iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** IV **Month of publication:** April 2024

DOI: https://doi.org/10.22214/ijraset.2024.60602

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue IV Apr 2024- Available at www.ijraset.com

Advancing Edge Computing with Federated Deep Learning: Strategies and Challenges

Geeta Sandeep Nadella[1], Karthik Meduri[2], Hari Gonaygunta[3], Santosh Reddy Addula[4], Snehal Satish[5], Mohan Harish Maturi[6], Suman Kumar Sanjeev Prasanna[7]
Department of Information Technology, University of the Cumberlands, Williamsburg, KY, USA

Abstract: ***Edge computation (EC) represents a transformative architecture in which cloud computing services are decentralized to the locations where data originates. This shift has been facilitated by the integration of deep learning (DL) technologies, notably in eliminating latency issues commonly referred to as the "echo effect" across various platforms. In typical EC-enabled DL frameworks where data producers are directly involved, it is often necessary to share data with third parties or edge/cloud servers to facilitate model training. This process, however, raises significant concerns regarding synchronization with high data rates, seamless migration, and security, consequently exposing the system to privacy vulnerabilities. These challenges can be addressed through the adoption of Federated Learning (FL), which provides a robust mechanism to mitigate risks associated with data loss, ensure data freshness, and enhance privacy. FL enables the decentralized training of standard neural networks across diverse nodes—including vehicles and healthcare facilities—without transferring local data to a central server. Thus, FL not only enhances privacy but also catalyzes collaborative learning in EC environments, allowing for the optimization of models through multi-peer engagements. Despite the potential of FL, comprehensive evaluations of its implementation and the associated challenges within EC contexts remain scarce. This paper seeks to methodically review existing literature on FL in EC scenarios, proposing practical solutions to unresolved issues. It aims to critically examine the integration of embedded systems and advanced learning methodologies, offering a detailed overview of the requisite protocols, architectures, frameworks, and hardware. This study will further explore the broader implications of information technology on global economic structures and delineate the applications of FL in social marketing, highlighting potential setbacks and future directions. By doing so, this research aspires to foster interdisciplinary linkages among foreign language studies, education, and technology, thereby contributing to the broader discourse in these fields.***

Keywords: Federated Learning; Edge Computing; Deep Learning; Intelligent Edge; Edge-AI; Data Privacy; Data Security; Neural Network

I. INTRODUCTION

Cisco insists that the total number of devices connected to the Internet of Things (IoT) by 2025 could reach 75 billion; this is three times more than the directory of data processing of IoT devices in 2020, which is now about 31 billion (see [1]). The latter drives the internet of people, which presents a massively distributed sensor environment with various sensors at different places for crowd-sensing applications, like intelligent integrations, healthcare, and Unmanned aerial vehicles (UAV). Besides, the IoT involves the key activities and those highly time and quality-driven ones that today come to the limelight via the production infrastructure with super high capacity and reliability.

Nevertheless, thoroughly integrating large-scale, multi-faceted, and faraway data in IoT, giving stable services with specific performance, looks like two presented and impossible options. EC has the potential to facilitate data access closer to the CC service at a fast delivery speed, minimizing latency and saving costs. It is highly resilient and available at any time. In short, exploiting the cloud-posing architecture will probably meet these applications with stringent SLAs and time deliverables.

Furthermore, the EC-Distributed computing logic is a scalable scheme that processes IoT big data and executes it using distributed heterogeneous processing power. DL and EC are the newest and fastest-widening technologies for biomedicine, engineering, and language. An enterprise keeps its database assets and model on powerful servers in those centralized cloud sites. Besides the local edge nodes, the remote cloud does part of the computing. These machines' tasks are vital to frame the local instructions and the distant operations. However, due to the inherent challenges listed below, transmitting all data collected from edge devices to a central data center for training a model over the network is not feasible. Though, being faced with the impediments mentioned below, loading all collected edge device information into a central data pool for the training purpose of the over-the-network model is impossible:

©IJRASET: All Rights are Reserved

1) *Communication Cost:* Absolutely some of the data captured from edge nodes and devices EC might be the correct route; however, transmitting such a large amount of information to a remote server, which is responsible for the encoding and transmission process, might be unnecessary as it could ultimately afford the luxury of taking the route, which will on the contrary, increase the time it might take for the network traffic to be completed. In general, the lack of bandwidth hinders the fast transmission of data and affects the capacity to provide high-quality data. Transcendent cloud connection is another edge emphasized, where every edge device acts as a bridge between its fellow devices and, then again, from the cloud servers to the end users. As a result, the network consisting of a thousand edge devices cannot satisfy real-time, low-latency, and QoS requirements because of the considerable time necessary for long-distance transmission, which is just one of many limitations of the 5th generation of ICT. Thus, a cloud-based system has all the mentioned requirements, and a substitution for it is required.
2) *Reliability:* Clients either strike the internet or another communication network to make the datasets travel to their cloud, which is somewhere far away, that is, the center of a cloud (the cloud server) in a typical centralized model-training architecture. As a result of the reliance on servers and network connections in the root and surrounding network, the number of servers scales up for deep learning neural network training and inference. So, the throughput depends on the devices in the IoT system being able to operate without accessing the network. Nevertheless, the same challenges happened in centralized architecture. The mentioned downside is the server and the client link failure, and their performance is slowed down because the connection is unreliable. This may change management's basic outline and interactive networks' total direction.

II. FUNDAMENTALS OF EDGE COMPUTING AND FEDERATED LEARNING

First, before diving into the questionnaire results, it is necessary to know the utilization of sources of information - Edge Computing and Fog computing. In section one, we start to provide the background and fundamentals of both sources - EC and FL. Then, we proceed with an overview of the recent literature on FL and edge computing.

A. Edge Computing

Edge computing is in the middle of its evolution from cloud onslaught to overflow of edges. Standalone boards or microcontrollers may act as a gateway, and sensors will send and receive the data from these boards to the cloud. This process has made the job easier for humans as they do not need to depend on machines. There are a variety of platforms within this category of digital marketing; some of them are cloud servers, smartphones, wearables, and the Internet of Things. Following the process, instead of the resources that embody large-scale data of the cloud in the data center, a wide range of distributed and blurred networked data sources will be replaced, which are further rich in advanced computing capabilities and impressive edge devices. At the same time, some researchers indicate that because of different factors, the current cloud-based infrastructure does not seem online anymore. The edge data has to be transported over from the cloud server KUDU for further processing, which, generally, these servers are located far from the deployed devices. The researchers will not perceive this situation as effective for time-urgent technicalities such as augmented reality, virtual reality, and the most prominent autonomous vehicle network system.

Moreover, edge devices generate more non-device data than what is used to, and the higher the quality of the data, the higher the need for network bandwidth, which will be a challenge to the cloud-based computing paradigms. Furthermore, the finance communication charges and computing power are needed for the intermediation or process of transmitting the data for storage in the cloud for further processing. One can also cite the fact that in releasing the patient health data, the server will keep a copy in the recovery machines for such cases, and this violates privacy and puts all data at risk of being hacked or forged (25).

Therefore, sides of the network will be better solutions for processing the data because of advancing edge computing. Now, the concepts of MDCCs, cloudlets, MEC (Mobile Edge Computing), and fog computing are under discussion for the evolution of EC (Edge Computing) [26–28, 30], and they are being built at or very near the network edge. Immediately after this, however, unfortunately, the European Commission could not wholly determine the regulations, architecture, and protocols for Energy Communities (EC) [32]. This Edge computing technology group is also commonly called what is its name. Edge computing enables the location of the devices and users to close, where decentralized computing can be possible. This is probably one of the most innovative approaches to connecting the dots for those who long to discover new temperaments. It decreases the amount of data transmission payment, we expect the latency, including the service, to be reduced, cloud computing will be contented with distributed computing, and privacy and security aspects will be considered for improvement.

