Journal of Autonomous Intelligence (2024) Volume 7 Issue 1
doi: 10.32629/jai.v7i1.624

ORIGINAL RESEARCH ARTICLE

Role of federated learning in edge computing: A survey

Sambit Kumar Mishra*, Nehal Sampath Kumar, Bhaskar Rao, Brahmendra, Lakshmana Teja

Department of Computer Science and Engineering, SRM University-AP, Andhra Pradesh 522502, India

*** Corresponding author:** Sambit Kumar Mishra, skmishra.nitrkl@gmail.com

ABSTRACT

This paper explores various approaches to enhance federated learning (FL) through the utilization of edge computing. Three techniques, namely Edge-Fed, hybrid federated learning at edge devices, and cluster federated learning, are investigated. The Edge-Fed approach implements the computational and communication challenges faced by mobile devices in FL by offloading calculations to edge servers. It introduces a network architecture comprising a central cloud server, an edge server, and IoT devices, enabling local aggregations and reducing global communication frequency. Edge-Fed offers benefits such as reduced computational costs, faster training, and decreased bandwidth requirements. Hybrid federated learning at edge devices aims to optimize FL in multi-access edge computing (MAEC) systems. Cluster federated learning introduces a cluster-based hierarchical aggregation system to enhance FL performance. The paper explores the applications of these techniques in various domains, including smart cities, vehicular networks, healthcare, cybersecurity, natural language processing, autonomous vehicles and smart homes. The combination of edge computing (EC) and federated learning (FL) is a promising technique gaining popularity across many applications. EC brings cloud computing services closer to data sources, further enhancing FL. The integration of FL and EC offers potential benefits in terms of collaborative learning.

Keywords: cloud computing; edge computing; federated learning; hybrid federated learning; cluster federated learning; asynchronous federated learning; multi-tasking federated learning (MTFL); multi access edge computing (MAEC); vehicular edge networks (VEN); mobile edge computing (MEC)

ARTICLE INFO

Received: 7 May 2023
Accepted: 31 July 2023
Available online: 11 October 2023

1. Introduction

A modern form of machine learning that deals with the issues of data privacy and decentralization is federated learning. Traditional machine learning environments frequently collect and store data in centralized servers, causing issues with security and privacy. A solution is provided by federated learning, which enables machine learning models to be trained directly on distributed edge devices such as mobile phones, tablets, or Internet of Things (IoT) devices without the need for raw data to leave the devices. Federated learning greatly improves privacy protection by keeping data local and decentralized. The models are trained locally on the devices utilizing the relevant data instead of transferring the data to a central server for analysis. Only model updates or combined data are then shared with a central server. By ensuring that critical data stays on the devices, this method reduces the possibility of hacking or unauthorized access. Applications that deal with sensitive or personal data, like those in healthcare, banking, or personalized recommendations, have a lot of commitment to federated learning. While protecting data privacy and control, it enables organizations to take use of the collective understanding of distributed devices. Developments in many fields are made possible through federated learning, which uses the power of machine learning while protecting privacy[1]. Coming to edge computing, edge computing emerged as an evolutionary paradigm that deals with the limitations of traditional cloud-centric architectures in the context of rapidly increasing data volumes and real-time applications. Rather than depending entirely on centralized cloud servers, edge computing brings computational resources and data storage closer to the site of data generation, which is often at the edge of the network. Edge computing offers better processing, less latency, and better reaction times by placing edge devices, such as edge servers, gateways or IoT devices, at the network edge. Edge computing has a major beneficial effect on real-time applications like self-driving cars, virtual reality, augmented reality, and industrial IoT because it reduces the time and bandwidth needed to transport data from a location to a centralized cloud. By distributing computational tasks across edge devices and reducing the load on centralized cloud servers, edge computing also provides scalability benefits. Organizations without their own data centers can use cloud resources efficiently and cheaply due to this decentralized architecture[1,2].

We observed from many studies that, as a result of cloud computing the way we live, work, and learn has greatly changed but not in all cases, for example consider cisco internet business solutions group, once said that by the year 2020 the usage of 50 billion things that are connected to the internet will be formed but their assumption had become false and yet present the number of IoT devices connected to the internet are 14.4 billion because certain IoT applications need very quick responses, some of which may contain private data, and others of which may generate a lot of data, placing a tremendous burden on networks[3]. Therefore, cloud computing is not effective enough to support these applications for this kind of network. We envision that the edge of the network is transitioning from a data consumer to a data producer as a result of the pull from IoT and the push from cloud services. Here, the data that the cloud is consuming is very high such that the processing of the data is also becoming a huge task for the cloud. In this type of situation, the capacity of the cloud will not be suitable for heavy computations. Take Amazon as an example, in the year 2022, the company has an increasing number of job applicants, but it is difficult to accommodate them all. As a result, the employees have been let go. The same thing is happening in the cloud environment, where the number of devices is growing, and data processing is getting more and more challenging.

We researched the benefits of edge computing over cloud computing as well as several edge computing applications on Internet of Things (IoT) devices. We also discussed Edge-Fed and Fed-Avg, the advantages of Edge-Fed over Fed-Avg and Edge-Fed's future possibilities. We worked on cluster federated learning, dynamic federated learning[4], multitasking in federated learning[5] and hybrid federated learning at edge devices. Multi-tasking federated learning for predicting traffic. Asynchronous federated learning[6], collaborative federated learning at IoT devices, collaborative federated learning in healthcare for COVID-19 diagnosis, collaborative data sharing in vehicle systems, and wireless communications performance for collaborative federated learning[7]. We have collaborated on a table that covers things that are taken into consideration from various federated learning in edge computing sectors, such as energy consumption management in federated learning in edge systems and how caching and offloading take place in federated learning in mobile edge computing[8].

The rest of the paper is organized as follows. We presented the different applications of federated learning. The usage of federated learning over edge computing in different areas. In section 2, we presented Edge-Fed and its applications. In sections 3 and 4, we presented the hybrid FL, clustered FL, and dynamic FL. In section 5, the asynchronous FL is presented. In section 6, federated SGD is discussed. In section 7, multi-tasking FL is discussed. In section 8, collaborative FL over different applications in edge computing is presented. A few case studies are also discussed for all sections mentioned above. In section 9, energy consumption in FL is discussed. In section 10, caching and offloading in FL over mobile edge computing are discussed. In section 11, FL in wireless communication is discussed.

There are different types of techniques in federated learning. As edge computing is an emerging technology, our main theme of the research is to relate usage of federated learning in edge computing. So that the edge technology can be more efficient in different applications. Data privacy and security has been improved with federated learning, because federated learning enables training of the models locally without sharing raw data to the central server. Edge devices often operate in dynamic and unreliable network conditions. Federated learning can handle intermittent connectivity and device failures by allowing training to continue on available devices.

We used below mentioned techniques of federated learning because these techniques are widely used in different applications and these techniques will be helpful for enhanced data privacy, reduced communication overhead, lower latency, improved scalability, resilience, personalization and energy efficiency. These improvements make federated learning an attractive approach for deploying machine learning models in edge computing scenarios.

The techniques we included in this paper are Edge-Fed, cluster-dynamic cluster FL, hybrid FL, asynchronous FL, federated SGD and multi-tasking in FL, collaborative federated learning over different applications of edge computing.

2. Edge-Fed

Actually, mobile devices have to perform lots of calculations if they use the federated averaging (Fed-Avg) algorithm. Federated averaging (Fed-Avg) is an algorithm that is efficient in communication for distributed training with a larger number of clients. In Fed-Avg, a central server is used to communicate between the clients, and clients keep their data locally for privacy protection, but in Fed-Avg the distance between the central server and the clients will be more where that much bandwidth will take time[2].

To solve this problem, we have Edge-Fed where, Edge-Fed is a network with the central cloud server, edge server, and IoT devices, where the central cloud server all the global aggregations will take place, in the edge server, it takes some of the workloads from the IoT devices and finally, in IoT devices, local aggregations happen. In this section, the efficient usage of federated learning in edge computing is elaborated, which is also shown in **Figure 1**.

