Fusion of IoT, AI, Edge–Fog–Cloud, and Blockchain: Challenges, Solutions, and a Case Study in Healthcare and Medicine

Farshad Firouzi, Shiyi Jiang, Krishnendu Chakrabarty, *Fellow, IEEE*, Bahar Farahani, Mahmoud Daneshmand, JaeSeung Song, *Senior Member, IEEE*, and Kunal Mankodiya, *Member, IEEE*

***Abstract*—The digital transformation is characterized by the convergence of technologies—from the Internet of Things (IoT) to edge–fog–cloud computing, artificial intelligence (AI), and Blockchain—in multiple dimensions, blurring the lines between the physical and digital worlds. Although these innovations have evolved independently over time, they are increasingly becoming more intertwined, driving the development of new business models. With more adaptation, embracement, and development, we are witnessing a steady convergence and fusion of these technologies resulting in an unprecedented paradigm shift that is expected to disrupt and reshape the next-generation systems in vertical domains in a way that the capabilities of the technologies are aligned in the best possible way to complement each other. Despite the fact that the convergence of the four technologies can potentially tackle the main shortcomings of the existing systems, its adoption is still in its infancy phase, suffering from several issues, such as the absence of consensus toward any reference models or best practices. This article provides a comprehensive insight into the fusions of these paradigms by discussing a blend of topics addressing all the importation aspects from design to deployment. We will begin this article by providing an in-depth discussion on the main requirements, state-of-the-art reference architectures, applications, and challenges. Following this, we will present a reference architecture and a case study on privacy-preserving stress monitoring and management to better elaborate on the corresponding details and considerations.**

***Index Terms*—Artificial intelligence (AI), big data, blockchain, cloud computing, edge computing (EC), eHealth, federated learning (FL), fog computing (FC), healthcare, Internet of Medical Things (IoMT), Internet of Things (IoT), medicine, privacy preserving.**

Manuscript received 15 November 2021; revised 22 April 2022; accepted 1 July 2022. Date of publication 18 July 2022; date of current version 20 February 2023. The work of JaeSeung Song was supported by the Ministry of Science and ICT (MSIT), South Korea, under the Information Technology Research Center (ITRC) Support Program supervised by the Institute for Information and Communication Technology Promotion (IITP) under Grant IITP-2022-2021-0-01816. *(Corresponding author: Farshad Firouzi.)*

Farshad Firouzi, Shiyi Jiang, and Krishnendu Chakrabarty are with the Electrical and Computer Engineering Department, Duke University, Durham, NC 27708 USA (e-mail: farshad.firouzi@duke.edu).

Bahar Farahani is with the Cyberspace Research Institute, Shahid Beheshti University, Tehran 1983969411, Iran.

Mahmoud Daneshmand is with the Business Intelligence and Analytics, Stevens Institute of Technology, Hoboken, NJ 07030 USA.

JaeSeung Song is with the Computer and Information Security Department, Sejong University, Seoul 15600, South Korea.

Kunal Mankodiya is with the Electrical, Computer, and Biomedical Engineering Department, University of Rhode Island, Kingston, RI 02881 USA.

Digital Object Identifier 10.1109/JIOT.2022.3191881

I. Introduction

THE Internet of Things (IoT), artificial intelligence (AI), distributed ledger technology (DLT), and edge–fog–cloud computing continue to drive digital transformation and the convergence of these technologies paves the way for the development of creative applications, innovative business models, and new opportunities in vertical domains, from healthcare to finance, mobility, energy, supply chain, and industry 4.0 [1]–[4]. These technologies have the ability to enhance business processes and disrupt entire sectors or industries. However, to date, a few concepts and applications have experienced this convergence with positive outcomes. Until recently, the correlation between these technologies has often been overlooked as they were mostly used independently of one another. Nevertheless, these technologies are complementary and when fused together can shift the paradigm (See Fig. 1). The convergence of the technologies creates synergy with the potential to transform how industries, businesses, and economies work. Combining AI, IoT, edge–fog–cloud, and DLTs empowers companies and organizations to leverage the benefits of each technology while simultaneously mitigating the limitations or risks each presents.

The IoT is a network of things and a chain of gadgets in the environment that are interconnected in order to generate uninterrupted communication and services [3]. The IoT includes an unprecedented number of connections between things as well as between things and people. With more than 35 billion connected devices in 2021 [5], the IoT is one of the primary contributors to the increasing volumes of big data [6]. The success of the current and future IoT landscape demands service provision characterized by scalability, ubiquity, reliability, and high performance, among others. Cloud IoT (i.e., a new paradigm where Cloud and IoT have fused) has emerged as an enabler to fulfill these attributes by providing smart services specific to aggregating, storing, and processing data generated by IoT [7]. While the fusion of Cloud computing and IoT brings opportunities, Cloud IoT suffers from shortcomings, e.g., latency, connectivity, and bandwidth that cannot to handle the time and context-sensitive needs of the a large variety of IoT applications. Thereby, computing models are now moving away from centralized models to distributed computing paradigms, such as Mobile-edge computing (EC),

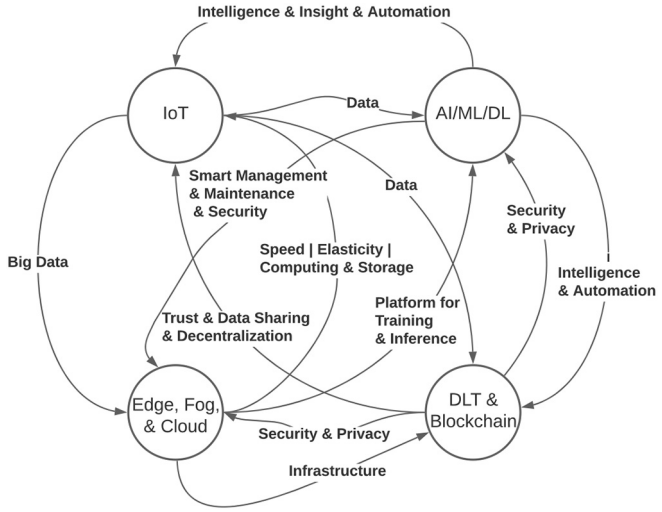


Fig. 1. Correlation among IoT, AI, DLT, and edge–fog–cloud. In nutshell, IoT generates data for AI/ML, enabling holistic models that can add an intelligent layer to the IoT. IoT also provides data to DLT/Blockchain in exchange for trust, transparency, and decentralization. Similarly, IoT is a new source of data for Edge/Fog/Cloud and in return, it benefits from low-latency hyperscale resources.