To summarize, EC does not eliminate the cloud's paradigm but provides supplementary methods for utilizing the cloud [33, 34]; edge computing is a type of cloud computing that has additional characteristics that are more than what is with cloud computing. Edge computing combined with cloud computing has three advantages over cloud computing alone [25]: (1) The supposed edge node's main function is to uplink aggregate traffic from the network, reducing backbone traffic by approximately 80%. As a result of this, the communication that is rendered pointless between Edge and the Cloud can be eliminated; (2) being cloud-enabled eliminates the re-transmittal of data that amounts to a meaningless redo of work; hence, it makes for a supportive backup for Edge This scanty Edge is precarious with limited resources and diverse ethnic complexion. On the other hand, outer networks are placed at the ends of the networks, achieved through nodes used in the edgenetworks.

B. Deep Learning

Academics and researchers in several domains, including computer vision theory [36, 37], healthcare [38], and others, have been interested in Deep learning [43, 44], which does not necessitate costly, manually crafted feature engineering like other ML approaches. Deep learning uses simulated neural networks that mimic brain activity. Following this, a Deep Neural Network (DNN) can contain more than one layer between its input and output layers because it is an ANN. Deep learning networks robotically extracting features from massive datasets was the intended use of the NNs. In subsequent stages, these features are used to classify the input, make judgments, and generate new information based on the provided item information [47]. Neural networks take their cue from the human brain, the primary paradigm for utilizing several layers of logic-based units called neurons.

Additionally, training neural networks is challenging when done independently because the abundance of local minima in their cost functions leads the network to learn how to accomplish that primarily, and it does so at the local minimum value that restricts the network's learning. Before neural networks' recent renaissance, deep learning—one of the hidden layers used to construct more robust ML algorithms—had been widely adopted in the previous decade. Frameworks that are atop neural networks include feed-forward and backpropagation algorithms. A feed-forward network, the most basic type of neural network, consists of three layers: function within a neural network, which is a system that includes an input layer, a hidden layer or layers, and an output layer. The regular feed-forward neural network produces the output after the activation function, which makes the function biased and non-linear at the input. The effectiveness of the model, which is heavily reliant on the activation function selection in the hidden layer, is also critically important. Because of their widespread recognition among DNNs, Sigmoid, Softmax, and Rectified Linear Unit (ReLU) are exemplary delivery functions. Table 1 comprises the mathematical representations of the activation function. The fact that DNNs can have hundreds of hidden layers is one of their defining features. Layers convert Inputs into outputs. Take picture classification as an example. A feature vector generated by a DNN, or deep neural network, may include indices of the vector's highest-ranking items. The loss function of a multilayered DNN seeks to reduce the potential distortion between the target value and the actual value; this function also contains another component known as the error function. Think about a scenario with N features in total (x1, x,..., xN), and each feature has its weighted connection (w1; w2; wN). A neural network activation function is f(). After plugging the features/input quantity and weights into the formula below, we get

$$y = f(\sum_{n=1}^{N} x_i * w_i + b) \quad (1)$$

Table 1. Mathematical expressions of activation functions.

Activation Function	Equation	Range
Linear Function	$f(x) = x$	$(-\infty, \infty)$
Step Function	$f(x) = \begin{cases} 0, & \text{for } x < 0 \\ 1, & \text{for } x \geq 0 \end{cases}$	$\{0, 1\}$
Sigmoid Function	$f(x) = \sigma(x) = \frac{1}{1+e^{-x}}$	$(0, 1)$
Hyperbolic Tangent Function	$f(x) = \tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$	$(0, 1)$
ReLU	$f(x) = \begin{cases} 0, & \text{for } x < 0 \\ x, & \text{for } x \geq 0 \end{cases}$	$(0, \infty)$
Leaky ReLU	$f(x) = \begin{cases} 0.01, & \text{for } x < 0 \\ x, & \text{for } x \geq 0 \end{cases}$	$(-\infty, \infty)$
Swish Function	$f(x) = x\sigma(\beta(x)) = \begin{cases} \frac{x}{2}, & \beta = 0 \\ max(0, x), & \beta \to \infty \end{cases}$	$(-\infty, \infty)$

Fig. 1 Mathematical Expressions of activation functions

DNNs usually reduce (or minimize) loss functions that include MEAN SQUARE ERROR (MSE), CROSS-ENTROPY (CE), and MAE as input. Backward propagation techniques are used. The backpropagation algorithm is the method that dictates how the NN functions.

This is a technique of adjusting the weights of a neural network based on the previous iteration's error rate (or the current epoch'The model becomes more stable by pinning the weights to counter the error and make efficient generalizations such as SGD, calibrated network weights are optimized to give the model its predictive ability. Weight resides in the next iteration's new weight update. This update is obtained by multiplication of gamma to the partial derivative of the loss function L concerning the weight W. Gradient descent iteration is the same gradient descent learning rate as the learning rate of the gradient descent. Using this particular formula. Point 2, the SGD formula no longer has to do with the entire training dataset; rather, it is a minibatch gradient descent.

The formula shown in equation (3), is an average sum of the Gradient Matrices overall B batches, with.

The latter may cause an issue of overfitting because of its slower learning rate, whereas the former helps prevent overfitting by maintaining fast learning rates and using more training data. The full-bayonet mode of GD may slow down the training and batch memorization, making GD unreliable. The course uses backward evaluation to calculate gradient matrices using the input error gradients described by e as ∂L (B) ∂y.

In order to optimize the cost, the numbers of training iterations of two algorithms, both forward and backward propagations, are repeated over multiple epochs. When training an artificial neural network properly, its generalization ability works predictably, making it possible to use the DNN safely to avoid overfitting or misclassification errors, especially when handling the test set. However, these learning algorithms can also be applied to specialists, such as semisupervised, unsupervised, and reinforcement learning models, among others. These are the kinds: CNNs, RNNs, and MLPs mentioned above the topics on which they have already shown efficiency.

C. Federated Learning

One approach to resolving issues with personal data McMahan proposed was federated learning, which he first mentioned in 2021. As previously mentioned, FL relies on a network of edge-computing devices supervised and managed by a central server. These devices work in tandem to train a model using decentralized local datasets and infrastructure. To get the required result, clients store the data in lace and transmit the parameters that must be changed to the server [18]. In addition, the FL reduces the dangers of centralized ML and the costs associated with data collection. FL opens the door to edge computing, which opens the door to communal Deep Learning construction and training for network improvement. Here, we will compare a centralized mathematical model approach with the benefits of applying FL to edge networks. Keeping the control parameters updated independently of the raw processing data allows data owners to reduce the required communication data packages drastically. This results in consistently high levels of network bandwidth use [57]. Finally, we have latency: For instance, data transmission becomes much more delicate in those time-constrained applications [58] (such as real-time media, industrial control, mobility automation, and remote control). Examples of Internet of Things (IoT) applications that benefit from local execution on edge devices include medical applications, event detection systems, and augmented reality systems, all of which require real-time data processing [59]. A very responsive FL system, noticeably quicker than the centralized system, is the end product. Thirdly, anonymity is ensured because raw data is disseminated rather than sent directly to a central server. However, if users stop uploading model data (such as images or audio) to a central server, they can train a better model [60].

FL enables users to work together in a shared environment to train a model using data from several devices without exposing user data to the server. This privacy-preserving collaborative learning technique [61] achieved using the three-step approach indicated in Figure 3: This privacy-preserving collaborative learning technique [61] achieved using the three-step approach indicated in Figure 3: During task setup, the server only considers a small subset of the hundreds of possible devices while working in a time-constrained environment. The training task determines the target device, fuel consumption, and air quality. The target device can be an automobile where all the mentioned actions are carried out. Finally, the server will assign the model and training procedure-related over-talking parameters, such as the learning rate. One of the first things this method does is take care of server initialization using weight initialization methods like random, He, or Xavier [3]. Selected participants will receive the parameter server-defined huge task and the global model w 0 G via their chosen devices.