Between the edge server and the central server, bandwidth is less than the bandwidth between the clients and the edge server. Edge-Fed can decrease the needed global communication frequency to reach a satisfying accuracy. Consequently, the global communication cost can be reduced compared with Fed-Avg. Edge-Fed has advantages in different bandwidth scenarios from mobile clients to the edge server. By offloading part of the calculations, the computational cost of the mobile devices and the global communication expense can be simultaneously reduced as compared to Fed-Avg. For optimizing the federated learning based on edge computing, we can overcome the large computational cost in edge devices while performing Fed-Avg.

Some experiments were done (a division of the process of local updates to be completed by both mobile devices and the edge server and in between the edge servers and the central server, a global aggregation process was conducted). Those results show that the total computational and communication cost of edge devices are simultaneously reduced than Fed-Avg. If you want to do many calculations, then it may take more time. To reduce this type of time complexity the researchers found edge computing which collaborated with federated learning and formed edge federated learning. Each device will process autonomously with the help of Edge-Fed. The edge server now collects all outputs from mobile devices in order to increase learning effectiveness and decrease global communication frequency. The computation costs and computations from mobile clients to the edge server will decrease as a result of this Edge-Fed.

Some of the primary benefits of Edge-Fed are as follows: the local updates will happen in both IoT devices and edge servers. Where IoT devices can focus more on low layers and edges, services can do more computational tasks with the required resources. IoT devices and edge servers make up the two categories of local updates for this model. While services can do more computational activities with the necessary resources, IoT devices may concentrate more on low layers and edges. So that the training needed for mobile devices will be low and faster. The bandwidth of clients on the edge server is very high so, by using Edge-Fed we can reduce bandwidth. By Edge-Fed we can also apply some guidelines in which the model can run accurately with less resource cost.

IoT helps the present world in many ways. IoT stands for the Internet of Things, which is the networking of physical things equipped with sensors, software and other technologies for establishing connections and exchanging data with other devices through the internet. These days, IoT devices are employed more in public safety equipment and by farmers to monitor their crops and do surveys, among other uses. Low latency and power consumption are required for IoT devices to carry out duties like monitoring and uploading sensor data, among others[3].

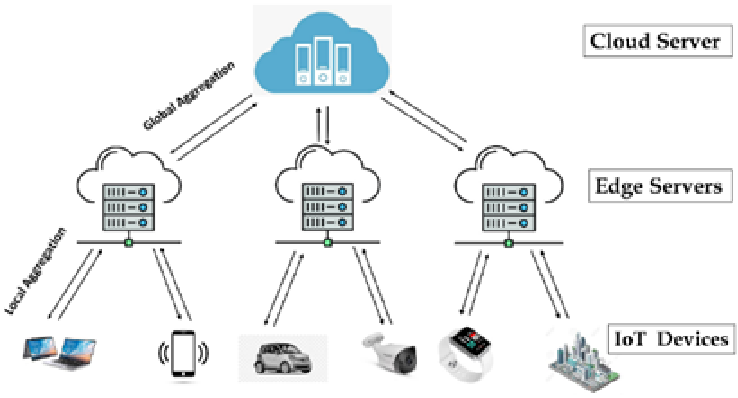

Figure 1. Efficient usage of federated learning in edge computing.

Generally, IoT devices are weak due to security issues. For IoT, device authentication is required to ensure that the connected devices on the IoT are trusted. For that unique identifier is needed. In order to optimize mobile edge computing and communication, Deep reinforcement learning (DRL) and federated learning (FL) frameworks were combined with mobile edge systems. Energy consumption and execution delays are some advantages of this partnership. Only the local information is required for this, the channel state information can be known afterward.

We came across some of the assumptions as follows: IoT devices are typically thought of as being extremely powerful gadgets with the capacity to train their own DRL agents on their own. IoT devices may not be as powerful in the near future, they may only be able to compute using minimal neural networks. Researchers were inspired to develop a new type of computing known as edge computing, which aids in processing data at the network's edges, as a result of the Internet of Things (IoT) explosive growth and the enormous popularity of its rich cloud services. We will examine a variety of case studies in this, ranging from the cloud to smart cities and smart homes, as well as collaborative edge to broaden the idea of edge computing.

Some of the case studies explain cloud offloading, smart city, and video analytics where they have used edge computing and federated learning for better results.

1) Akamai: Akamai is a leader in content delivery networks and edge computing. Akamai has a network of servers deployed at the edge of the internet, providing advanced services like streaming video and web acceleration. Akamai's edge computing platform provides the scalability, security, and performance necessary to serve the world's most heavily trafficked websites.

2) Fastly: Fastly is an edge computing platform that enables organizations to build, deploy, and scale applications at the edge of the internet. Fastly's edge network provides customers with faster page load times, lower latency, and improved reliability. Fastly's edge computing platform is powered by its proprietary varnish cache technology, which allows customers to quickly deliver content and applications without relying on the traditional data center model.

2.1. Applications of Edge-Fed

There are several applications of Edge-Fed in daily life where we see the usage of Edge-Fed. Now we will discuss some of the scenarios on Edge-Fed.

2.2. Smart city

In a smart city we have a number of IoT devices that all are connected to an edge server where all the computations will occur and that is sent to the cloud server for global aggregation. For example, take a house in a smart city where all the devices are connected using edge computing where all the computations of that house are stored in an edge in the same way all the houses in that city are connected to the edge computing and all their computations and security should be maintained by the edge computing. So, if we use cloud computing for this type of computing it will not be efficient.

2.3. Vehicular network

Federated learning in edge computing can be utilized in autonomous vehicles to improve their perception and decision-making capabilities. Edge devices within vehicles can collaboratively train models using data collected from multiple vehicles while preserving data privacy. This allows for real-time learning, personalized driving experiences, enhanced safety through shared knowledge and insights.

3. Hybrid federated learning at edge devices

In a typical FL process, clients (end devices) finish model training on local data during each training cycle, and the cloud combines local models into a global model using a weight-averaging technique called Fed-Avg[2]. Hybrid FL's energy-saving function might actually draw in more end devices throughout each cycle. The amount of energy used by a gadget can have a significant impact on its owner's willingness to take part in FL training.

In this study, comprehensive tests were done, and the findings show that hybrid FL greatly enhances FL in the MAEC system by lowering device-side energy consumption, reducing the average round length, accelerating the convergence of the global model, and enhancing model correctness. The suggested protocol's (hybrid FL) performance in enhancing FL efficiency, enhancing the quality of the global model and reducing on-device energy consumption in a three-layer MAEC system has been evaluated using the two machine learning tasks in various environmental circumstances. They performed the FL process in two approaches: Stop the procedure after a certain number of rounds. Tmax and stop when a preset accuracy is achieved for the global model. Some of the case studies that include hybrid federated learning at edge devices for better results are as follows:

- A large retail chain wanted to develop a personalized customer experience by predicting customer preferences in a privacy-preserving manner. The company had millions of customers and wanted to use this data to develop a personalized customer experience for their customers. The company decided to use hybrid federated learning to address its needs. They used a private cloud infrastructure to deploy a federated learning system. The federated model was trained using customer data from two of the company's stores. The first store was used as the main training set and the second store was used as a test set. The model was trained using a federated learning algorithm and the data was anonymized before being sent to the cloud. This enabled the company to keep customer data secure and private. The company was able to successfully train its model and obtain better results than using traditional centralized training. This helped them increase customer satisfaction and profits.

- A large financial institution was looking for a way to securely share customer data with its partners without compromising privacy. The company wanted to use machine learning to gain insights from the data but wanted to ensure that the data remained private and secure. The company decided to use hybrid federated learning to address its needs. They used a private cloud infrastructure to deploy a federated learning system. The federated model was trained using customer data from the financial institution and its partners. The model was trained using a federated learning algorithm and the data was anonymized before being sent to the cloud. This enabled the company to keep customer data secure and private while still obtaining insights from the data. The company was able to successfully train their model and was able to gain insights from the data without compromising privacy. This helped them increase profits and customer satisfaction.

3.1. Applications of hybrid federated learning

There are several applications of hybrid federated learning in daily life where we see the usage of hybrid federated learning. Now we will discuss the scenarios of hybrid federated learning.

3.1.1. Health care

Scenario:

Hospitals and medical institutions often have sensitive patient data that cannot be shared due to privacy regulations. Hybrid federated learning can be employed to train predictive models on patient data from multiple institutions without the need to transfer the data to a centralized location.