Cloudlet, fog computing (FC), and transparent computing [1]. In this context, EC and FC add new dimensions to the Cloud IoT by bridge the gap between the cloud and endpoint IoT devices through the deployment of computing power closer to the edge near the source of data [8]–[14]. It should be noted that the characteristics of EC/FC suggest it was not created as a cloud computing replacement, but it is considered a as complementary solution to cloud computing by expanding the cloud to the edge of networks. This technology enables the realization of distributed applications addressing the growing demand to tackle latency challenges of real-time applications, processing the increasing amount of IoT data on edge, and improving resilience to network connection issues [1], [15], [16]. Combining the capabilities of EC/FC with cloud computing results in a hierarchical, heterogeneous, and dynamic paradigm, known as edge–fog–cloud, enhancing resource usage and Quality of Experience of IoT services thanks to its flexibility in terms of latency versus computation [1], [17]–[19].

AI and IoT have also been blending, and several businesses have already adopted AI in their IoT applications [20]. While IoT deals with data collection and device management, AI simulates smart behavior and enables automation, intelligence, reasoning, planning, and perception. Augmented intelligence has also emerged as a new section of AI by combining the complementary strengths and problem-solving capabilities of humans and AI leading to human–computer symbiosis in IoT. In general, the fusion of AI and IoT, known as the Artificial/Augmented Intelligence of Things (AIoT), empowers data management and analytics, enhances human-machine interactions, and improves the efficiency of IoT. The IoT's highest potential can only be achieved through combination with AI—including, machine learning (ML), deep learning (DL), augmented intelligence, and big data analytics—that continues to remove the barriers and obstacles found in conventional IoT models [20]–[26]. The following are some of the most common benefits of merging these two disruptive streams [15], [27]–[30].

1) The precise value of IoT is specified by its analysis step and AI can wring insights from the generated data unlocking IoT potentials. AI enables the processing and analyzing large amounts of multiscale, multimodal, and heterogeneous data generated in IoT devices, networks, and infrastructures.
2) In addition, because IoT networks/infrastructures are made up of a multitude of devices/servers, there are many points of vulnerability, making them open to data theft, hacking, and fraud. To safeguard security/privacy, AI can fend off hackers or attacks. Network, infrastructures, and data security can also be protected by AI because it prevents unauthorized access and data modification.
3) AI also improves IoT functionality by making it more autonomous and smarter (e.g., AI-enabled task offloading in edge–fog–cloud IoT), forecasting operation circumstances, enhancing Operational efficiency (e.g., AI-enabled thermal management of IoT data centers), evaluating maintenance and minimizing unplanned downtime (e.g., predictive maintenance of IoT infrastructures), enabling real-time monitoring and traffic management, and increasing scalability (e.g., compressing and managing vast amounts of data generated by the IoT ecosystem).

DLT, of which blockchain is a popular example, provides a robust and secure decentralization, automation of contracts as well as transparency, accountability, integrity, scalability, cost-efficiency, security, and privacy [1], [31]–[33]. The convergence of blockchain and IoT technologies is an unprecedented paradigm shift that is expected to be a game changer driving the next wave of digital transformation. DLT has exactly what is needed to tackle the shortcomings, vulnerabilities, and weaknesses of IoT. Security challenges have hindered many large-scale deployments of IoT. Another issue with the current IoT solutions is rooted in centralized systems and their scalability. DLTs have the potential to help alleviate some of the security and scalability concerns associated with IoT [2], [32]–[35]. For instance, DLT removes the need for trust among stakeholders by providing a tamper-resistant record of shared transactions. The shortcomings of client/server systems can be tackled by the peer-to-peer nature of DLTs. It brings a revolutionary level of transparency in a way that has not existed in traditional IoT systems [32]–[36]. In addition, DLT can accelerate the adoption of the data/device/service marketplace [23].

These four transformational technologies represent an unprecedented opportunity in many areas on their own but they are exponentially more powerful when combined to synergize their competencies. Before these converging systems get widely adopted, the corresponding challenges and design/deployment implications must be identified and addressed. This article explores the properties, value propositions, and practical implications of these distinct but

increasingly converging technologies. This article is also geared toward providing interdisciplinary insight into opportunities, challenges, issues, and reference models of the convergence of these technologies.

The remainder of this article is organized as follows. Section II demonstrates data monetization strategies, techniques, and architectures, including data sharing and data/AI marketplaces. Section III presents the fundamentals of the Convergence of IoT and edge–fog–cloud. In Section IV, we discuss how AI could benefit the IoT and what challenges and barriers need to be addressed to grow further. This section also describes the main techniques and models for privacy-preserving ML (PPML) and edge analytics. Section V presents the integration of DLT and IoT. Section VI presents a conceptual reference architecture for the convergence of these technologies in healthcare with the help of a case study. Finally, Section VII concludes this article.

II. IoT: Where Do the Data Go

As the IoT rapidly evolves and the public embracement of wearable devices increases, many companies are inundated with huge amounts of data. Discovering how to profit from all of that data can help companies thrive in the market because data can add significant value to various parts of the business. Data monetization has already begun across several verticals as connected devices are layered with AI/ML-enabled smart services and Software-as-a-Services (SaaS) options like additional smart insights or subscription services. These developments create space for a "machine economy" in which accurately monetizing data will produce a substantial advantage in the digital environment. Recent advances in the areas of AI, analytics, and Big Data have created new opportunities for competition because IoT data can be used strategically to open new revenue streams. This growth has created space for new business models, tools, marketplaces, architectures, and platforms that companies use to monetize data. At this point, new business models are working to change the balance of power between companies that gather data, and the users [23]. Creating revenue using IoT data is typically possible through indirect or direct data monetization [23], [37], [38].

1) *Direct Data Monetization:* There are buyers for raw data. While several methods of selling data are available, information is largely sold in data marketplaces.
2) *Indirect Data Monetization:* Data can be helpful when it comes to enhancing business operations, generating new services or products, and developing innovative business models. Optimizing IoT data typically requires the use of [23] and [37].
    a) *Data-Driven Optimization:* Data are utilized to lower costs and improve process efficiency using Big Data analytics and AI or ML. However, it is worth noting that AI/ML and analytics can be handled by one participant or by multiple parties in a collaborative environment by employing multiparty and privacy-saving computational techniques. This method of data optimization can be used by multiple industries. For example, it can shorten manufacturing test periods, or data could be utilized to influence the design of a product [23], [37].
    b) *Data-Driven Business Models:* Data could be used to facilitate business development or customer acquisition through the generation of new services or products. Such a business model enables data owners to create entirely new data-driven businesses instead of only adjacent businesses. These business models are also elemental to supporting various revenue streams. For example, Bosch utilizes manufacturing data to create personalized subscription services that take care of hydraulic monitoring for customers [23], [37].