Local model training: Participants use their data and resources to update the local model parameters w t i. They start with the current iteration global model w t G, where t is the current iteration step. Client's goal at iteration t i is to find the regulator w t_i that minimizes the loss function value at time t_i, denoted as L (w t_i) as per [50].

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue IV Apr 2024- Available at www.ijraset.com

$$w_i^t = \arg\min_{w_i^t} L(w_i^t)$$

Fig. 2 Formula to find optimal model parameters

Finally, each local model's updated parameters are sent again back to the FL parameter server; • Global model aggregation: The local Gradient Descent is performed on every local copy of the model by each participant, which then sends back the gradient information as well as their local model parameters to the centralized server, which in turn compute the global update of the model parameters, denoted as w t+1 G, based on the global loss function, L(w t G), and then send it back to all the participant models to minimize the global loss function.

$$L(w_G^t) = \frac{1}{N} \sum_{n=1}^{N} L(w_i^t)$$

Fig. 3 Formula to minimize the loss function in machine learning

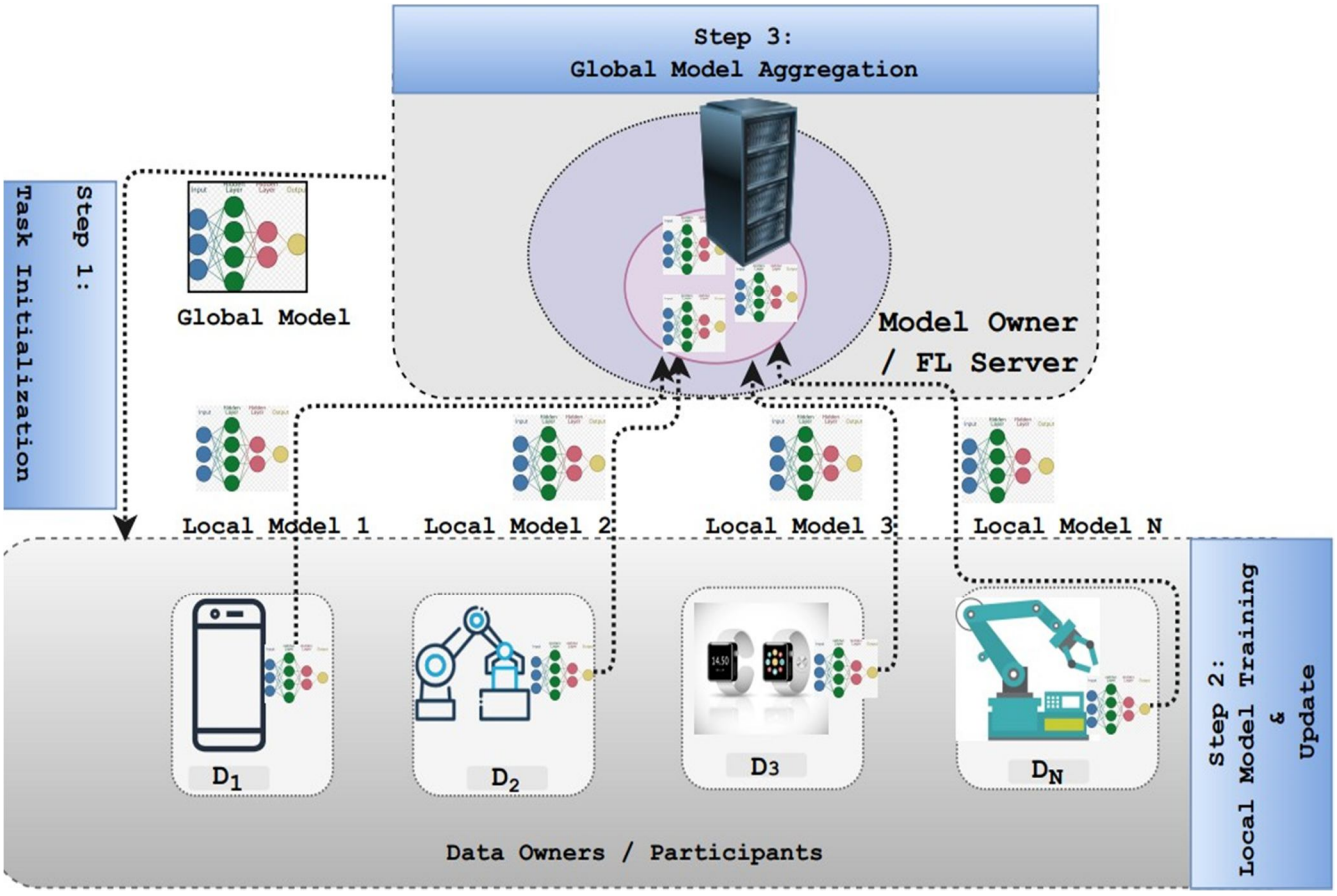


Fig. 4 An FL general process with the server and N participants synchronizing with a server and updating it with an updated global model in a single communication round

III. LITERATURE REVIEW

The following is a discussion of the opinion polls that researchers in CFL have carried out in the EP case. The charts in the summary will help readers understand the differences and results compared to the previous surveys. In addition, the submission of this paper is covered in a concise sentence in the next part.

©IJRASET: All Rights are Reserved

A. Related Works

As far as we know, no other systematic research has focused on FL in European countries (EC). However, there is a dearth of research combining the two fields, and the surveys on both EC and FL are well-developed. Another problem has been the hardware requirements, which have been completely disregarded in the current literature. The authors' work on "MEC and architecture with computational offloading" is available in a published paper at [8]. The integration of intelligent edges and edge intelligence was covered in both [25]. Additionally, authors Nguyen et al. from the DRL field investigated new issues in networking and communications. The authors provided examples of fog computing-related computing paradigms, traced their history, highlighted present difficulties, and predicted future developments [33]. In addition, the authors of [6] highlighted the interdependence of MEC's wireless communication and mobile computing resources through MEC administration. Furthermore, in the [63] article, researchers have also looked at edge intelligence's specific frameworks and architectures. Researchers looked at the works of [edge intelligence] writers who worked on [6G] networks in [64]. In addition, Cui et al. [65] looked into ML's applications in IoT management. In particular, they examined computing offloading tactics in edge computing for smartphones [66]. In addition, the researchers conducted [67] to uncover approaches for computational offloading. Abbas and his colleagues [68] study MEC's software and hardware developments. In addition, the authors included the computation loading modeling in their discussion [80]. Furthermore, the subjects of computation, storage, and communication inside the MEC system were investigated by the writers in [70]. Furthermore, Yao et al. [71] illustrated the L3 cache and the differences between several caching algorithms. Additionally, a follow-up survey on MEC for the 5G network design was provided in the [27] chapter. However, there has been no evaluation of the submitted papers [25, 33] or work [64, 66, 69–73] for FL. In addition, there are no SLRs among the available literature reviews (n=12,71). In addition, the infrastructure issues not stated in [25] were not picked up in the questionnaires. Investigating the potential threats to data privacy and security from a FL perspective was highlighted in the study by Mothukuri et al. (2022, p. 129). Many unanswered data privacy and security questions exist, and Mothukuri et al. (2022, p. 129) examine these questions. The writers of Reference [60] investigated the impact of FL on applications in industrial engineering and computer science. Additionally, they highlighted the latest advancements in asynchronous training and gradient aggregation. They discussed six research fronts in their definition of FL, including returned model verification, block-chain-based FL, and federated training for unsupervised ML. In addition, they brought attention to new research issues and potential solutions that could work together in the future. Furthermore, Zhang et al. [61] covered five angles in their discussion of previous FL studies: data partitioning, data privacy (which is crucial and needs attention), machine learning (ML) models based on relevance, communication architecture, and heterogeneity solutions. The indicated elements also include future academic research and the vestigial organs. Additionally, data distribution, ML model, privacy method, communication architecture, federation size, and participation motivation are the six elements of the FL system that Li et al. [62] suggested as a taxonomy of FL building blocks. All these things stand for study design considerations, numerous case study reports, and potential avenues for further investigation. For instance, in those mentioned above, the writers sought out a thorough FL lesson, highlighted key study areas, and provided a thorough and critical review of the pertinent literature. There were four main obstacles to FL that they took into account: communication speed, system heterogeneity, statistical heterogeneity, and integrity. Also, the research conducted by Kairouz et al. [18] highlighted the current state of the art and the significant issues and questions that academics had uncovered. Some of the technical challenges that have been brought up include issues with communication efficiency, data variability, privacy, and model aggregation. Florida is on the verge of becoming popular, as pointed out in [53]. In their presentation of FL system design, the authors [54] drew a skyline of FL evolution. The paper also highlighted various FL structures and their associated classifications for training data organization based on data distributions [17]. Pages 50–11 out of 45 in Sensor,s 2019, Volume 22, Issue 4, In discussing FL's role in smart city sensing, the authors identified several unanswered questions, obstacles, and potential solutions [55]. In contrast, Liu et al.'s study report [86] provides an example of an evaluation of FL's security risks and vulnerabilities; the same authors also reviewed all of FL's security defense strategies typically employed. However, these writers saw that using two or more defense strategies is advisable as one cannot identify all the attack modalities. Industries not well-versed in data privacy issues would not have had access to these technologies had Reference [67] outlined the processes and paths that would have allowed them to be developed. Furthermore, the software engineering viewpoint on FL was also the subject of a systematic review and literature (SLR) by Lo et al. [89]. The writers of the articles [17–19, 50–59, 61] failed even to consider the potential effects on the infrastructure when they emphasized the importance of establishing the FLOM. Additionally, they hindered the effectiveness of UMF-related FL-to-EC translations. Even while they used the SLR method for diseases (52, 59), they did not proactively use it for investigations.