Details:

The central server provides an initial model, and local clients (hospitals) train the model on their respective datasets. The updated models are then sent back to the central server, which aggregates the model updates and shares a new global model. This iterative process ensures that the model improves while preserving patient privacy.

3.1.2. Cybersecurity

Scenario:

Organizations often collect vast amounts of security-related data, including network traffic, logs and intrusion detection alerts. Sharing this data for centralized training poses privacy and security risks. Hybrid federated learning can be employed to collaboratively train intrusion detection or malware detection models across multiple organizations while preserving the confidentiality of their data.

Details:

Each organization acts as a local client and trains a model using its own security data. The central server coordinates the training process and aggregates model updates from the organizations. By combining knowledge from different sources, the global model becomes more robust against emerging cyber threats. However, the raw data remains on the local clients, ensuring data privacy and minimizing the risk of exposing sensitive information.

3.1.3. Natural language processing (NLP)

Scenario:

NLP models often require large amounts of text data for training, which may come from various sources, such as social media, news articles and private documents. In situations where the data sources cannot be centrally accessed due to privacy concerns, hybrid federated learning can be used to collaboratively train NLP models across multiple data owners while maintaining data privacy.

Details:

Data owners, such as different organizations or individuals, act as local clients and train NLP models using their respective text datasets. By combining knowledge from diverse data sources, the global NLP model becomes more versatile and accurate. Importantly, the raw text data remains localized, ensuring data privacy and confidentiality.

4. Cluster federated learning

The researchers have created a cluster-based, hierarchically aggregated federated learning system in this work. CFL also can be done dynamically too. They have developed a successful strategy that divides the edge nodes into *K* clusters using balanced clustering. The edge nodes in one cluster send their local updates to the cluster header for synchronous aggregation or cluster aggregation in order to determine the optimal number of clusters with resource constraints and perform edge computing training. To address the actual network dynamics, they further developed their method. Their experimental findings showed that, as compared to baselines, the suggested mechanism may achieve exceptional performance when faced with resource limitations[4]. Please refer to "**Table 1**", for the results of experimental findings between cluster FL and dynamic cluster FL.

Table 1. Observations and results of cluster FL and dynamic cluster FL.

Algorithms performed	**Accuracy (while training CNN model over CIFER-10 dataset)**	**Accuracy (while training CNN model over MNIST dataset)**	**Observations**
Cluster-based FL (CFL)	35%	74.2%	Effectively deals with data imbalance.
Dynamic CFL (DCFL)	51%	95.9%	Handles the node failure and maintains the training process well.

Some of the case studies that include cluster federated learning for better results are as follows:

1. Cluster federated learning on electronic health records: a case study of anemia diagnosis.

2. Cluster federated learning on mobile applications: a case study of predicting user engagement.

4.1. Applications of cluster federated learning

There are several applications of cluster federated learning in daily life where we see the usage of cluster federated learning. Now we will discuss the scenarios on cluster federated learning.

4.1.1. Autonomous vehicle training

Autonomous vehicles require large amounts of data to train and traditional methods of data sharing are not efficient or secure. Cluster federated learning can be used to train autonomous vehicles, with data from multiple sources, in a secure and decentralized manner.

4.1.2. Healthcare

Cluster federated learning can be employed in healthcare systems that consist of multiple hospitals or healthcare institutions. Devices within each cluster, such as hospital servers or patient monitoring devices, can collaborate to train models specific to their cluster while adhering to data privacy regulations. This enables knowledge sharing, disease detection, treatment recommendation systems, and improved healthcare outcomes across different clusters.

4.1.3. Recommendation systems

Recommendation systems require large amounts of data to be trained on. Cluster federated learning can be used to train recommendation systems on multiple datasets without having to share the underlying data.

4.2. Applications of dynamic cluster federated learning

There are several applications of dynamic cluster federated learning in daily life where we see the usage of dynamic cluster federated learning. Now we will discuss the scenarios of dynamic cluster federated learning.

4.2.1. Autonomous driving

Dynamic cluster federated learning can be used for intelligent traffic management in urban areas. Clusters can be formed based on traffic patterns, such as congestion-prone areas or specific road segments. Devices within each cluster can collaborate to train models for real-time traffic prediction, adaptive traffic signal control, and congestion mitigation. This approach enhances traffic flow efficiency, reduces travel time, and optimizes transportation systems.

4.2.2. IoT devices

Dynamic cluster federated learning can be employed in adaptive IoT systems where the clustering is based on the changing context or network conditions. Clusters can be dynamically formed to optimize data processing, resource allocation and service provisioning in IoT networks. This approach enables intelligent and self-adaptive IoT systems that can adjust their operations based on the dynamically changing environment.

4.2.3. Healthcare

Dynamic cluster federated learning can enhance personalized healthcare by considering the dynamic formation of clusters based on patient characteristics, disease types or treatment requirements. Clusters can be formed to allow collaboration between medical devices, hospitals, and research institutions. This facilitates privacy-preserving model training for personalized diagnosis, treatment recommendation, drug discovery and improving healthcare outcomes while maintaining data confidentiality.

5. Asynchronous federated learning

With the massive growth of IoT, a large amount of data is generated from the real world every day. The data will move to the central place or center through the network for the instruction, which will use huge amounts of bandwidth usage. Due to this, it can keep more load on the local server, and it has sent a lot of work on the edge, which we can say is edge computing. This FL application has been used at those edges. We use FL because it keeps the security of the clients whose data is stored on those servers.

FL can be done on one or many servers (parameter server) which can be known as a group of servers. In which we need many numbers of edge nodes or edges. In these many servers, each one is controlled by the head and they have kept a division of the widely shared parameters. Let us take the example as one parameter server in which each edge worker is taken care of computing the work in the local server after completing the work in each server, they send the data of the completed one into the parameter server.

To perform this there are many factors such as resource constraints, data imbalance, edge uncertainty.

The two methods for federated learning in the edge computing:

1) Synchronous scheme: using this approach, data is gathered from all local servers and transferred to the training data's global servers. The parameter server will then divide the work into smaller tasks and deliver information about each edge node's updated nodes for the specified amount of time. However, there will be drawbacks such as issues with bandwidth and training time.
2) Asynchronous scheme: with this approach, data is gathered from a few local servers and transferred to the training data's worldwide servers. The parameter server will then divide the work into smaller tasks and deliver information about each edge node's updated nodes for the specified amount of time. We also require a lot of time and bandwidth for this.

Communication-efficient asynchronous federated learning (CE-AFL) approach is employed to get around this. This approach collects data from a specific number of edge nodes' local servers and sends it to the training data's global servers. The work of a specific number of edge nodes will be combined by the parameter server, which will then update the local server in the order of arrival time. In edge computing with limited resources, federated learning enables the training of global models over over-dispersed datasets. The ideal outcome of training data from various users and devices is enhanced model representation and generalization[6].

For optimizing this federated learning, we use two solutions like gradient descent, stochastic gradient descent.

The case study of asynchronous federated learning for mobile applications is described below:

- A corporation that has to access client data from various sources, such as mobile applications, online browsers and other connected devices, may be the subject of a federated learning case study. The corporation wants to use the information gathered from various sources to create a model that can precisely predict client behavior. The business would need to create an asynchronous federated learning system in order to achieve this goal. Multiple models of various sources of client data would be used in this system. The findings would be synced among the models after each model had been trained on a portion of the data. By doing so, the model would be able to use data from different sources while learning patterns that might be particular to each data source. The models can be used to forecast customer behavior once they have been trained. After that, the business can utilize this data to decide how to improve customer service. This might entail adjustments to goods or services, marketing plans or customer service techniques. The organization can acquire insights from several sources of consumer data by using asynchronous federated learning without having to move the data between sources. This keeps customers' personal information private while still giving the business useful information.

5.1. Applications of asynchronous federated learning

5.1.1. Federated learning in unstable environments

Asynchronous federated learning is more resilient to unstable network conditions, device failures and dropouts. Since updates can be sent independently, the learning process can continue even when devices experience intermittent connectivity or are temporarily offline. This makes it suitable for applications in remote areas, disaster-prone regions, or mobile networks with unreliable connections.