Typically, there are two means of data exchange to turn IoT into a data economy.

1) *Centralized Solutions:* Solutions, such as online stores, collaborative environments, data markets, or shared repositories are often created using a cloud platform that facilitates the sharing, storing, and selling of data through an access management system. The main concern when a single organization (i.e., market owner) gathers massive amounts of data from various data providers or producers is that data maintained on a centralized server can be vulnerable to hacking, leaks, or other security breaches. When critical data are held by the server owner and all data passes through that single point, data providers cannot control how their data are utilized once the data has been released to the platform owner. Such issues are foundational to centralized solutions, making them less enticing for primary stakeholders. In addition, the opportunity for exposure, competitiveness, or a lack of trust often keeps stakeholders from storing private data via a centralized solution with the potential to reveal proprietary information or business strategy [23].
2) *Decentralized Solutions:* These solutions have been offered as a means of handing power back to data providers by enabling them to control the information they generate or share. Decentralized solutions allow data producers and consumers to interact, communicate and exchange, or share data in a peer-to-peer environment without the need for centralized administration. Such a method is fault tolerant, so there is no single point of system failure. Typically, decentralized markets, such as Enigma Data, Ocean Protocol, Streamer, or IOTA are used via a secure smart contract, DLT, and blockchain that allows data producers to directly complete transactions with data consumers while completely controlling data [23].
    a) *Blockchain Based:* Storing data on the blockchain can generally be done using two techniques. On-Chain Solutions—Data producers store raw, encrypted data directly on the blockchain. However, such a method is not typically scalable because of the throughput and transaction processing ability of blockchain. Off-Chain Solutions—Raw data are stored off chain, and reference points for the raw data are stored on the

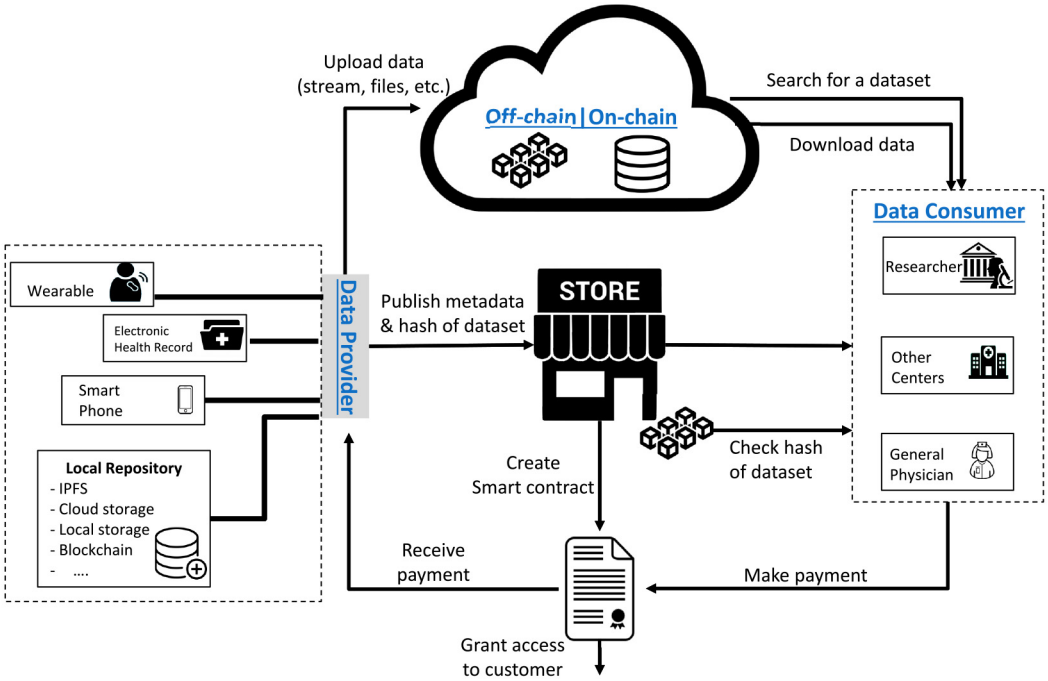


Fig. 2. Blockchain-based data marketplace, data sharing, and data monetization.

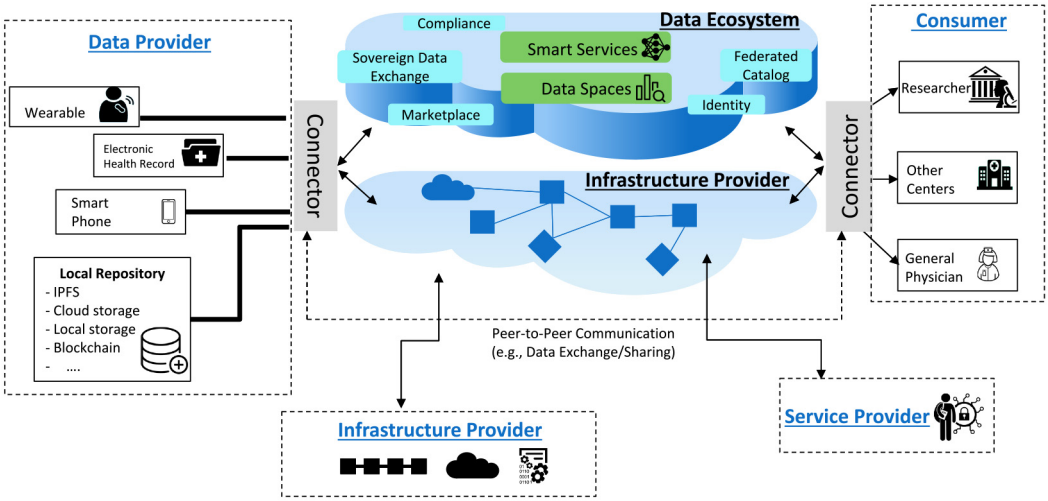


Fig. 3. Federated data ecosystem platform.

blockchain. The raw data are not maintained on the chain because it is more scalable and secure to keep and exchange metadata that refers to raw data. Data producers can maintain ownership of the data and decide if they want to share data by exchanging reference points. An off-chain solution also stops any party that intercepts a reference point from accessing data without authorization [39]–[45]. Fig. 2 demonstrates a conceptual blockchain-based data sharing solution.

b) *Federated Systems:* Fig. 3 illustrates a conceptual decentralized architecture, such as GAIA-X [46] and international data space (IDS) [47], providing a sovereign data sharing via a peer-to-peer connection between consumers and producers. Federated Systems not only can be applied to monetize data but also can be applied to infrastructures and smart services. As shown in this figure, the solution consists of the following components/roles [23].