B. Research Contributions

However, the number of surveys and studies [17–19,61,80–100] [17–19, 61] conducted to investigate federated learning in edge computing is numerous, which shows the issue's importance. Nevertheless, no systematic review of FL in EC has been done. Having a taxonomy with a delineation for advanced methods and works as part of the current problem, here is a systematic review of the FL implementation in the EC environments. To show the range of what the FL adoption in the EC model can lead to in the future, we conduct a Systematic Literature Review (SLR) mapping the existing efforts' differences, breakthroughs, and directions. Furthermore, to our understanding, this is the first survey of the state-of-the-art literature considering the installation possibility of FL implementation and client-side hardware requirements in space-limited resources of edge computing environments. First, we discuss the main principles of the support ecosystem and the fundraising enabling environments, and then we will review the literature surrounding the need to foster entrepreneurship through fundraising. On top of the methodology and the model, we provide the details on protocols, architecture, framework, and hardware designed for FL execution in an EC environment. Moreover, we also consider the areas of implementation, difficulties, and current solutions for edge-enabled FL. The last section illustrates how FL can be implemented, along with two case studies from edge computing and recommendations for further research in this field.

IV. RESEARCH METHODOLOGY

The SLR is indeed a renowned process that is taken into consideration for papers that have been published through credible resources. Essentially, SLR stands behind the theory of developing questions the researcher taxes and provides with the responses. Thus, this paper explores the up-to-date titles that can be considered applications of the SLR research methodology. Multiple search techniques were utilized to subsample the original test results, involving manual and automatic methods. We have done a judicial examination of federated learning being implemented in an edge computing environment through the lens of its result, support, application, and implications. However, the primary studies were scrutinized for quality purposes to select studies of the best quality. We adopted a procedure, both a backward and forward snowball, to produce the most relevant results. Researchers should conduct the study with a specific protocol to create bias-free research. The fact that SLR summarizes all research involved and then identifies existing gaps helps in the already ongoing research and delving deeper into new phenomena. Through the methodology development described in the figure, we established the research framework used in the SLR, which comprised five stages. The research questions were assembled first, then the sources and search methods were determined. Step 4 entailed the selection of the results that were included and those excluded from the consideration. The process of differentiating the research objectives and the breakdown of the research papers were explained in the fourth step, while the fifth step consisted of synthesizing the information gathered. Excel is not the only software used for the analysis phase; it will include other software, which will be mentioned later.

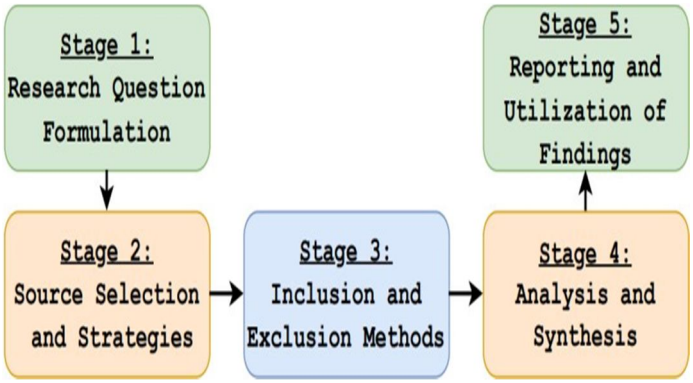


Fig. 5 Five-stage research model

A. Research Question

Survey RQs that might arise during the project can be related to the role of digital technology in the new EC to solve if necessary.

RQ-1: What are the future directions and open research areas for advancing the integration of Federated Learning and Edge Computing to enable secure, efficient, and collaborative intelligence at the edge?

RQ-2: What are the future directions and open research areas for advancing the integration of Federated Learning and Edge Computing to enable secure, efficient, and collaborative intelligence at the edge?

B. Analysis and Synthesis

The stage of collecting and analyzing the literature part, which has the same scope as FL implementation, design, and applications in the EC method, was provided. This scenario contained six classes or challenges (among them communication efficiency, privacy and confidentiality, client selection and scheduling, homogeneity, and price of the service) listed in Fig. 5. In addition, we show the total number of articles relevant to the FL (i.e., 2019) just with the Figure 6. Moreover, in the same concern, Fig.7 illustrates the number of nanotechnology-related publications from generic subject bases in different years. The graph below illustrates the full number of publications in the subject area. Lastly, the last graph shows that these journal articles have focused on different subject areas.

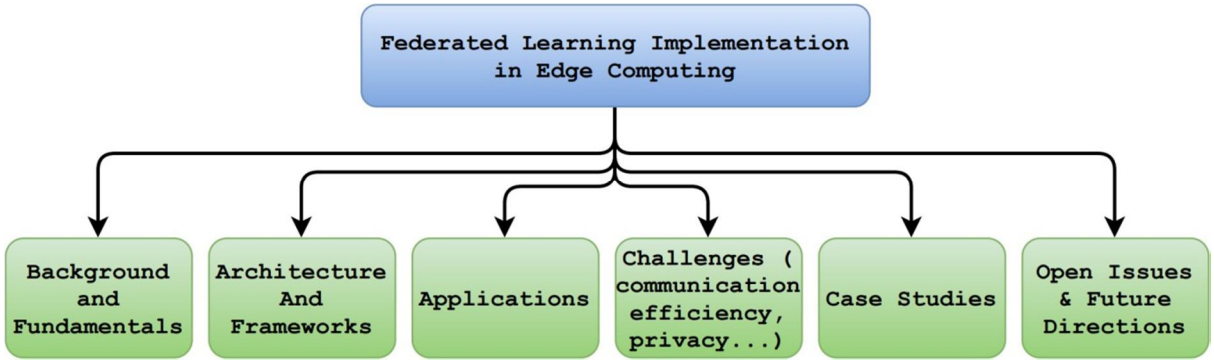


Fig. 6 Taxonomy of federated learning in EC papers based on the high-level classification to be analyzed in this survey

V. TOTAL NUMBER OF ARTICLES ON FEDERATED LEARNING IN EDGE COMPUTING VS PUBLICATION YEAR

This paragraph explains the EC (EC) perspective about the FL architecture. For the next part of this lesson, we move to the FL protocols and how to set them up. Besides, the hardware demands of FL edge computer actives are also addressed. Ultimately, we will discuss the challenge and state-of-the-art solutions for FL in the context of epidemic control. Figure 5 (see below) describes how FL applications could be classified under the framework of the EC paradigm. The classification is based on: (i) Section 2 comprises the background and fundamentals, which have already been covered (ii) This section focuses on the Edge computing framework that introduces FL applications; (iii) this section highlights implementation challenges for FL in the edge computing paradigm, (iv) Section 4 presents FL architectures and frameworks with Edge computing environment, as well as (v) edges computing case studies in section 6.