5.1.2. Large-scale collaborative systems

In scenarios where a massive number of devices or nodes participate in the federated learning process, synchronous communication can become a significant challenge. Asynchronous federated learning enables a more scalable approach by allowing devices to contribute updates at their own pace. This makes it suitable for applications like social networks, federated recommendation systems and crowd-sourced data analysis, where numerous users contribute to the learning process.

5.1.3. Real-time adaptation and personalization

Asynchronous federated learning enables real-time adaptation and personalization of models. Devices can update their local models based on their individual data and preferences, without waiting for synchronization with other devices. This enables applications like personalized news recommendations, adaptive chatbots and real-time user behavior modeling, where timely updates and personalized experiences are critical.

6. Federated SGD

These days, the corporate community is paying close attention to machine learning, which the company will handle the data. The data is gathered from many sources, and ML applies operations to it. However, the problem is that the collected data should be moved to a single location where the ML can execute its duty as stated above, but due to security concerns, which prevented anybody from authorizing the move, the company's reputation and their clients' data were damaged. The solution is PPML (privacy preserving machine learning), but because ML algorithms in telecommunications are data-hungry, we can't take the data into a single spot due to a number of problems.

Due to the aforementioned factors, the FL framework was applied in this communication. Tensor flow and pytorch were used to create this framework.

The optical networks are becoming their own distinct entities and ML is probably going to play a different role in this. However, there are a lot of difficulties to be aware of throughout this era of change. The inaccessibility of the original data that was collected makes it difficult to create ML-based solutions. While the FL framework fixes the ML issues in optical networks.

Federated SGD (stochastic gradient descent) is a distributed machine learning technique that allows multiple devices to collaborate in training a machine learning model. This technique enables devices that are not connected to a central server to train a model without needing to share their local data. Federated SGD can be used to train models on data that is distributed across multiple devices, such as smartphones, IoT devices, and edge devices. This allows for distributed training in a secure manner and can be used to train models that would otherwise require a central server.

Some of the case studies that include for better results are as follows:

- The first case study concerns a study carried out in the UK that employed federated SGD to lower the price of optical network planning. In order to learn a model of the network topology and evaluate it to improve the arrangement of the optical links, the project employed federated SGD. This decreased the cost of network setup and maintenance and allowed the team to plan the network quickly and precisely.
- The second case study focuses on a study that was carried out in the US and employed federated SGD to lower optical network's power requirements. In order to reduce power consumption, the project optimized the location of optical links and components using federated SGD, which was used to learn a model of the optical network. As a result, the team was able to considerably lower the network's overall power consumption. These case examples demonstrate that federated SGD can be used to plan effectively and precisely.

6.1. Applications of federated SGD

Federated SGD has a wide range of applications in machine learning, particularly in the areas of healthcare, finance, and communication.

6.1.1. Healthcare

Fed-SGD can be employed in federated healthcare systems, where multiple hospitals or healthcare institutions collaborate to train models on patient data while preserving privacy. Each institution performs local updates using stochastic gradient descent on its own patient data, contributing to the collective learning process. Fed-SGD can be used for disease prediction, treatment recommendation, clinical decision support, and medical research while adhering to privacy regulations.

6.1.2. Finance

Fed-SGD can be applied to federated financial systems to train models on distributed financial data while respecting privacy and security. Each financial institution performs local updates using stochastic gradient descent on its transactional data, allowing for collaborative learning without sharing sensitive customer information. Fed-SGD can be used for fraud detection, risk assessment, personalized financial services and regulatory compliance in a privacy-preserving manner.

6.1.3. Distributed machine learning platforms

Fed-SGD can be integrated into distributed machine learning platforms that span multiple devices or cloud servers. Each computing node performs local stochastic gradient descent updates, and the model updates are aggregated to create a global model. Fed-SGD enables distributed training, efficient utilization of computing resources and scalability for large-scale machine learning applications.

7. Multi-tasking federated learning (MTFL)

It is a unique function of federated learning that solves concerns like user-content recommendation and enhances the accuracy of each individual user model (UA). Non-federated batch-normalization (BN) layers are added to the data by MTFL. By enabling users to train models specifically tailored to their data, MTFL improves user model accuracy and convergence speed.

Some of the case studies that included multitasking are as follows:

- One example of a study using MTFL comes from a research project conducted at the university of Oxford in which the researchers used MTFL to train a model for predicting hospital readmission risk for patients with chronic obstructive pulmonary disease (COPD). The researchers used data from multiple hospitals in the United Kingdom, which were collected in a federated manner. The model was trained using MTFL, which allowed the researchers to take advantage of data from multiple sources while preserving the privacy of the patient's data. The researchers found that MTFL was able to improve the accuracy of the model compared to traditional machine learning methods. Moreover, the model was able to generalize well to new datasets and was able to be deployed in a real-world setting. The results of the study demonstrate the potential of MTFL for healthcare applications.
- Multi-task federated learning has been used in computer vision and speech recognition tasks. Researchers have developed federated learning models that jointly learn to recognize objects in images and transcribe speech. By combining these tasks, the models benefit from shared representations and improve performance on both tasks.
- In the context of autonomous driving, multi-task federated learning was employed to simultaneously learn perception tasks, including object detection and lane detection. The study involved a fleet of vehicles that collaboratively trained a shared model while respecting data privacy. Each vehicle contributed its local sensor data for training, and the federated learning framework facilitated the exchange of model updates between the vehicles. By jointly learning multiple perception tasks, the vehicles improved their ability to detect objects and lanes, enhancing overall safety and performance.
- Multi-task federated learning was applied to IoT applications, specifically in a smart home environment. The goal was to collectively train a shared model on devices within the home to perform tasks like activity recognition, energy optimization, and anomaly detection. By training the model collaboratively while

keeping data local, the privacy of users' activities and preferences was preserved. The shared model learned from the distributed data, enabling it to make accurate predictions about activities, optimize energy consumption, and identify anomalous events within the smart home ecosystem.

7.1. Applications of multi-task federated learning

There are several applications of multi-tasking federated learning in daily life where we see the usage of multi-tasking federated learning. Now we will discuss the scenarios of multi-tasking federated learning.

1) Without actually exchanging the data obtained among traffic stations, we can still optimize the traffic forecast models. The route planning problem, which utilizes a modified A* algorithm, is addressed by the multi-task FL model that is proposed. The suggested multi-task FL framework's improved prediction accuracy has been confirmed by simulation results.

2) We can choose an efficient route with shorter travel time than Google maps and distance-based schemes by using multi-horizon traffic speed prediction.

3) Different optimization strategies (including Fed-Avg-Adam can be used within MTFL).

Now, we discuss the MTFL algorithm working in edge computing. This algorithm is mainly based on the client-server framework which is shown in **Figure 2**.

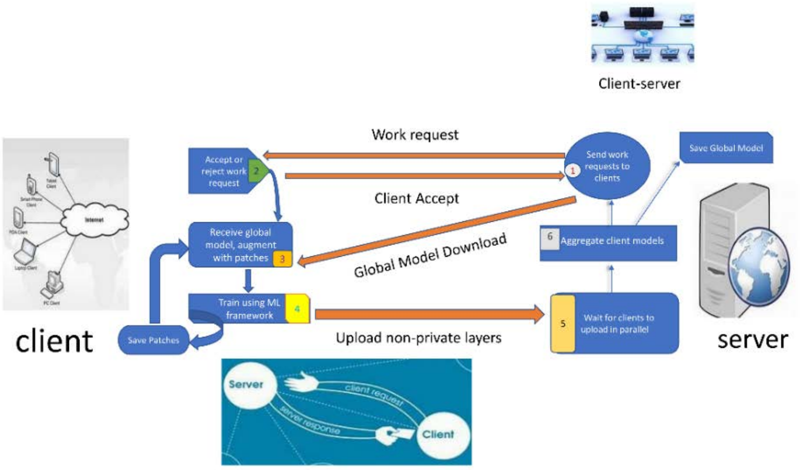

Figure 2. Client-server framework.

Rounds of training are conducted here until a termination condition is satisfied. The server, which is seen in the above image, starts these rounds.

The processes involved in using the MTFL algorithm in edge computing are listed below.