i) *Data Producers/Suppliers/Vendor/Provider:* A producer is an entity, organization, or business that generates or legally owns/controls the data. On the other hand, providers are mediators that collect the generated data from different producers and offer those data assets to the ecosystem on behalf of the producers. Note that Producers can also act as providers themselves.

ii) *Service Provider:* Service providers offer software and ML models to the ecosystem.

iii) *Consumer:* A Consumer is an entity that consumes some resources/assets (e.g., data, services) in accordance with the usage guidelines and policies provided by Producers/Providers.
iv) *Infrastructure Provider:* An infrastructure provider, known as a node, is a participant that provides computing resources to the ecosystem.
v) *Broker:* A broker facilitates the registration and discovery of available resources (e.g., infrastructures, services, and data sets) via managing a set of metadata and self-descriptions. Note that this role is not ot exclusive and, thus, a number of brokers can exist simultaneously in the system.
vi) *Identity Provider:* Identity Providers create and manage the identity information of the participants.
vii) *Federator:* A federator enables and facilitates the interaction between providers and consumers.

III. Convergence of IoT and Edge–Fog–Cloud

Traditionally, the widespread popularity of cloud computing resulted in reduced local edge storage demand and enabled edge devices to offload computing tasks. However, more and more data are produced at the network's edge; therefore, it is most adequate to analyze data at the edge of the network. Several technologies, such as Cloudlet, have been introduced because as cloud computing might not lead to the best data processing performance when the data are generated near the network's edge, particularly for delay-sensitive and bandwidth-hungry applications. The rapid expansion of the IoT and the rapid embracement of cloud services have pushed the limits of edge/FC, that requires processing data at the edge. EC/FC has the capability to resolve challenges around latency, data privacy and security, battery life, and response time while saving bandwidth costs. The main reasons behind the integration of edge and FC into IoT applications are listed below [1], [48]–[50].

1) *Big Data Meets Edge:* Virtually, all types of devices will eventually become part of the IoT and produce or consume data. Therefore, many IoT applications are facing the challenge of incremental volumes of data generated on the edge of the network. Although centralized cloud computing as the standard route for computational data could be the answer to some data processing demands, it cannot accommodate the tsunami of data under the presence of billions of devices leading to latency, congestion, and connectivity-related issues. Moreover, most of that IoT data will be consumed at the network's edge. However, the current IoT-cloud paradigm is not sufficient leading to unnecessary computing and bandwidth resource use. Additionally, the privacy needed will create a challenge for cloud computing. Finally, most IoT devices are generally energy constrained (powered with batteries) and, thus, offloading a task to the cloud may not be very efficient as wireless communication is resource hungry [48]
2) *Push From Cloud Services:* Moving computing and data processing to the cloud has emerged as an effective means of data processing because cloud computing power outpaces edge device capability. However, in comparison to the quickly evolving data processing techniques, network bandwidth has ground to a halt [48]. As the amount of data produced at the edge grows, data transmission speed has created a bottleneck for cloud-based computing models. If all data must be transmitted to the cloud, the response time becomes too lengthy. In such cases, data should be processed at the network's edge to shorten the response time and reduce network pressure [48].

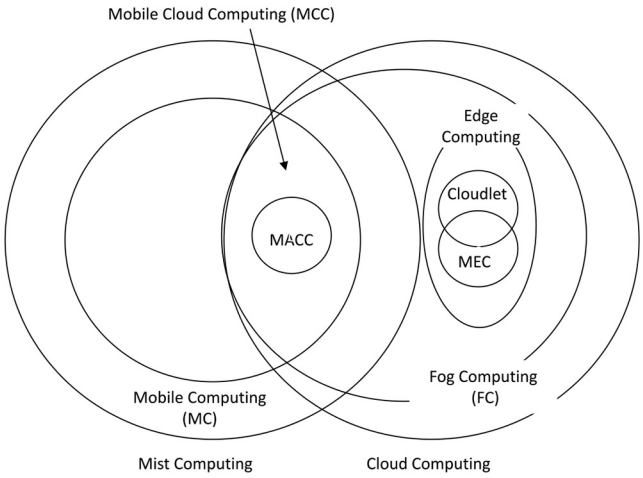


Fig. 4. Comparison of EC/FG and the other related computing paradigms [1].

Besides EC/FG, there are a number of other computing models that can be exploited in IoT applications to address the shortcomings of the existing Cloud-based IoT models. The frequently debated and most promising technologies are summarized in Fig. 4 [1]. In a nutshell, EC pushes the intelligence and processing capabilities down closer to the endpoint devices. In EC, processing resources are placed one or a few hops away from devices. FC is a mediator between the edge and the cloud. Compared to EC, FC has a broader concept providing a hierarchical model consisting of computing, storage, networking, and control anywhere across the edge devices and the cloud. Mobile computing (MC) refers to wireless technology enabling mobile hardware (e.g., smartphones, laptops, portable PCs, and tablet PCs) to transport data over a network without getting any physical link. Mist Computing harvests the extreme edge of the network, typically consisting of microcontrollers and sensors (i.e., computations, local analytics, and decision-making are provisioned and conducted directly on the IoT device itself). Mobile cloud computing (MCC) is the fusion of cloud and mobile intended to tackle mobile devices' processing/storage limitations. MCC is designed based on the idea of remote execution of offloaded tasks outside of the mobile device. However, MCC is not

geographically distributed and cannot provide ultralow latency services. Multiaccess EC (MEC), formerly known as mobile EC, provides capabilities and features of the cloud computing at the edge of the cellular network via existing base stations. Typically, MEC is considered as a subset of EC because EC is a more general technology covering all types of connectivity (e.g., Cellular, WiFi, and LAN). MEC is also a different technology from MCC in a way that the operator of MEC is a network infrastructure provider while the operator of MEC can be either a user or a cloud provider. Moreover, the computing resource of MEC is lower than MEC. Additionally, MCC latency is higher than MEC. The concept of Cloudlet is very similar to MEC. Cloudlets consist of virtual machine infrastructures (small-scale local clouds) present in logical vicinity to the IoT devices addressing data interactivity in real time. However, this paradigm is different from FC in a such a way that FC is heterogeneous (e.g., routers, switches, access points, and gateways) and provided over one or multiple hops (placed anywhere between cloud and IoT devices), whereas Cloudlets are small servers provided over one-hop access. Finally, mobile ad hoc cloud computing (MACC) is a subset of MCC that aims to provide more efficient services by forming a temporary group of mobile devices in the vicinity to create an ad hoc cloud [1].