For sensor network 2022, the model with 450 is the 22 best solution. Federated Learning in Edge Computing: Technologies, Architectures, and Frameworks A good starting point is establishing a network protocol to obtain a deeper idea of how the system is structured. The authors of [14] developed a lower-level FL process to better the whole system's performance. The communication protocol considers all the interactions the machine learns during the FL training process. However, it also has the subject of communication interruption between the server and sensor, including security and communication, availability, and message processing order. The FL system includes a server that is a cloud-based distributed service (the FL server), and the devices at the endpoints (e.g., phones) participate in the protocol. At this step, the device tells the server that it is allowed to use acquired FL population information to do an FL operation. Subsequently, the server will operate using the population information collected from the device. Problems and applications are represented by their distinct global name and are part of the entire FL (Flat List of entries presented hierarchically) population. Tasks for the FL population may include model finalization with the local data introduction or testing the trained models against the provided data. In a prespecified time interval, the server usually operates with a few devices out of the huge community to which the server is connected, ranging around a few hundred devices. This part is the segment that deals with the language task (FL task). This game has a classic feature that involves the interaction of apparatuses and the server; we call it a round. Throughout the session, they maintain a virtual connection to the server. The Master tells the selected devices to do certain computations. Through the specification of a Tesla graph or its execution, this goal is accomplished with the aid of TensorFlow JS. Once the round has started, the server notifies each player of the FL checkpoints, together with the most up-to-date global model's settings and any other relevant information. While working in Florida, each participant updates the server with a checkpoint that includes local data and the current global state.

Parameter server processes typically change the global state of the model server, which FL runs on. Each training round has three phases, as shown in Figure 9, which include the communication technique used to construct a global singular population model: Each training round has three phases, as shown in Figure 9, which include the communication mechanism used to construct a global singleton population model:

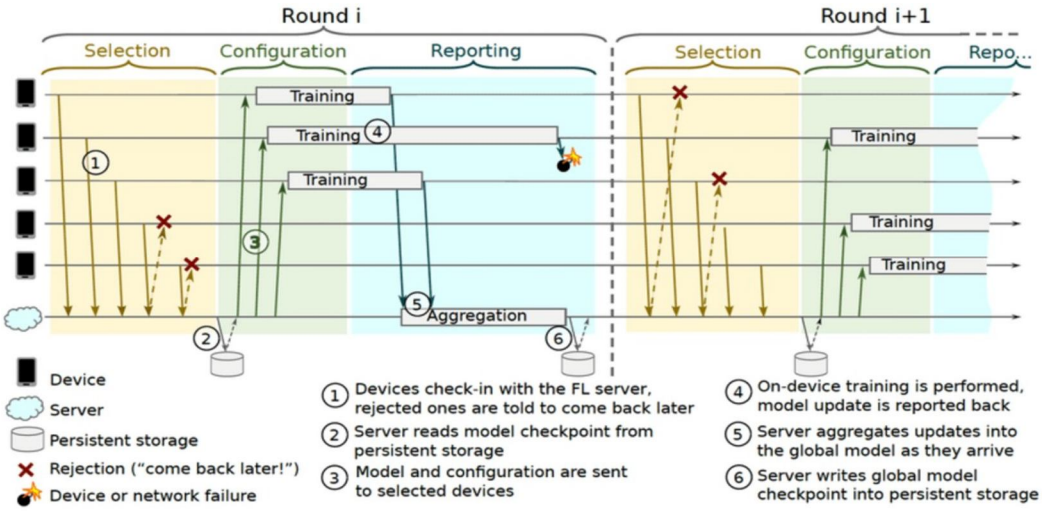


Fig. 7 Federated learning protocol [14]

A. *Hardware Requirements*

The DL employs many advanced duties previously unthinkable, such as image classification/object detection, audio/speech recognition, and anomaly detection, which demand many computer resources. In contrast, edge devices possess only modest computational abilities, occupy minimal memory footprints, and consume little power. In order to make sure FL is not putting an extra load on EC, it is necessary to do thorough research on EC's recent hardware requirements and what should be corrected. We were motivated by this idea, so we conducted a short research on the types of EC hardware that may be needed, as shown in Table 3. Harmony evolves from the comparison of hardware accelerators for the execution of federated learning in edge computing.

The leveling up of DNN-related collective training on an edge network structure, including FL, has contributed to the need for the fast development of hardware to hold the computational workload and storage. They are such enhancements as expansion in minimum component processing units (CPUs), graphical component processing units (GPUs), and new applications, particularly integrated circuits (ASICs), that are purposed to ensure DNN calculation. The complexity of NN is high, and the depth of this model is also high. For the training and testing of all these operations, including billions of them and millions of parameters, tremendous computing resources are needed. Such a feature is a computing problem for General-Purpose computer machines (GPP). As a result, hardware accelerators to improve the performance of the DNN model can be explored by categorizing them into three edge hardware types for FL; subsequently, exploring options for hardware accelerators can be done by considering the three edge hardware types for FL:

1) *GPU-based Accelerators:* The GPU presents itself as the multi-faceted general-purpose processor capable of operating as many vector units at once, thus providing tensor cores that can accomplish so many operations simultaneously, which in turn helps speed up the throughput.

Name	Owner	Pros	Cons
CPU/GPU	NVIDIA and Radeon	High memory, bandwidth, and throughput	Consumes a large amount of power
FPGA	Intel	High performance per watt of power consumption, reduced costs for large-scale operations, excellent choice for battery-powered devices and on cloud servers for large applications	It requires a significant amount of storage, external memory and bandwidth, and computational resources on the order of billions of operations per second
ASIC	Intel	Minimizes memory transfer, most energy efficient compared to FPGAs and GPUs, and best computational speed compared to FPGAs and GPUs	Long development cycle, Lack of flexibility to handle varying DL network designs