The server selects a group of clients to participate in the round from its whole database and sends them a task request. The clients now get the request from the server and send back an accepted message in response based on the user preferences. This suggests that a user may set up their gadget such that it only participates in the FL when it is fully charged and linked to a Wi-Fi network. Clients who accept the request then signal the server with an accept message. The server now sends the global model to its receptive clients. The clients modify the local copy of the global model with their own fixes. We use patches from batch-normalization layers in this. The clients now carry out local training before conserving their patches for the following round. A subset of clients that agreed to participate in the rounds submit their non-private model and optimizer settings to the server after waiting a predetermined amount of time. The server then saves the aggregate to create a single global model that is saved on the server after sorting all models by average. After that, the server may start a fresh round.

7.1.1. Traffic prediction by multi-task federated learning

In this, we did the traffic system prediction of the traffic jam and the speed detection of the vehicle for the upgrade of the intelligence transportation systems. Because it helps the citizens when choosing their raid and both to improve fuel economy and to reduce air pollution. These will benefit the populace and are essential to cutting-edge traffic management systems[5]. An inductive loop may monitor travel speed by measuring how inductance varies over time in the corresponding traffic lights and Internet of Things devices and this information can be utilized to estimate traffic speed. They'll utilize a lot of IoT devices to gather an unrivaled amount of traffic data from the actual world. This enormous amount of data may lead to an increase in studies in this area. They can be taken into an entity into two categories: the ARIMA model, and the Kalman filtering models.

When dealing with traffic simply in regular changes, for instance during rush hours or peak hours in a city, we may provide reports that are worthwhile. The value will differ from the real values during peak hours due to the unusual nature of the traffic and the parametric approach's projections. We may fit the data and do ML-like non-parametric techniques by gathering the data and performing the above-mentioned mathematical procedures. Using data gathered by the signal and its arriving and departing traffic signals, a long short-term memory (LSTM), recurrent neural network (RNN) can anticipate the speed and filling of a traffic jam. To capture the latent traffic evolution patterns inside the underlying traffic network, we shall employ the convolution neural network (CNN). This project seeks to use cutting-edge ML, where we may utilize a single task learning approach, sometimes referred to as the (STL). However, in actuality, there will be a variety of elements at play, including changes in the weather as well as several events like camps and protests. A significant responsibility or load for STL is laying the new roads and modifying the existing ones. We can presume that the local data or traffic signal may have a memory limit where there will be issues with bandwidth and storage. To address this, they provide multi-task federated learning (FL), which will improve the traffic conditions as previously described. We first employ the acquired traffic data's field values, which are impacted by their immediate neighbors, and build a divisive hierarchical clustering (DIANA), which separates the traffic data at each signal into a collection of clusters. The federated learning will then do or train them at each local data set without doing so on the global or central cluster for each data cluster scattered across signals. In this situation, the proposed multi-task FL architecture can safeguard privacy and lower transmission costs because no local data is exchanged among stations. On the map where the plan was carried out, we applied the multi-horizon speed prediction from the multi-task federated learning framework. The dependency graph will be used to create them in this road map and the modified A* method will be used to identify the cheapest and fastest route possible. To provide the most accurate traffic predictions in a variety of anomalous traffic scenarios, we additionally take into account the traffic and route map.

8. Collaborative federated learning

Collaborative federated learning allows for the implementation of FL on edge devices with a reduced need for a central controller. Supporting privacy framework having the capacity to use more training data samples than original FL. Due to collaborative FL's ability to support more devices in the FL process compared to original FL, it is suitable for application in large-scale systems. For secure data sharing this collaborative federated learning is efficient.

Some of the case studies in collaborative federated learning are as follows:

- Google's federated learning of cohorts (FLoC): Google's federated learning of cohorts (FLoC) is a collaborative federated learning method that uses a machine learning model to group users into cohorts based on their browsing history. This helps Google to deliver more relevant ads to users while protecting their privacy. FLoC is a privacy-preserving technique that allows Google to use the data from a user's browsing history to identify similarities between them and other users. The data is then used to group users into cohorts, or groups of users, who share similar interests and behaviors. This allows Google to target ads more accurately and efficiently.

8.1. Applications of collaborative federated learning

There are several applications of collaborative federated learning in daily life where we see the usage of collaborative federated learning. Now we will discuss the scenarios of collaborative federated learning.

8.1.1. Collaborative federated learning for diagnosis of COVID-19 using X-ray and ultrasound

Collaborative federated learning in the health care system has the main advantage of data privacy of the patient. Sending data directly between end devices and cloud servers may lead to security problems. By using ML at the edge devices, the security of the system increases. The two datasets X-ray and ultrasound are taken to detect COVID-19. Clustered federated learning is in which the parameters are trained jointly[7].

Each cluster has a number of clients. Each client trains the model which is shared by the server. After completion of the training of models the clients share the learned weights with the server, then the server calculates the Fed-Avg. The model is updated with new weights after which multi-modal data testing will be performed and check whether the criteria are matched if not the process repeats.

The datasets in this instance are split into two categories: a training set and a testing set. Each data set has 20% for testing and 80% for training. The two baselines specialized federated learning and conventional federated learning are used to compare the performance of clustered federated learning. The model is trained in a federated learning environment while using customized FL. An ML model is introduced in the federated learning environment in conventional FL.

The results of the approach are that the performance of clustered federated learning is better than the performance of conventional federated learning. In ultrasound, if the inflection point in the value of the loss function is reached the federated learning rounds will be stopped. If the multimodal machine learning model beyond which cannot be enhanced, then the point is called the inflection point.

8.1.2. Collaborative data sharing in vehicular edge devices using federated learning

The Internet of vehicles (IoV) is a newly emerging technology in which all vehicles are connected and share vehicular data through wireless connections to control traffic and to make traffic well organized.

Multi-access edge computing (MAEC) can retain sensitive vehicular data close to the vehicles by utilizing federated learning in the vehicular edge networks (VEN) and only transferring the locally learned feature parameters to the cloud server for global aggregate[9].

The architecture has four layers:

Cloud computing layer (CCL): traditional cloud servers and VEN-specific cloud servers are the two categories of cloud servers. Wired fiber lines are used to connect the two cloud servers.

Load balance and cache layer (LBCL): load balancers identify the traffic source by gathering protocol headers and distributing traffic according to application scenarios to increase the effectiveness of resource usage for network traffic and data. Caches are set up to work with load balancers to resolve duplicate services and pre-allocate storage resources.

Edge computing layer (ECL): VEN's vehicular edge devices are supported primarily by the ECL in terms of a variety of services and applications.

Vehicular edge layer (VEL): in the VEL, vehicle edge devices are directly connected to neighboring MAECS (multi access edge computing servers) over wireless cellular networks, and they produce data traffic from different user defined VEN services and applications.

Advantages of using federated learning for MAEC-empowered IoV:

Reduction in network bandwidth. Massive VS data is not transmitted to the data center for training since federated learning can decentralize to learn a deep model at different MAECSs. The use of network bandwidth, energy consumption, and data transfer could all be significantly decreased via federated learning. Privacy protection. Users' sensitive information is shielded from the risk of hacking because less data must be sent to the data center and their privacy is somewhat secured by low latency. Each local participant (edge server or device) in federated learning takes decisions in real time, allowing the deep models to be consistently trained and updated while lowering transmission latency.

The results for the approach are as follows, the convergence of the proposed algorithm is faster than the centralized scheme. The latency of the proposed algorithm and centralized scheme are similar which means both methods give optimal results.

8.1.3. Wireless communications for collaborative federated learning

In order to use FL over IoT networks in practice, edge devices must repeatedly send their trained ML models to a central controller over wireless links. Only some devices can use FL due to constrained wireless resources, such as those found in an IoT.

The learning process is hampered by wireless channel faults and delays that are transmitted from IoT devices to a central controller (such as a base station). In order to enhance FL performance, it is required to take into account wireless network optimization[10].

Performance of collaborative federated learning over wireless networks:

Loss function: CFL training's objective is to locate an ML model that depends on the local FL models of all involved devices, which minimizes the loss function. These models encounter transmission delays and mistakes when sent over wireless networks, which might have an adverse effect on the loss function during training. Due to limited energy only a few data samples were involved, and the loss function increased.