While neither EC/FC nor Cloud computing is perfect, the insightful integration of edge–fog–cloud allows IoT to benefit from the combined competencies and attributes [1].

1) *Collaborative and Distributed Computing:* AI unleashes IoT potential by drawing insights from data faster. Because of the velocity, variety, volume, and veracity of IoT data, traditional cloud-based models are unable to meet the needs of emerging IoT applications, making it vital that intelligence be distributed across all IoT layers to shorten response time and lower data transmission.
2) *Reduced Network Load:* Managing data streams at the network's edge by locating computing nearer to data sources more evenly distributes the workload and lessens the backhaul burden. Therefore, it is not necessary to increase the rate of backhaul data, and data linked to the far side of the network are saved for the services that need them most.
3) *Scalability:* Scalability is the main need when IoT is spread across a large geographic area with various levels of density. This requires that IoT depend on various distributed configurations and topologies to adapt to each network's needs.
4) *Latency-Aware Computing:* Spreading resources across the network reduces the network load and improves task processing time. FC significantly increases the Quality of Service (QoS), enabling latency-critical applications to offload tasks without infringing on latency restraints.
5) *Native Support for Mobility:* Moving resources nearer to the edge enables the network to react to user mobility more accurately and quickly. Because fog notes can communicate horizontally, request handling and task offloading can happen in higher mobility situations.
6) *Providing Context:* Because resources are located nearer to end users, it is possible to provide content specific to particular geographical areas. Fog nodes (FNs) conscious of their absolute position can also provide extra built-in services such as precise localization.
7) *No Single Point of Failure:* Moving resources throughout a network also means that if a specific data link is disabled due to maintenance or an FN goes down due to a cyberattack, other FNs can handle the workload while the network continues functioning.
8) *Extended Battery Life:* Because FC reduces data processing delays, devices at the edge can depend on task offloading to reduce energy needs and expand battery life, which provides greater long-term autonomy.
9) *Reduced Energy Use:* Because the majority of data is processed near the source, long-distance data transfers are not required, which lessens the network's overall energy consumption. In addition, because power loads are more equitably dispersed, power demands can more easily be met by existing infrastructure as well as renewable energy sources.
10) *Heavy Load Support:* Edge–fog–cloud computing models can depend on the cloud to have substantially more resources than any FN. This means an overwhelmed FN can send a heavy computational task to the cloud for completion; however, the network propagation time is longer in such scenarios.
11) *Limitless Storage:* Space-constrained devices can rely on the cloud's infinite storage and resource capabilities to avoid the need to integrate additional storage capacities.
12) *Location Awareness:* This makes IoT applications stronger by supporting improved adaptability and consciousness while enhancing environment-sensing knowledge that allows the network to improve the quality of applications as well as task execution.
13) *Mobility:* As the IoT adapts to manage high mobility devices, system knowledge has to transfer openly within data-heavy mobile applications. Finding the right data enhances performance, data models, and local caching. IoT solutions must also allow mobile devices to transition between authority regions without creating a system disruption.

The state-of-the-art architectures of EC/FC are illustrated in Fig. 5 [1], [3], [51]–[54]. The three-layer IoT architecture is undoubtedly the most widely used one. This model consists of only one layer between the edge and cloud, serving as the fog layer. In some applications, FNs are laterally/vertically connected to each other, forming a grid to address concerns associated with reliability, load balancing, communication overhead, and data sharing. In this model, generally, data/tasks are shared vertically, whereas, typically, there is no communication among peers inside the same horizontal layer. Usually, FNs closer to the edge act as data collectors as well as actuators/sensors controllers. On the other hand, FNs in the above layers close to the cloud are concerned with data filtering, near real-time processing/analytics, compressing, and transforming. Moving further from the edge toward the cloud typically provides more sophisticated ML and data processing capabilities at the cost of higher latency and communication overhead. Some architectures proposed to structure the fog in

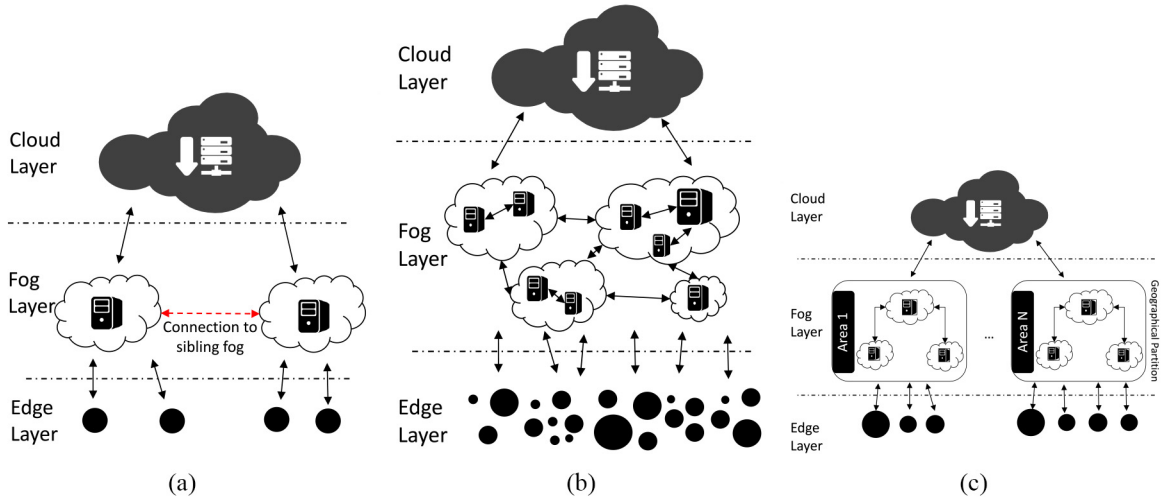


Fig. 5. Examples of edge–fog–cloud architectures. (a) 3-Layer model. (b) Clustered model. (c) Tree-based model.

the shape of a tree with nodes rooted in the cloud. Finally, FNs can also be clustered based on geographical constraints while facilitating inter and intrafog communication ability [1].

As shown in Fig. 6, a typical FN consists of the following main components [1], [8], [11], [12], [50], [55], [56].