Fig. 8 Advantages and Disadvantages of different types of processors

2) *FPGAs-based Accelerator:* Diffusion of superior neural network performance. Furthermore, let's take the example of the NNV Deep Learning Compute Library. We conclude that the library incorporates the Integral, Voltari, and Haswell 1000 series GPU accelerators and also provides hardware support for deep learning training (NLP-L 1000) and inference processing (NLP-I 1000). The excitement of the latest generation of mobile phones and tools, such as wearable devices and surveillance cameras, comes with the distribution of DL apps, one of which is deploying them, and a close venue is even more useful. Regarding DL Computation, there are a variety of techniques available, like cloud mobile devices, except that the low power, small ROM, and calculation facilities limit them. In order to deal with the disadvantages of human-vision-based surveillance around the edges, several researchers have proposed the development of efficiency-improving GPU accelerators. Brands of ARM microcontrollers, such as the Cortex-M series, represent an interesting type of microcontrollers. In this way, we will solve this problem using CMSIS-NN as the named NN units. Memory requirements of NNs on a sensor of AM 2022, Cortex-M families of processors have been reduced by CMSMIS-NN 22X compared to the standard, allowing the DL model to be deployed even inside IoT devices and keeping the output efficiency and power consumption.
3) *ASIC-based accelerator:* FPGA-based accelerator: The regular type of old-school GPU model utilization, which is considered to be most appropriate for DL in the public cloud infrastructure, is considered to be the only way how to address the needs of some number of industrial applications since the power and costs are considered to be the only constraints for those. Next, edge nodes must deploy large computer tasks that occupy CPUs or GPUs for a long time to maintain their service orchestration capabilities. Consequently, the hardware built for edge computing brings together with distributed FPGA implementation of DL. Unlike supercomputer architecture-based GPUs, which are more power efficient but with less performance, they are suitable for completing machine learning (DL) tasks in the edge computing environment. Hence, FPGAs are climbing the AI implementation ladder to displace the current market leader GPUs. The next contender would be AI Mark 2. Traditionally, FPGAs are considered approximately ten times more power efficient than ASICs, influenced by the research of Microsoft Research Institute's Catapult Project (Patton: 2009). Among several solutions of FPGA-based accelerators, Project Brainwave is one of the most influential to have been initiated by Microsoft. Its (It is) hard and soft centralized systems are equipped with DNN engines accelerated with Stratix FPGAs that can continuously process real-time and delay-sensitive artificial intelligence applications [27]. While this does exacerbate several issues, composing a gigabyte of space had better memory, huge bandwidth consumption, and billions of operations every second. Hence, along with the thorough realization of the local dataset's sufficient storage, there will be a need for educated research in the local implementation of FL systems based on the capabilities of FPGA edge devices.

B. Applications in an Edge Computing Environment

The phrase FL in the above paragraphs means the same thing as an effective collaborative learning tool that integrates over a global network. The most important feature of the FL is the level of decentralized decision-making provided, along with the possibility for users to customize and distribute their tasks across all peers is important to generate an ecosystem of applications in the EC field. The subsequent part of this paper allows us to focus on the FL application's main mechanisms that may help to overcome environmental issues. Notwithstanding many studies revealing that FL applications involving accurate time, like cancer detection [73] and COVID-19 detection [74, 75] applications, are turning out to be the focus, this review mainly dwells on three main FL applications in EC, described below.

1) *Computation Offloading and Content Caching:* We will overcome the pole between edge computing power and the required computing force at edge devices by providing intelligent features at the Edge. It involves QoS and the content delivery service (CND). Consequently, EC incorporates the computation resources everywhere in the network, connecting all nodes using acceptable computation offloading algorithm, thus enabling the Sensors 2022, 22, 450 24 intelligent decision-making to sensor, compute, or not to cache; determine when and how to offload, which has a direct impact on the system performance and affect QoS and QoE. In the VE environment, for the past decade, researchers and engineers have been implementing intelligent content caching and computation offloading strategies using DL with many research papers on this subject matter [3, 3, 7, 7, 7].

C. Malware and Anomaly Detection

IoT devices have become a visited target for cyber-attacks, while malware has become more and more in numbers and has more and more functions with the rapid growth of IoT applications. This makes the task of threat detection and analysis harder. The primary difficulty of implementing a working malware detection system comes from the vast and fast-growing space of malware samples, which vary in syntax and behavior.

Henceforth, the most critical steps regarding the preclusion and remedy of the most severe problems in the edge computing environment are learning different types of attack and malware development, or cyber-attacks, and anomaly detection (including the improvement) [142]. There is a host of malware and anomaly detection, and steps to counter it from the other environments have been proposed. A deep learning-aided cyber pervasive-attack detection approach [43] was proposed as per the Gaussian naive Bayes classifier model.

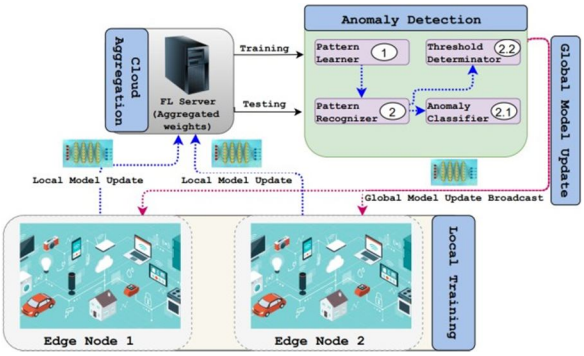


Fig. 9 Federated learning system workflow

To train [46], it was shown that CNN could significantly contribute to accelerating the feature extraction process from data, which are used to fulfill the required detection for accuracy. A version of attack detection models, FL-based, has been developed for embedded networks to overcome the issues with conventional attacks Flocality, cloud aggregation, and anomaly detection; globally, participants forward their trained local samples to the FL server after the global model has been trained. The server provides all the computers with the global model, and the data gets updated on each personal computer as the server collects the data. The catering procedure is repeated until the desired level of performance is reached; this human energy management will be equipped with a new global model that the anomaly detection engine can be used independently for specific data for

VI. CONCLUSION

Federated Learning fits perfectly as it benefits from edge servers' streaming capabilities alongside edge devices that are data producers' distributed designs. DL algorithm makes the typical structure of network optimization for edge computing DL possible. Accordingly, it can be referred to as EDGE, an emerging disruptive graphics engine. Here, we present a systematical literature review of FL in the context of the event cognition paradigm. It comprised 200 initial sources. Similarly, we received more attention to the challenges in FL design at the Edge, including inherent communication and computation inefficiency, specification heterogeneity, performance requirements, accuracy, data collection and device selection, and service pricing. We looked deeply into the challenges above and compiled a comprehensive roundup of the latest advancements and alternative approaches. We specified two case studies (innovative healthcare and UAV, for example) that illustrate the application of FL in e-commerce. Summarizing this, we addressed the open questions and practical case study directions on devoting the learning process to the sFL into the EC paradigm, with students preferably taking an active part in the research sphere.