Convergence time: the time it takes for each device to train its local FL model plus the number of iterations needed for FL convergence makes up the CFL convergence time.

For each CFL iteration, there are a predetermined amount of local FL model changes.

Energy consumption: the amount of energy used by CFL depends on a number of factors, including the size of the FL model data, the distance between the BS and the devices, the needed convergence time, and the goal loss function value.

Reliability: the wireless channel conditions affect CFL's reliability. Each device's transmit power improves as a result, which reduces the number of inaccurate local FL models and raises CFL reliability.

9. Energy consumption management in federated learning in edge environment

This study proposes energy-aware resource management for the MAEC-enabled FL. By enabling mobile users to offload a portion of their local dataset to the MAEC server, the tradeoff between the performance of the training model and the energy consumption at user devices with regard to the number of data samples used for local training is particularly handled. To accomplish this, a resource management problem that considers energy limitations is created with the intention of decreasing training loss and time usage. The formulated problem is split into multiple more manageable problems due to the relationship between the decision components. The strategy for controlling computing resources is then created by ensuring the energy budget of mobile users. The phrasing of the dataset offloading and uplink resource management problem as a GNEP (generalized nash equilibrium problem) also leads to the conclusion that a GNE exists. The dataset offloading and uplink bandwidth allocation issues are resolved to decrease overall time consumption. To accomplish this, the energy-aware resource management algorithm is recommended. The proposed MAEC-enabled FL model's overall time consumption is competitively lower than that of the traditional FL technique, according to extensive simulations that employ the recommended resource management algorithm. The performance parameter energy consumption is considered by many researchers.

Due to the limited energy and computational capabilities of mobile devices, the efficiency of model training to meet the aim of local energy reduction is always in jeopardy. Multi-access edge computing (MAEC)-enabled FL eliminates the trade-off between the model performance and the energy use of mobile devices by allowing users to send a portion of their local dataset to an edge server for the model training. Due to the edge server's enormous processing capability, the amount of time needed for model training is minuscule[11].

10. Caching and offloading in federated learning in mobile edge computing (MEC)

The MEC system's communication and computing resources, as well as both scenarios of compute offloading and edge caching, are managed by deep reinforcement learning (DRL). The distributed training of these DRL agents is carried out in the following manner using federated learning as a framework:

Significantly lowering the quantity of data that needs to be uploaded over the wireless uplink channel. A response on the part of the mind to cellular network circumstances and the mobile communication environment. adaptation using diverse UEs in a real-world cellular network. protecting the privacy of personal data. Major flaws due to the wireless MEC system communication. Increasing the importance of uplink wireless channels and increasing the amount of training data while taking into account many UEs. Might result in privacy-related privacy mishaps, and training data uploading to edge nodes, or the cloud might be sensitive to privacy. Due to the alteration of training data for privacy reasons, proxy data on the server is less pertinent as compared to on-device data.

Mobile edge computing networks can perform and operate more effectively by using task offloading and data caching. The quality of service and resource utilization of mobile edge computing networks can be improved by concurrently improving task offloading and data caching. Please refer to "**Table 2**", for observations of different factors of computing resources and offloaded data.

Each client has to submit the most recent version of their model within the FL framework. Without the FL framework, UEs would have to use centralized DRL to upload all of the training data via wireless channels, which would use up more communication resources.

A mobile edge system's edge caching and compute offloading scenarios are investigated through tests, and the "in-edge AI" is assessed and shown to be capable of achieving performance that is very close to ideal[12].

Table 2. Observations of different factors between computing resources and offloaded data.

Increase in	Loss	Time consumption	Energy consumption
Computing resources allocation	Decreases	Decreases	Increases
Offloaded data set size	Decreases	Increases	Decreases

The results are as follows, the proposed MAEC-enabled FL performs better than the traditional FL due to the offloading and collaborative training of the local datasets at the edge server. The MAEC-enabled FL can function with less time consumption than the traditional FL thanks to the proposed resource management technique. Although there will ultimately be an energy cap for mobile users, the energy use of cell-center users fluctuates more than that of cell-edge users.

11. Federated learning for wireless communications

The wireless communication network contains a large amount of data, which makes it possible to use ML models to train the data. However, the acquisition of ML is insufficient since wireless communication is more complex there. It is a distinct manner of applying mathematical techniques. As the data is moved from each server to the central or global server and contains a lot of private data, it might provide challenges for the individual company since it is located at the center and has a large quantity of data that could be attacked. The decentralized ML technique, which we called FL, has been introduced to address this problem. The training has been carried out in this decentralized system at each server and the trained data has been sent to the central server. This federated ML is a development of the first federated strategy that Google just released. We used the FL framework for communication across 5G networks. Because they concern performance, privacy, and security.

In contrast to the previous ML, which sent the raw data, this ML trains the data at each local server before sending it to the global server. If the collection has usage, it will transmit the data to the local server after using the global model's local model parameters should be updated. As a result, without explicitly accessing their privacy-sensitive data, each local learner uses the datasets of the aggregator's global model, which is the only way the other students can learn. Wireless communication presents various security and privacy problems[13].

There are characteristics of non-IID (independent and identically distributed), distributed, and unbalanced training data that have been mentioned, including distributed learning, parallel learning, distributed ensemble learning and there are numerous applications for wireless communication that uses federated learning. These are edge computing, caching, spectrum management and 5G core network.

Some of the parameters that we came across are the various literatures which are considered by the researchers in their work. Please refer to "**Table 3**", for factors considered by researchers are as follows:

Accuracy (A): which tells us how closely the sample parameters match the characteristics of the population by providing the near value results to the provided parameters quality.

Latency (L): the delay in time it takes a data packet to go from one network node to another is referred to as latency in networking. A network tool called ping, or a diagnostic command called traceroute are frequently used to monitor latency on the internet.

Time (T): time is what appears to be an unstoppable sequence of existence and events moving from the past through the present and into the future.

Energy consumption (E): energy consumption is the term used to describe the use of all energy to perform an action, build something or simply occupy space.

Reliability (R): reliability is the potential for a product, system or service to perform as intended for a set period of time.

Edge computing (EC): edge computing is a distributed computing paradigm that moves processing and data storage closer to the data sources. This ought to cut down on bandwidth usage and quicken response times. Edge computing is a sort of distributed computing that is not specific to any one technology but is topology and location sensitive.

Federated learning (FL): federated learning is a machine learning technique that trains an algorithm using a number of distributed edge devices or servers that each maintain their own local data samples without sharing them. This strategy is distinct from more traditional decentralized approaches, which usually assume that local data samples be distributed equally and traditional centralized machine learning techniques, where all local datasets are uploaded to a single server.

Environment (Env): there are two types of environments in which we conclude they are known as

- Heterogeneous environment (het): using equipment and operating system software from several suppliers. Computers, operating systems, and databases from many suppliers are often used by businesses.
- Homogenous environment (hom): using software and hardware from the same supplier.

Throughput (TP): throughput is the amount of data that actually moves over a certain period of time. It may also be described as the maximum quantity of traffic that a website or application can manage.

Scalability (S): scalability is the capacity of a system to modify its cost and performance in response to changes in application and system processing demands.

Computational power (CP): computing power is a computer's capacity to do a task quickly and accurately.

Hardware (HWR): data on hardware capacity reveals the communication and computing power of various devices.

Bandwidth (BND): to ensure minimum transmission delay for training traffic, edge computing introduces bandwidth slicing to allow federated learning.

Communication cost (CC): the communication cost of a task is the size of the input to the task and can also be in bytes.

Loss (L): loss of the data happened during different communication rounds.

Table 3. Factors that are considered from different areas of federated learning in edge computing.

References	A	L	T	E	R	EC	FL	Env	TP	S	CP	HWR	BND	CC	L
[1]	✓	✓			✓	✓	✓	Het	✓	✓	✓	✓	✓	✓	
[2]	✓	✓				✓	✓	Het				✓	✓		✓
[3]						✓	✓	Het	✓	✓	✓	✓	✓		
[4]	✓					✓	✓							✓	
[5]	✓		✓			✓	✓		✓					✓	
[6]	✓		✓			✓	✓	Het					✓		
[7]	✓														✓
[8]	✓		✓	✓											
[9]		✓													
[10]			✓	✓	✓										✓
[11]	✓		✓			✓	✓							✓	
[13]	✓	✓	✓	✓		✓	✓	Het		✓			✓	✓	
[14]	✓	✓	✓			✓	✓	Het		✓	✓	✓	✓		
[15]	✓	✓				✓	✓	Het							✓
[16]						✓	✓								
[17]	✓					✓	✓							✓	
[18]	✓		✓			✓	✓						✓		

Table 3. (*Continued*).