1) *Fog Gateway Nodes (FGN):* FGNs facilitate the connection between endpoint IoT devices and the fog. In other words, FGNs are gateways, access points, or entry points.
2) *Fog Orchestration Controller (FOC):* These components create a control layer in fog, creating a holistic and ubiquitous picture of the whole fog. They continuously monitor all the existing fog resources and provide a set of resource performance metrics. Additionally, they are responsible for coordinating communications within a fog or among fogs. Task offloading, task migration, scheduling, service placement, and resource provisioning and management are also orchestrated by FOCs. Typically, these tasks need to consider communication cost, computation/energy efficiency, and latency. The existing solutions can be classified into the following categories.
   a) *Offline Versus Online:* Task placement and scheduling using offline/static techniques are typically conducted using optimization tools and solvers, such as CPLEX, Choco, and Gurobi. Although these solvers provide acceptable results, they are very time consuming and cannot capture the dynamic input patterns of the incoming workloads. On the other hand, online techniques can better react to online situations based on the information about the tasks arrived rather than assuming prior knowledge about arrival times, execution times, and deadlines.
   b) *Centralized Versus Decentralized:* Centralized techniques utilize a central controller to optimize the system. Such controllers need to have a comprehensive overview of the whole system to be able to fulfill their tasks.

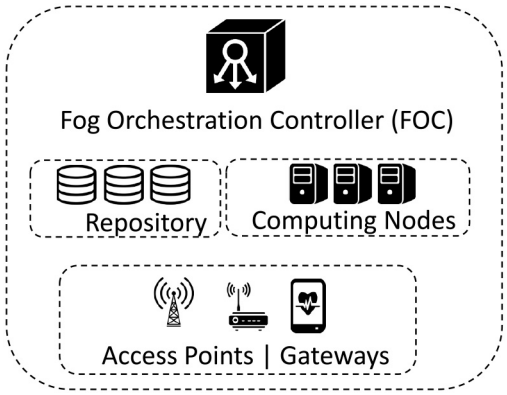


Fig. 6. Conceptual architecture of an FN.

3) *Fog Computing Nodes (FCNs):* These entities are formed by at least one or more physical devices with processing and sensing capabilities enabling the fog to run and execute a task assigned by FOCs.
4) *Fog Storage Nodes (FSNs):* FSNs implement a distributed database/repository inside a fog.

IV. Convergence of IoT and AI

Integrating AI into the IoT systems across the entire stack—from infrastructures to applications—effectively creates a layer of intelligence. There are a number of ways to implement this new paradigm, AIoT, but generally, the role of AI in the era of IoT can be classified into the following ways.

1) *Role of AI in the Things (IoT Edge):* The essence of IoT is about collecting data from connected devices. AI in the sensing layer provides innovative breakthroughs when it comes to handling tsunami of data. The leverage of AI in IoT things brings actionable intelligence and insights as close as possible to the source of data minimizing the decision-to-action latency. In addition, balancing resource constraints of IoT devices and the astonishing growth in data volume, variety, velocity, and veracity (known as the 4 V's of Big Data)

requires selective sensing; otherwise, ubiquitous sensing will be a challenging task as it needs an extensive pool of resources [57]. This means honing in on the most appealing data by carefully selecting only those data/information of interest from a noisy environment with limited resources [21], [57]

2) *Role of AI in the IoT Infrastructures:* Another layer of complication with IoT is the infrastructures. Indeed, there are many aspects and subcomponents to IoT infrastructures that AI is integrated as analyzing the deluge of constraints (e.g., energy consumption, latency, and QoS) and data generated by IoT infrastructures that otherwise was simply impossible for humans to review and extract insights. The following are some of the most common benefits of incorporating AI into the IoT network/infrastructures layer: security (e.g., tracking down a wide variety of cyber threats from malware to phishing attacks, breach risk prediction, and fog-assisted endpoint protection, among others), optimization of connectivity/network, infrastructure and service monitoring, infrastructure maintenance, load balancing, traffic management, capacity, and resource planning (i.e., identify the optimal configuration of resources by considering the location-based user requests as well as the performance of the existing hardware resources) [1], [58]–[60], resource provisioning (i.e., providing resources, such as memory and computing resources), resource allocation (i.e., assignment of resources to address an incoming request), offline/online centralized/hierarchical/distributed service orchestration, service migration, and task offloading [61]–[66], [66]–[70].

3) *Role of AI in the Application/Service Layer:* The sheer quantity of IoT data is significant. AIoT surfaces relevant insights, enabling the next level of automation and productivity, rapidly transforming everything from the supply chain to healthcare. AI/ML algorithms can assist in making decisions on behalf of users in IoT in a large variety of contexts. AI algorithms improve model robustness and generalizability compared to non-AI algorithms. They can also be employed to detect security issues at the data, software, and system level.

A. Power of Combining AI With the IoT Edge

The explosive growth of smart devices and sensors used everywhere, from factories to communities, produces vast amounts of data. Constantly growing computing power pushes computation from the cloud to the network's edge. From smart manufacturing to facial recognition and smart cities, AI-driven applications/services are flourishing. However, because of latency and bandwidth challenges, the traditional cloud computing model is unable to support the full potential of AI across all organizations and use cases. Therefore, deploying AI/ML/DL services and edge resources nearer to the data's origin presents viable solutions. The interaction and convergence of IoT edge and AI can be summarized as follows [71].

1) *AI For Edge:* AI can be utilized to optimize EC resources, networks, and systems, from edge caching to resource management and maintenance, task scheduling, offloading, communication, security, etc. [71].
   a) *Offloading:* Depending on the application and runtime requirements, offloading of IoT tasks is required to transfer heavy computation and storage to fog/cloud resources and, thus, handle the latency and resource gaps. In this context, several works have been conducted to investigate and address various aspects of offloading, such as application characteristics, latency challenges, resource heterogeneity, uncertainty and the dynamics of the environment, mobility, energy consumption, load balancing, resource utilization, computational time, communication overhead, among others. As a consequence, there are many ongoing research projects focusing on offloading from heuristic methods to the Markov decision process, fuzzy logic system, and reinforcement learning, among others [72], [73].
   b) *Edge Caching:* It has been shown that idle storage resources can be utilized to tackle a series of challenges (e.g., unnecessary end-to-end (E2E) communication) raised by limited spectrum and bandwidth resources of edge [74]. This promising approach, known as Edge caching, is mainly constructed based on AI technology to address the corresponding questions: where to cache, and when and how to cache [74].
   c) *Edge Management and Maintenance:* To deal with the edge management and maintenance problem, a large number of AI techniques have been proposed. For instance, AI techniques have been successfully applied to the handover problem of the edge devices as well as control communication modes for energy minimization.
   d) *Security/Privacy:* The field of IoT security/privacy is incredibly vast in its size and scope as IoT comprises many devices and heterogeneous network systems in a multivendor environment with their own unique features and vulnerabilities. Typically, edge systems are prone to many attacks, such as jamming attacks, DDoS attacks, side-channel attacks, malware injection attacks, authentication, and authorization attacks, mainly due to their resource-constrained hardware and software heterogeneities [75], [76]. Incorporating AI brings significant advantages [77]–[82]: 1) AI Inference on the edge can significantly eliminate the need for data transmission between edge and cloud, resulting in higher data locality. Such AI-enabled data locality assists with safeguarding user privacy when a gadget catches privacy-sensitive data and 2) AI-enabled security/privacy solutions can also enhance traditional cybersecurity, especially among IoT devices, from scanning devices to discovering vulnerabilities (e.g.,