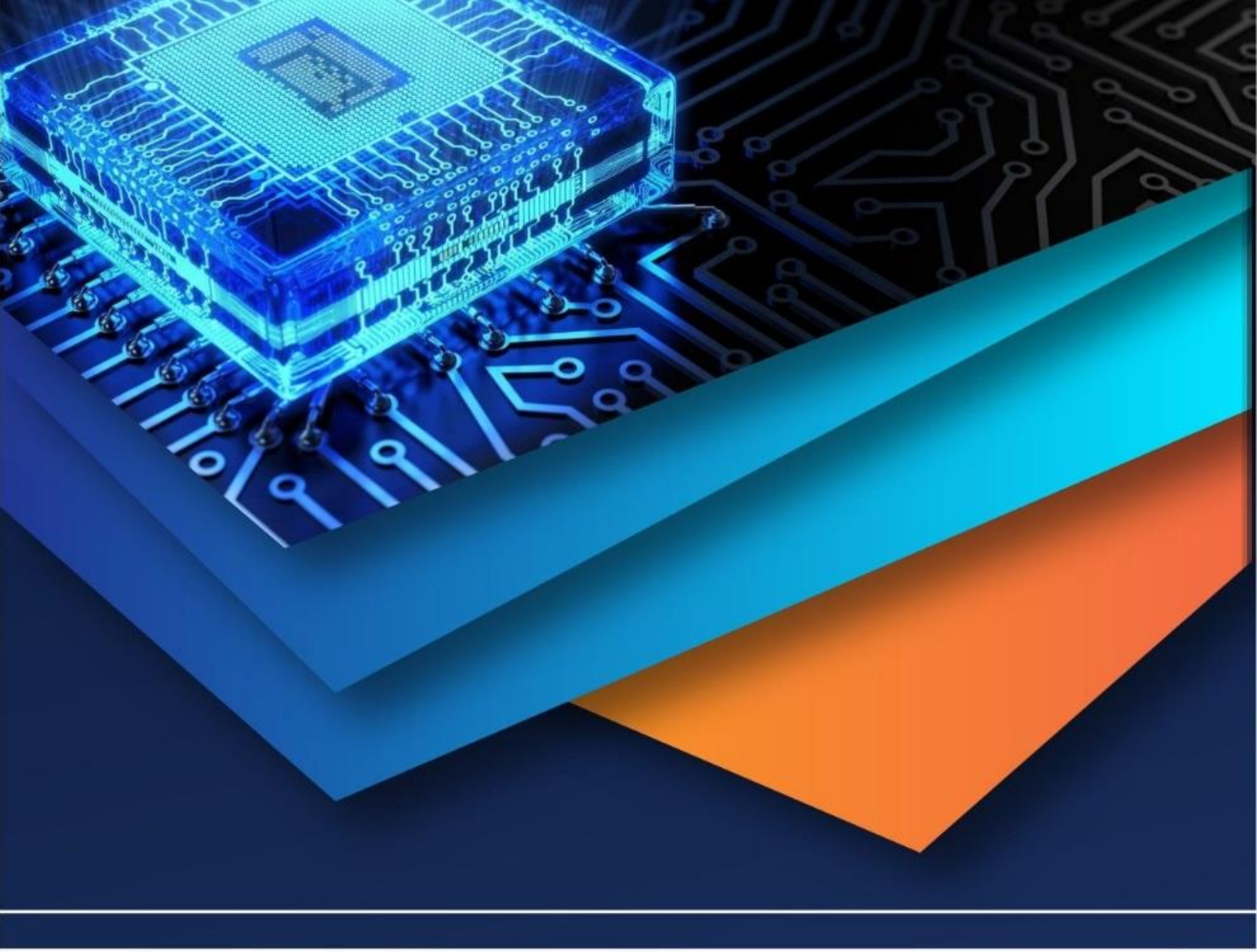
REFERENCES

[1] pp. 20–20, 2020. [Online]. Available: https://www.cisco.com/c/en/us/solutions/internet-of-things/future-of-iot.html
[2] R. Basir, S. Qaisar, M. Ali, M. Aldwairi, M. I. Ashraf, A. Mahmood, and M. Gidlund, "Fog Computing Enabling Industrial Internet of Things: State-of- the-Art and Research Challenges," Sensors, vol. 19, pp. 4807–4807, 2019.
[3] R. Pryss, M. Reichert, J. Herrmann, B. Langguth, and W. Schlee, "Mobile Crowd Sensing in Clinical and Psychological Trials-A Case Study," in Proceed- ings of the 2015 IEEE 28th International Symposium on Computer-Based Medical Systems, 2015, pp. 23–24.
[4] L. Liu, C. Chen, Q. Pei, S. Maharjan, and Y. Zhang, "Vehicular Edge Computing and Networking: A Survey," Mob. Netw. Appl, 2021.
[5] A. H. Gebreslasie, C. J. Bernardos, A. D. L. Oliva, L. Cominardi, and A. Azcorra, "Monitoring in fog computing: State-of-the-art and research challenges," Int. J. Ad Hoc Ubiquitous Comput, vol. 36, pp. 114–130, 2021.
[6] Y. Mao, C. You, J. Zhang, K. Huang, and K. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," IEEE Commun. Surv. Tutor, vol. 19, pp. 2322–2358, 2017.
[7] P. Mach and Z. Becvar, "Mobile Edge Computing: A Survey on Architecture and Computation Offloading," IEEE Commun. Surv.Tutor.
[8] L. Yann, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, pp. 436–444, 2015.
[9] P. Corcoran and S. K. Datta, "Mobile-Edge Computing and the Internet of Things for Consumers: Extending cloud computing and services to the Edge of the network," IEEE Consum. Electron. Mag, vol. 5, pp. 73–74, 2016.

[10] M. Chiang and T. Zhang, "Fog and IoT: An Overview of Research Opportunities," IEEE Internet Things J, vol. 3, pp. 854–864, 2016.
[11] P. Voigt and Avdb, The EU General Data Protection Regulation (GDPR). Cham, Switzerland: Springer International Publishing AG, 2017.
[12] pp. 20–20, 2021. [Online]. Available: https://oag.ca.gov/privacy/ccpa
[13] pp. 20–20, 2012. [Online]. Available: https://sso.agc.gov.sg/Act/PDPA2012
[14] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, and J. Roselander.
[15] J. Konecvný, H. B. Mcmahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, 2016.
[16] B. Mcmahan, E. Moore, D. Ramage, and S. Hampson, "Aguera y Arcas, B. Communication-efficient learning of deep networks from decentralized data," Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, vol. 54, pp. 1273–1282, 2017.
[17] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," ACM Trans. Intell. Syst. Technol, 2019.
[18] P. Kairouz, H. B. Mcmahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, and R. Cummings.
[19] T. Li, Sahu, A. A. Talwalkar, and V. Smith, "Federated Learning: Challenges, Methods, and Future Directions," IEEE Signal Process. Mag, vol. 37, pp. 50–60, 2020.
[20] T. Nishio and R. Yonetani, "Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge," in Proceedings of the ICC 2019-2019 IEEE International Conference on Communications (ICC), 2019, pp. 1–7.
[21] L. Liu, J. Zhang, S. Song, and K. B. Letaief, "Client-Edge-Cloud Hierarchical Federated Learning," in Proceedings of the ICC 2020-2020 IEEE Interna- tional Conference on Communications (ICC), 2020, pp. 1–6.
[22] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," IEEE Internet Things J, vol. 3, pp. 637–646, 2016.
[23] X. Hou, Y. Lu, and S. Dey, "Wireless VR/AR with Edge/Cloud Computing," in Proceedings of the 2017 26th International Conference on Computer Communication and Networks (ICCCN), 2017, pp. 1–8.
[24] R. Hussain and S. Zeadally, "Autonomous Cars: Research Results, Issues, and Future Challenges," IEEE Commun. Surv. Tutor, 2019.
[25] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of Edge Computing and Deep Learning: A Compre- hensive Survey," IEEE Commun. Surv. Tutor, vol. 22, pp. 869–904, 2020.
[26] M. Aazam and E. N. Huh, "Fog Computing Micro Datacenter Based Dynamic Resource Estimation and Pricing Model for IoT," in Proceedings of the 2015 IEEE 29th International Conference on Advanced Information Networking and Applications, 2015, pp. 687–694.
[27] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for vm-based Cloudlets in mobile computing," IEEE Pervasive Comput, vol. 8, pp. 14–23, 2009.
[28] W. Hu, Y. Gao, K. Ha, J. Wang, B. Amos, Z. Chen, P. Pillai, and M. Satyanarayanan, "Quantifying the Impact of Edge Computing on Mobile Applications," in Proceedings of the 7th ACM SIGOPS Asia-Pacific Workshop on Systems, 2016, pp. 1–8.
[29] 2021. [Online]. Available: https://www.etsi.org/technologies/multi-access-Edge-computing
[30] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, 2012, pp. 13–16.
[31] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog computing: A platform for internet of things and analytics," in Big Data and Internet of Things: A Roadmap for Smart Environments. Springer, 2014, pp. 169–186.
[32] K. Bilal, O. Khalid, A. Erbad, and S. U. Khan, "Potentials, trends, and prospects in edge technologies: Fog, cloudlet, mobile Edge, and micro data centers," Comput. Netw, vol. 130, pp. 94–120, 2018.
[33] A. Yousefpour, C. Fung, T. Nguyen, K. Kadiyala, F. Jalali, A. Niakanlahiji, J. Kong, and J. P. Jue, "All one needs to know about fog computing and related edge computing paradigms: A complete survey," J. Syst. Archit, vol. 98, pp. 289–330, 2019.
[34] B. A. Mudassar, J. Ko, and S. Mukhopadhyay, "Edge-Cloud Collaborative Processing for Intelligent Internet of Things: A Case Study on Smart Surveil- lance," in Proceedings of the 2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC), 2018, pp. 1–6.
[35] A. Homepage, 2021. [Online]. Available: https://www.alibabaCloud.com/blog/extending-the-boundaries-of-the-Cloud-with-Edge-computing_594214
[36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet Classification with Deep Convolutional Neural Networks," in Proceedings of the NIPS, 2012, pp. 1097–1105.
[37] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," Proceedings of the 31st International Conference on Machine Learning, vol. 32, pp. 647–655, 2014.
[38] R. Miotto, F. Wang, S. Wang, X. Jiang, and J. T. Dudley, "Deep learning for healthcare: Review, opportunities and challenges," Brief. Bioinform, vol. 19, pp. 1236–1246, 2018.
[39] N. Justesen, P. Bontrager, J. Togelius, and S. Risi, "Deep Learning for Video Game Playing," IEEE Trans. Games, vol. 12, pp. 1–20, 2020.
[40] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, 2013.
[41] B. Kisacanin, "Deep Learning for Autonomous Vehicles," in Proceedings of the 2017 IEEE 47th International Symposium on Multiple-Valued Logic (ISMVL), 2017, pp. 142–142.
[42] R. Qing and J. Frtunikj, "Deep learning for self-driving cars: Chances and challenges," in Proceedings of the 1st International Workshop on Software Engineering for AI in Autonomous Systems, 2018, pp. 35–38.
[43] D. Li and Y. Liu, Deep Learning in Natural Language Processing. New York, NU, USA: Apress, 2018.
[44] V. Sorin, Y. Barash, E. Konen, and E. Klang, "Deep Learning for Natural Language Processing in Radiology-Fundamentals and a Systematic Review," J. Am. Coll. Radiol, vol. 2020, pp. 639–648.
[45] D. Kumar, P. Pawar, Hari Gonaygunta, and S. Singh, “Impact of Federated Learning on Industrial IoT - A Review,” International journal of advanced research in computer and communication engineering, vol. 13, no. 1, Dec. 2023, doi: https://doi.org/10.17148/ijarcce.2024.13105.
[46] J. Schmidhuber, "Deep Learning in Neural Networks: An Overview," Neural Netw, vol. 61, pp. 85–117, 2014.
[47] G. Ian, Y. Bengio, A. Courville, and Y. Bengio, Deep Learning, vol. 1, 2016.
[48] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in Proceedings of the International Joint Conference on Neural Networks, vol. 1, 1989, pp. 593–605.
[49] H. Geoffrey, N. Srivastava, and K. Swersky, 2012.