References	A	L	T	E	R	EC	FL	Env	TP	S	CP	HWR	BND	CC	L
[19]	✓	✓	✓	✓	✓	✓	✓	Both		✓			✓	✓	
[20]	✓		✓												✓
[21]	✓														

12. Discussion and analysis

Despite its benefits, federated learning also has some drawbacks. For example, it can be challenging to ensure the accuracy and reliability of the model when training with distributed data sources. Additionally, federated learning can require more computing power than a centralized approach, as each device needs to be able to process the updates sent by the server[22–26].

Model aggregation delay is indeed one of the drawbacks associated with federated learning (FL). During the training process in federated learning, the individual edge devices or clients perform local model updates using their respective data[26–28]. These local updates need to be aggregated to create a global model that represents the collective knowledge of all participating devices. This aggregation typically occurs on a central server or in a distributed manner. However, the process of aggregating the local model updates introduces a delay in FL. This delay is primarily caused by factors such as network latency, communication bandwidth limitations, and computational overhead[29–32]. As the number of participating devices increases, the time required for aggregation also increases, resulting in longer model aggregation delays.

Model aggregation delay can have several implications:

- Training efficiency: longer model aggregation delays can impact the overall training efficiency in federated learning. The delay can prolong the time required to complete a training round, leading to slower convergence or slower updates to the global model[33–36].
- Real-time applications: in scenarios where real-time decision-making is crucial, the model aggregation delay can introduce latency that hinders the ability to respond quickly[37–39]. Applications requiring immediate or near-real-time predictions may be affected by the delay in updating the global model.
- Communication overhead: model aggregation involves transferring model updates from multiple devices to a central server or among devices. This communication incurs additional overhead in terms of bandwidth usage and network resources, particularly when dealing with large models or a large number of participating devices[40].

Edge-Fed computing helps to reduce the amount of data that needs to be transferred over the network, reducing latency and improving performance. Edge-Fed computing also allows devices and applications to be more responsive to user needs, as they can process data closer to where it is created. Hybrid federated learning is a new type of machine learning approach that combines traditional federated learning with distributed learning. In hybrid federated learning, data is stored in a single location, but it is shared between multiple different locations. This approach allows for greater scalability and privacy, as well as faster training and inference. Cluster FL has the potential to dramatically reduce the amount of data that needs to be stored and shared among devices for training. By using only local data, it eliminates the need for a central repository of data, which can be expensive and difficult to manage. In addition, dynamic cluster federated learning can improve scalability. By allowing multiple clusters of devices to train a model in parallel, the size of the model can be increased without sacrificing accuracy. Federated SGD is a powerful optimization technique for distributed machine learning that offers many advantages over traditional distributed computing. It allows for more efficient use of resources, reduces communication requirements, and increases data privacy[38]. Additionally, because updates can be made asynchronously, Asynchronous FL can be more efficient than traditional federated learning, as participants can continue training even when others are offline. In order to ensure that the model is able to learn from multiple tasks, it needs to be able to identify and isolate the features that are relevant to each task. This is accomplished by introducing regularization terms into the model, which penalize the model for using features that are irrelevant to the task. Overall, collaborative FL provides an efficient and secure way to share and learn from distributed data without compromising data privacy and security[38,41]. It is particularly useful in scenarios where data is sensitive or distributed across multiple devices. Additionally, it has been shown to improve the accuracy of machine learning models in federated settings.

Finally, from the complete study of literature, we found that federated learning in edge computing gives better results than traditional works.

13. Conclusion and future work

The report highlights that cloud computing can present challenges related to security and privacy. To address these issues, the report suggests integrating edge computing with federated learning. This collaboration enables the achievement of accurate results while preserving data privacy. Caching and offloading techniques, when combined with federated learning, reduce the amount of data transmitted compared to traditional approaches. Caching involves storing frequently accessed data closer to edge devices, while offloading distributes computational tasks between edge devices and centralized servers. This integration improves efficiency by minimizing data transmission requirements. In communication networks, the usage of federated learning through edge computing is found to be more effective than traditional machine learning methods. By adopting federated learning, devices within the network can collaboratively train a shared model while keeping data local. This decentralized approach reduces communication overhead and improves efficiency. In the context of Android malware detection, the use of federated learning through edge computing outperforms centralized approaches. By training the model collaboratively on edge devices while preserving data privacy, the federated approach enhances the accuracy of detecting malware on Android devices. Future considerations include containerization, which enables developers to create and deploy applications faster and with improved security. The implementation of a security framework is recommended to effectively manage cybersecurity risks. Further action is needed to incorporate edge federated learning into current edge computing programming models for better results.

In summary, the report concludes that edge computing integrated with federated learning resolves security and privacy concerns in cloud computing. The report highlights the benefits of caching and offloading, the effectiveness of federated learning in communication networks, the superiority of multi-task federated learning, the performance advantages in Android malware detection and the utility of collaborative federated learning. Future considerations include containerization, security frameworks and further advancements in edge federated learning for enhanced outcomes.

From this work, we conclude that, if we take cloud computing we may get some issues regarding security and privacy. To resolve this type of similar issues edge computing is collaborated with federated learning to get accurate results. It is observed that caching and offloading with federated learning takes very less amount of data size than without federated learning. In a communication network the usage of federated learning through edge computing is more effective than machine learning. Usage of multitask federated learning gives more appropriate results than single task federated learning. In Android malware detection federated learning through edge computing the performance of federated approach gives better results than centralized approach. Collaborative federated learning is another type of federated approach which gives better security and is useful in a large range of systems like in healthcare systems and vehicular systems and in wireless communication. Using federated learning in edge computing has given efficient output compared to other approaches.

In future, we will consider containerization which allows developers to create and deploy applications faster and with more security. Security framework for managing cyber security risks. Action in current edge computing programming model to the edge federated learning for better results.

Author contributions

Material preparation, data collection and analysis, SKM, NSK, BR, B and LT; writing—original draft preparation, NSK, BR, B and LT; writing—review and editing, SKM. All authors have read and agreed to the published version of the manuscript.

Conflict of interest

The authors declare no conflict of interest.