weak passwords identification), pattern recognition, threat prediction, and protection. It should be noted that although the adoption of AI brings new waves of innovation in cybersecurity solutions, security is also needed to be implemented throughout AI environments. Security needs to be incorporated starting from the training phase to the inference phase as AI also suffers from several vulnerabilities, such as evasion attacks (influencing the model to provide incorrect predictions), data poisoning, and privacy attacks (e.g., membership inference attacks, and model inversion attacks).

2) *AI Applications on Edge:* There is an urgent need to push the AI frontiers, including model training and inference, from the network core to the IoT edge to handle the tremendous amount of data generated by billions of mobile and IoT devices [71], [83].
   a) *Inference:* Cloud-only inference cannot satisfy the requirement of the emerging IoT applications, and it is significantly hindered by the data safety/privacy challenges as well as network latency leading to unacceptable real-time performance and quality of experience, limiting the ubiquitous deployment of ML/DL services. Therefore, there is a necessity of striking a balance between the inference accuracy and the execution latency while considering the resource-constrained environments of edge [71].
      i) *Model Optimization:* ML/DL models should be tailored and optimized to be able to deploy them to edge devices. State-of-the-art optimization techniques can be classified into: weights quantization, parameter pruning and sharing, low-rank factorization, knowledge distillation, and transferred filters [71].
      ii) *Model Segmentation:* In this technique, the model is divided into several partitions and distributed across heterogeneous local resources (e.g., GPUs and CPUs) of the edge device or across edge–fog–cloud [71].
      iii) *Early Exit:* To strike a balance between accuracy, processing delay, and energy consumption, the concept of early exit has evolved. In this approach, the original baseline ML model is augmented with additional side models, each with different performance and overhead/cost. The main idea is that a simple model/side may often be sufficient for a majority of the inputs (data points). On the other hand, more complex models (e.g., additional DL layers) might be required to handle more complex samples. Early Exit techniques switch between different models to co-optimize the performance and costs [84].
   b) *Training:* Typically, due to the limited capacity of resource- and energy-constrained IoT devices, the training phase takes place in the cloud with data collected from IoT devices, and then the extracted model is deployed to the IoT edge. However, this approach might not be appropriate for all use cases. Distributed training techniques, as well as hardware accelerators based on spiking neural networks (SNNs), are some of the promising techniques that emerged as a response to the challenge of training at the edge.

B. Privacy-Preserving Machine Learning in IoT

IoT data across vertical domains are severely fragmented across silos for a range of inhibitor reasons—e.g., data sovereignty, trust, compliance, and legal concerns—halting discovery and technology-driven innovation. For instance, the healthcare sector is currently in the midst of a paradigm transition. Digital advancements are producing massive amounts of data inside healthcare organizations. The rapid expansion of health data is due to the use of EHR, wearable IoT devices, diagnostic lab results, social media feeds, external patient information, and medical imaging. Comprehending that all of this data is an asset that produces business opportunities for healthcare systems and patients to take advantage of monetizing data. However, there are multiple barriers that can block the monetization of IoT data in the healthcare industry. For example, hospitals are understandably uncertain about sharing patient data, and California's Consumer Privacy Act (CCPA), HIPPA, and GDPR regulations hamper the ability to integrate private data. While HIPPA was the inaugural national standard for safeguarding personal health data in the U.S., Europe's GDPR also protects personal data by emphasizing that the collection of user data must be clearly communicated to users. After the GDPR was put into place, CCPA as well as the China Internet Security Law, in conjunction with others, provided additional data security. This means that data from a variety of sources require that those analyzing it do so in compliance with the requirements of multiple regulations, which can complicate medical research. Additionally, data continuously changes, and information must be merged repeatedly to research successfully, which is significantly harder to do than combining data only one time. The fact that smart data are heterogeneous, contain multiple dimensions, are often complicated, and can come from fractured sources creates additional obstacles to conducting research between multiple organizations [23].

Valuable, even vital information often remains unshared and unharnessed as the rise and adoption of AI parallel that of the associated concerns centered around privacy. This adversely impacts the collaborative use of AI/ML/DL not only among institutes but also internally among departments of the same organization. To alleviate this challenge, Collaborative and PPML techniques—e.g., federated learning (FL), homomorphic encryption (HE), Secure multiparty computation (MPC), and differential privacy (DP)—have emerged to bridge isolated IoT data islands as well as the gap between reaping the benefits of AI/ML and privacy concerns (See Fig. 7).

1) Federated Learning: In contrast to traditional centralized ML methods that require all local IoT data to be uploaded to a single server, in FL, data and training remain on local

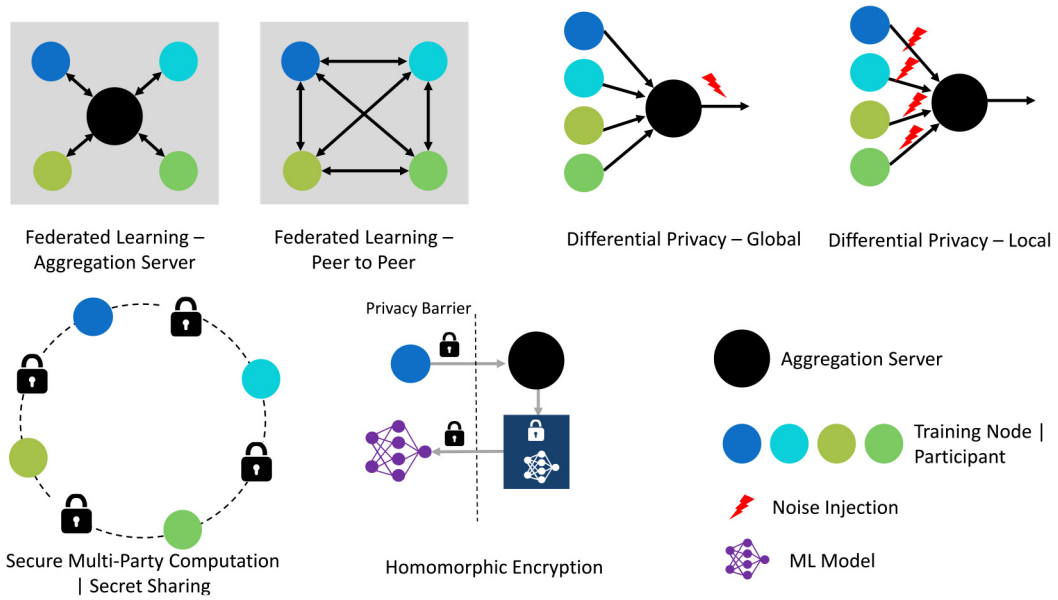


Fig. 7. Comparison of PPML techniques.