[50] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y. C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated Learning in Mobile Edge Networks: A Comprehensive Survey," IEEE Commun. Surv. Tutor, vol. 22, pp. 2031–2063, 2020.
[51] C. Baur, S. Albarqouni, and N. Navab, "Semi-supervised Deep Learning for Fully Convolutional Networks," in International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, 2017, pp. 311–319.
[52] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in Proceedings of the 4th International Conference on Learning Representations, ICLR 2016-Conference Track Proceedings, 2016, pp. 2–4.
[53] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep Reinforcement Learning: A Brief Survey," IEEE Signal Process. Mag, vol. 34, pp. 26–38, 2017.
[54] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in Proceedings of the 2017 International Conference on Engineering and Technology (ICET), 2017, pp. 1–6.
[55] L. Mou, P. Ghamisi, and X. X. Zhu, "Deep Recurrent Neural Networks for Hyperspectral Image Classification," IEEE Trans. Geosci. Remote Sens, vol. 55, pp. 3639–3655, 2017.
[56] M. W. Gardner and Dorling, "SR Artificial neural networks (the multilayer perceptron)-A review of applications in the atmospheric sciences," Atmos. Environ, vol. 32, pp. 2627–2636, 1998.
[57] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani, "Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization," Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics, vol. 2020, pp. 2021–2031.
[58] E. Homepage, pp. 27–27, 2021. [Online]. Available: https://www.ericsson.com/en/reports-and-papers/ericsson-technology-review/articles/critical-iot-connectivity
[59] F. Ang, L. Chen, N. Zhao, Y. Chen, W. Wang, and F. R. Yu, "Robust Federated Learning with Noisy Communication," IEEE Trans. Commun, vol. 68, pp. 3452–3464, 2020.
[60] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning," in Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP), 2019, pp. 739–753.
[61] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," Knowl. Based Syst, vol. 216, 2021.
[62] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y. C. Liang, and D. I. Kim, "Applications of Deep Reinforcement Learning in Communications and Networking: A Survey," IEEE Commun. Surv. Tutor, vol. 21, pp. 3133–3174, 2019.
[63] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge Intelligence: Paving the Last Mile of Artificial Intelligence with Edge Computing," Proc. IEEE 2019, vol. 107, pp. 1738–1762.
[64] A. Al-Ansi, A. Al-Ansi, A. Muthanna, I. Elgendy, and A. Koucheryavy, "Survey on Intelligence Edge Computing in 6G: Characteris- tics, Challenges, Potential Use Cases, and Market Drivers," Future Internet, vol. 13, 2021.
[65] L. Cui, S. Yang, F. Chen, Z. Ming, N. Lu, and J. Qin, "A survey on application of machine learning for Internet of Things," Int. J. Mach. Learn. Cybern, vol. 9, pp. 1399–1417, 2018.
[66] A. Shakarami, M. Ghobaei-Arani, and A. Shahidinejad, "A survey on the computation offloading approaches in mobile edge computing: A machine learning-based perspective," Comput. Netw, vol. 182, 2020.
[67] K. Kumar, J. Liu, Y. H. Lu, and B. Bhargava, "A Survey of Computation Offloading for Mobile Systems," Mob. Netw. Appl, pp. 18–18, 2013.
[68] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile Edge Computing: A Survey," IEEE Internet Things J, vol. 5, pp. 450–465, 2018.
[69] H. Lin, S. Zeadally, Z. Chen, H. Labiod, and L. Wang, "A survey on computation offloading modeling for edge computing," J. Netw. Comput. Appl, vol. 169, 2020.
[70] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, "A Survey on Mobile Edge Networks: Convergence of Computing, Caching and Communications," IEEE Access, vol. 5, pp. 6757–6779, 2017.
[71] J. Yao, T. Han, and N. Ansari, "On mobile edge caching," IEEE Commun. Surv. Tutor, vol. 21, pp. 2525–2553, 2019.
[72] A. Filali, A. Abouaomar, S. Cherkaoui, A. Kobbane, and M. Guizani, "Multi-Access Edge Computing: A Survey," IEEE Access, vol. 8, pp. 197 017– 197 046, 2020.
[73] J. L. Salmeron and I. Arévalo, "A Privacy-Preserving, Distributed and Cooperative FCM-Based Learning Approach for Cancer Research," in Proceedings of the Lecture Notes in Computer Science. Springer, 2020, pp. 477–487.
[74] I. Feki, S. Ammar, Y. Kessentini, and K. Muhammad, "Federated learning for COVID-19 screening from Chest X-ray images," Appl. Soft Comput, vol. 2021, 107330.
[75] B. Yan, J. Wang, J. Cheng, Y. Zhou, Y. Zhang, Y. Yang, L. Liu, H. Zhao, C. Wang, and B. Liu, "Experiments of Federated Learning for COVID-19 Chest X-ray Images," in International Conference on Artificial Intelligence and Security. Springer, 2021, pp. 41–53.
[76] Z. Ning, K. Zhang, X. Wang, L. Guo, X. Hu, J. Huang, and R. Y. Kwok, "Intelligent edge computing in internet of vehicles: A joint computation offloading and caching solution," IEEE Trans. Intell. Transp. Syst, vol. 2020, pp. 2212–2225.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)