cross-silo federated learning in realistic healthcare settings. arXiv:2210.04620. (2022)
6. Karthick, S., Muthukumaran, N.: Deep regression network for single-image super-resolution based on down- and upsampling with RCA blocks. Natl. Acad. Sci. Lett. https://doi.org/10.1007/s40009-023-01353-5. (2023)
7. Adnan, M., Kalra, S., Cresswell, J.C., Taylor, G.W., Tizhoosh, H.R.: Federated learning and differential privacy for medical image analysis. Sci. Rep. **12**(1), 1953 (2022)
8. Passerat-Palmbach, J., Farnan, T., Miller, R., Gross, M.S., Flannery, H.L., Gleim, B.: A blockchain-orchestrated federated learning architecture for healthcare consortia. arXiv:1910.12603. (2019)
9. Lakhan, A., Mohammed, M.A., Nedoma, J., Martinek, R., Tiwari, P., Vidyarthi, A., Alkhayyat, A., Wang, W.: Federated-learning based privacy preservation and fraud-enabled blockchain IoMT system for healthcare. IEEE J. Biomed. Health Inform. **27**(2), 664–672 (2022)
10. Long, G., Shen, T., Tan, Y., Gerrard, L., Clarke, A., Jiang, J.: Federated learning for privacy-preserving open innovation future on digital health. In: Humanity Driven AI: Productivity, Well-being, Sustainability and Partnership, 113–133. Springer International Publishing, Cham (2021).
11. Hakak, S., Ray, S., Khan, W.Z., Scheme, E.: A framework for edge-assisted healthcare data analytics using federated learning. In: 2020 IEEE International Conference on Big Data (Big Data), pp. 3423–3427. IEEE (2020).
12. Guo, K., Chen, T., Ren, S., Li, N., Hu, M., Kang, J.: Federated learning empowered real-time medical data processing method for smart healthcare. IEEE/ACM Trans. Comput. Biol. Bioinform. 1–12 (2022). https://doi.org/10.1109/TCBB.2022.3185395
13. Qayyum, A., Ahmad, K., Ahsan, M.A., Al-Fuqaha, A., Qadir, J.: Collaborative federated learning for healthcare: multi-modal covid-19 diagnosis at the edge. IEEE Open J. Comp. Soc. **3**, 172–184 (2022)
14. Liu, Y., Yu, W., Ai, Z., Xu, G., Zhao, L., Tian, Z.: A blockchain-empowered federated learning in healthcare-based cyber physical systems. IEEE Trans. Netw. Sci. Eng. **10**(5), 2685–2696 (2022)
15. Silva, S., Altmann, A., Gutman, B., Lorenzi, M.: Fed-biomed: A general open-source frontend framework for federated learning in healthcare. In: Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning: Second MICCAI Workshop, DART 2020, and First MICCAI Workshop, DCL 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 4–8, 2020, Proceedings 2, pp. 201–210. Springer International Publishing, New York (2020)
16. Elayan, H., Aloqaily, M., Guizani, M.: Sustainability of healthcare data analysis IoT-based systems using deep federated learning. IEEE Internet Things J. **9**(10), 7338–7346 (2021)
17. Lu, W., Wang, J., Chen, Y., Qin, X., Xu, R., Dimitriadis, D., Qin, T.: Personalized federated learning with adaptive batchnorm for healthcare. IEEE Trans. Big Data. (2022)
18. AbdulRahman, S., Tout, H., Mourad, A., Talhi, C.: FedMCCS: multicriteria client selection model for optimal IoT federated learning. IEEE Internet Things J. **8**(6), 4723–4735 (2020)
19. McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics, pp. 1273–1282. PMLR (2017)
20. Nishio, T., & Yonetani, R. Client selection for federated learning with heterogeneous resources in mobile edge. In: ICC 2019–2019 IEEE international conference on communications (ICC), pp. 1–7. IEEE (2019)
21. Rahman, M.M., Tabash, M.I., Salamzadeh, A., Abduli, S., Rahaman, M.S.: Sampling techniques (probability) for quantitative social science researchers: a conceptual guidelines with examples. Seeu Rev. **17**(1), 42–51 (2022)
22. Srivastava, A., Das, D.K.: A new aggrandized class topper optimization algorithm to solve economic load dispatch problem in a power system. IEEE Trans. Cybern. **52**(6), 4187–4197 (2020)
23. Otchere, D.A., Ganat, T.O.A., Gholami, R., Ridha, S.: Application of supervised machine learning paradigms in the prediction of petroleum reservoir properties: comparative analysis of ANN and SVM models. J. Petrol. Sci. Eng. **200**, 108182 (2021)
24. Gu, Y., Wang, Y., Li, Z., Zhang, T., Li, Y., Wang, G., Cao, H.: A fault diagnosis method of four-mass vibration MEMS gyroscope based on ResNeXt-50 with attention mechanism and improved EWT algorithm. Micromachines. **14**(7), 1287 (2023)
25. Shahid, F., Zameer, A., Muneeb, M.: Predictions for COVID-19 with deep learning models of LSTM, GRU and Bi-LSTM. Chaos, Solitons & Fractals **140**. 110212 (2020)
26. Chen, Y., Qin, X., Wang, J., Yu, C., Gao, W.: Fedhealth: A federated transfer learning framework for wearable healthcare. IEEE Intell. Syst. **35**(4), 83–93 (2020)
27. Li, D., Wang, J.: Fedmd: Heterogenous federated learning via model distillation. arXiv:1910.03581. (2019)
28. He, C., Mushtaq, E., Ding, J., Avestimehr, S.: Fednas: Federated deep learning via neural architecture search. https://openreview.net/forum?id=1OHZX4YDqhT. (2021)

Mamta Narwaria is currently a Research Scholar in the Department of Computer Science & Engineering at Jaypee Institute of Information Technology (JIIT), Noida, India. She received the M Tech degree in Computer Science and Engineering from R.G.P.V University, Bhopal, in 2011. She received the B. Tech degree in Computer Science and Engineering from R.G.P.V University, Bhopal, in 2003. She is currently performing active research in the area of Federated Learning. Her research interests include Distributed Machine Learning, Deep learning, computer vision for medical imaging, and big data problems in healthcare.

Shruti Jaiswal is working as an Assistant Professor, in the Department of Computer science Engineering & Information Technology at Jaypee Institute of Information Technology(JIIT), Noida, (U.P).She has received her Ph.D degree from Delhi Technological University, Delhi. She has more than 10 years of teaching and research experience. She has Published various research Papers in International Journals and Conferences. She has delivered Special Lectures, Keynotes in many Conferences. Her research interest includes Software Engineering, Requirements Engineering, Security Engineering, Machine learning and federated learning.