References

1. Abreha HG, Hayajneh M, Serhani MA. Federated learning in edge computing: A systematic survey. *Sensors* 2022; 22(2): 450. doi: 10.3390/s22020450
2. Ye Y, Li S, Liu F, et al. Edge-Fed: Optimized federated learning based on edge computing. *IEEE Access* 2020; 8: 209191–209198. doi: 10.1109/ACCESS.2020.3038287
3. Shi W, Cao J, Zhang Q, et al. Edge computing: Vision and challenges. *IEEE Internet of Things Journal* 2016; 3(5): 637–646. doi: 10.1109/JIOT.2016.2579198
4. Wang Z, Xu H, Liu J, et al. Resource-efficient federated learning with hierarchical aggregation in edge computing. In: Proceedings of the IEEE INFOCOM 2021 Conference on Computer Communications; 10–13 May 2021; Vancouver, Canada. pp. 1–10.
5. Zeng T, Guo J, Kim KJ, et al. Multi-task federated learning for traffic prediction and its application to route planning. In: Proceedings of the 2021 IEEE Intelligent Vehicles Symposium (IV); 11–17 July 2021; Nagoya, Japan. pp. 451–457.
6. Liu J, Xu H, Xu Y, et al. Communication-efficient asynchronous federated learning in resource-constrained edge computing. *Computer Networks* 2021; 199: 108429. doi: 10.1016/j.comnet.2021.108429
7. Qayyum A, Ahmad K, Ahsan MA, et al. Collaborative federated learning for healthcare: Multi-modal covid-19 diagnosis at the edge. *IEEE Open Journal of the Computer Society* 2022; 3: 172–184. doi: 10.1109/OJCS.2022.3206407
8. Wu W, He L, Lin W, Mao R. Accelerating federated learning over reliability-agnostic clients in mobile edge computing systems. *IEEE Transactions on Parallel and Distributed Systems* 2020; 32(7): 1539–1551. doi: 10.1109/TPDS.2020.3040867
9. Li X, Cheng L, Sun C, et al. Federated-learning-empowered collaborative data sharing for vehicular edge networks. *IEEE Network* 2021; 35(3): 116–124. doi: 10.1109/MNET.011.2000558
10. Chen M, Poor HV, Saad W, Cui S. Wireless communications for collaborative federated learning. *IEEE Communications Magazine* 2020; 58(12): 48–54. doi: 10.1109/MCOM.001.2000397
11. Zaw CW, Pandey SR, Kim K, Hong CS. Energy-aware resource management for federated learning in multi-access edge computing systems. *IEEE Access* 2021; 9: 34938–34950. doi: 10.1109/ACCESS.2021.3055523
12. Wang X, Han Y, Wang C, et al. In-edge AI: Intelligentizing mobile edge computing, caching and communication by federated learning. *IEEE Network* 2019; 33(5): 156–165. doi: 10.1109/MNET.2019.1800286
13. Niknam S, Dhillon HS, Reed JH. Federated learning for wireless communications: Motivation, opportunities, and challenges. *IEEE Communications Magazine* 2020; 58(6): 46–51. doi: 10.1109/MCOM.001.1900461
14. Xia Q, Ye W, Tao Z, et al. A survey of federated learning for edge computing: Research problems and solutions. *High-Confidence Computing* 2021; 1(1): 100008. doi: 10.1016/j.hcc.2021.100008
15. Ren J, Wang H, Hou T, et al. Federated learning-based computation offloading optimization in edge computing-supported Internet of Things. *IEEE Access* 2019; 7: 69194–69201. doi: 10.1109/ACCESS.2019.2919736
16. Fantacci R, Picano B. Federated learning framework for mobile edge computing networks. *CAAI Transactions on Intelligence Technology* 2020; 5(1): 15–21. doi: 10.1049/trit.2019.0049
17. Wang X, Wang S, Wang Y, et al. Distributed task scheduling for wireless powered mobile edge computing: A federated-learning-enabled framework. *IEEE Network* 2021; 35(6): 27–33. doi: 10.1109/MNET.201.2100179
18. Shariati B, Safari P, Mitrovska A, et al. Demonstration of federated learning over edge-computing enabled metro optical networks. In: Proceedings of the 2020 European Conference on Optical Communications (ECOC); 6–10 December 2020; Brussels, Belgium. pp. 1–4.
19. Suomalainen J, Juhola A, Shahabuddin S, et al. Machine learning threatens 5G security. *IEEE Access* 2020; 8: 190822–190842. doi: 10.1109/ACCESS.2020.3031966

20. Guo H, Huang W, Liu J, Wang Y. Inter-server collaborative federated learning for ultra-dense edge computing. *IEEE Transactions on Wireless Communications* 2021; 21(7): 5191–5203. doi: 10.1109/TWC.2021.3137843
21. Hsu RH, Wang YC, Fan CI, et al. A privacy-preserving federated learning system for Android malware detection based on edge computing. In: Proceedings of the 2020 15th Asia Joint Conference on Information Security (AsiaJCIS); 20–21 August 2020; Taiwan, China. pp. 128–136.
22. Mishra SK, Sahoo B, Parida PP. Load balancing in cloud computing: A big picture. *Journal of King Saud University-Computer and Information Sciences* 2020; 32(2): 149–158. doi: 10.1016/j.jksuci.2018.01.003
23. Huang X, Li P, Yu R, et al. Fedparking: A federated learning based parking space estimation with parked vehicle assisted edge computing. *IEEE Transactions on Vehicular Technology* 2021; 70(9): 9355–9368. doi: 10.1109/TVT.2021.3098170
24. Zhang J, Zhao Y, Wang J, Chen B. FedMEC: Improving efficiency of differentially private federated learning via mobile edge computing. *Mobile Networks and Applications* 2020; 25(6): 2421–2433. doi: 10.1007/s11036-020-01586-4
25. Mishra SK, Mishra S, Alsayat A, et al. Energy-aware task allocation for multi-cloud networks. *IEEE Access* 2020; 8: 178825–178834. doi: 10.1109/ACCESS.2020.3026875
26. Feng C, Zhao Z, Wang Y, et al. On the design of federated learning in the mobile edge computing systems. *IEEE Transactions on Communications* 2021; 69(9): 5902–5916. doi: 10.1109/TCOMM.2021.3087125
27. Kumar Swain C, Routray P, Kumar Mishra S, Alwabel A. Predictive VM consolidation for latency sensitive tasks in heterogeneous cloud. In: *Advances in Distributed Computing and Machine Learning*. Springer Singapore; 2023.
28. Chen N, Li Y, Liu X, Zhang Z. A mutual information based federated learning framework for edge computing networks. *Computer Communications* 2021; 176: 23–30. doi: 10.1016/j.comcom.2021.05.013
29. Mishra SK, Puthal D, Rodrigues JJ, et al. Sustainable service allocation using a metaheuristic technique in a fog server for industrial applications. *IEEE Transactions on Industrial Informatics* 2018; 14(10): 4497–4506. doi: 10.1109/TII.2018.2791619
30. Yu R, Li P. Toward resource-efficient federated learning in mobile edge computing. *IEEE Network* 2021; 35(1): 148–155. doi: 10.1109/MNET.011.2000295
31. Kumar Mishra S, Kumar Sahoo S, Kumar Swain C, et al. CS-based energy-efficient service allocation in cloud. In: *Advances in Distributed Computing and Machine Learning*. Springer, Singapore; 2023.
32. Zhang J, Chen B, Cheng X, et al. Poisongan: Generative poisoning attacks against federated learning in edge computing systems. *IEEE Internet of Things Journal* 2021; 8(5): 3310–3322. doi: 10.1109/JIOT.2020.3023126
33. Zhang W, Wang X, Zhou P, et al. Client selection for federated learning with non-IID data in mobile edge computing. *IEEE Access* 2021; 9: 24462–24474. doi: 10.1109/ACCESS.2021.3056919
34. Mishra SK, Puthal D, Sahoo B, et al. Energy-efficient VM-placement in cloud data center. *Sustainable Computing: Informatics and Systems* 2018; 20: 48–55. doi: 10.1016/j.suscom.2018.01.002
35. Mishra SK, Sindhu K, Teja MS, et al. Applications of federated learning in computing technologies. *Convergence of Cloud with AI for Big Data Analytics: Foundations and Innovation* 2023; 107–120. doi: 10.1002/9781119905233.ch6
36. Fan S, Zhang H, Zeng Y, Cai W. Hybrid blockchain-based resource trading system for federated learning in edge computing. *IEEE Internet of Things Journal* 2021; 8(4): 2252–2264. doi: 10.1109/JIOT.2020.3028101
37. Mishra SK, Puthal D, Sahoo B, et al. Energy-efficient deployment of edge dataenters for mobile clouds in sustainable IoT. *IEEE Access* 2018; 6: 56587–56597. doi: 10.1109/ACCESS.2018.2872722
38. Makkar A, Ghosh U, Rawat DB, Abawajy JH. FedLearnSP: Preserving privacy and security using federated learning and edge computing. *IEEE Consumer Electronics Magazine* 2022; 11(2): 21–27. doi: 10.1109/MCE.2020.3048926
39. Xiao H, Zhao J, Pei Q, et al. Vehicle selection and resource optimization for federated learning in vehicular edge computing. *IEEE Transactions on Intelligent Transportation Systems* 2021; 23(8): 11073–11087. doi: 10.1109/TITS.2021.3099597
40. Rowe WST, Waterhouse RB. Edge-Fed patch antennas with reduced spurious radiation. *IEEE Transactions on Antennas and Propagation* 2005; 53(5): 1785–1790. doi: 10.1109/TAP.2005.846797
41. Elmezughi AS, Rowe WST, Waterhouse RB. Further investigations into Edge-Fed cavity backed patches. In: Proceedings of the 2007 IEEE Antennas and Propagation Society International Symposium; 9–15 June 2007; Honolulu, USA. pp. 920–923.