nodes enabling many participants to build a robust common ML model without exchanging data. FL seeks to overcome issues around data privacy and governance through collaborative algorithm training that does not require the exchange of data. FL was originally created for use in the domains of edge device use cases and mobile devices but has recently gained popularity in the realm of healthcare applications. FL enables ML from distributed data. Each data controller (participant or party) defines the governance of its own processes as well as associated privacy policies. Each data controller also oversees data access throughout training and validation. With this goal in mind, a typical FL utilizes the following strategy to train a model [23], [85], [86].

1) A pretrained, centralized ML model (global model) and parameters are initiated.
2) A subgroup of training participants is chosen for each round and receives the most current global model copy (e.g., from a server).
3) Each participating client trains and updates a local model with their privacy-sensitive data.
4) Locally trained models are returned to the centralized server and the server in which all local models are aggregated to create/update the global model. The most common aggregation technique is federated averaging (FedAvg). FedAvg is based on parameter averaging to aggregate local models. Once the global model is updated, it is shared with participants for the next iteration. This process is repeated until convergence is achieved.

Generally, there are three methods for exchanging parameters [87].

1) *Round Robin:* FL is run sequentially by participants. Each participant downloads a portion of the current parameters from the server, trains locally, and then uploads chosen parameters (e.g., gradients). Then, the next participant does the same, following a set order [21].
2) *Random:* Participants randomly download, learn, and upload the model.
3) *Asynchronous:* Participants do not coordinate with one another. While a participant is training, other participants can update the server.

FL can be classified based on the network topology into three main categories [23], [88]–[92].

1) *Centralized FL:* A centralized server manages various algorithm steps and coordinates participating nodes throughout the learning process. The central server handles node selection at the start of training as well as model update aggregation. Because all chosen nodes must provide updates to only one entity, the server may become a bottleneck point [88], [91], [93], [94].
2) *Decentralized FL:* No centralized server is utilized, and participants (local nodes or parties) agree on a decentralized protocol to receive and transmit shared parameters until every participant has been included. Nodes can coordinate independently to acquire the global model. Because there is no concept of an actual global ML model, each local node individually upgrades its own local model by exchanging information with neighbors, which averts single point failures because updates are shared only by interconnected nodes [95]. Even so, the network topology can impact learning process performance. Generally, every participant receives trained parameters from other participants. Then, the weight is updated using small batches of local data, and the updated values are sent to others. Typically, only locally trained parameters, rather than data, are shared; therefore, any privacy leaks are indirect and generally occur because of local parameters. Practically using completely decentralized learning

requires the addition of other algorithms or technologies [88], [93], [94], [96].

3) *Heterogeneous FL:* Most current FL strategies assume all participants (local ML models), regardless of their different computation and communication capabilities, use exactly the same global model architecture. However, Heterogeneous FL techniques (e.g., HeteroFL) enable heterogeneous local models while still producing a single global model [97].

FL can also be categorized into Horizontal (HL), Vertical (VL), or Transfer (TL) Learning [23], [89], [95].

1) *Horizontal FL:* Also known as sample-based FL or Homogenous FL, this approach utilizes data sets with identical feature space across all devices.
2) *Vertical FL:* Also known as feature-based FL or Heterogeneous FL, this approach utilizes data sets consisting of different features spaces. In other words, data sets have the same sample space but different features spaces. This approach often includes an intermediary third party to provide encryption logic, ensuring only common data are shared. For example, a healthcare company and social networking company in the same city likely have a high level of user convergence. However, because they capture different kinds of data, their feature spaces are not the same. VL enables participants to gather data from different feature spaces while only one party can see the label. Therefore, neither participant can train the model alone, which adds aggregation complexity. Sample spaces must line up in order for partial model updates to be exchanged. Similarly, a bank's credit card marketing team may want to enhance an ML model by learning about customers' online shopping habits. Shopping data (only for common customers between the bank and online shopping) would be shared to train an ML model, and third-party encryption logic would ensure security and restricted sharing. In the end, the model would enable the bank to refine product offers, and online shopping domains could revise credit card point models [23], [95].
3) *Transfer FL:* This approach is FL used with a pretrained model that was trained using similar data to solve a different problem. Note that transfer learning approaches are employed to train new requirements on a pretrained model used to solve another problem. When it comes to FL, training with a pretrained model provides better results when compared to results from a brand-new model [95].

2) Homomorphic Encryption: HE is a centralized PPML method in which raw data are encrypted and transmitted to a server for computation. HE is specialized encryption that allows a variety of computations to be executed on nonplaintext (encrypted) data; therefore, results are also encrypted. When decrypted, the outputs match plaintext operational results [23], [98]–[102].

1) *Partially Homomorphic Encryption:* These cryptosystems exhibit either a multiplicative or additive homomorphic property but cannot satisfy both properties. Examples include: Paillier (additive), EIGamal (multiplicative), or RSA (multiplicative).
2) *Fully Homomorphic Encryption:* These cryptosystems exhibit both multiplicative and additive properties.

Technical Limitations: Currently, HE is challenging to scale, making it impractical for real-world use. Additionally, execution is very sluggish in comparison to plaintext analysis. Thereby, training an ML model is not feasible, but as research evolves, it may become more practical in the future.

3) Secure Multiparty Computation: Secure MPC (sMPC or MPC) seeks to develop methods for participants to collaboratively compute while maintaining the privacy of inputs. In contrast to conventional cryptography that protects data from outsiders, MPC protects each participant's privacy from the other participants. MPC is centered on the secret sharing of all inputs [23]. Each participant (p_1, p_2, p_N) holds private data (d_1, d_2, d_N). Every participant seeks to calculate the public function value using private data: $F(d_1, d_2, d_N)$, while maintaining the secrecy of their own inputs/data. For example, if three individuals want to determine who has the highest salary without revealing their individual salaries, this would translate into: $F(x, y, z) = \max(x, y, z)$.

There are significant differences between two-party computation (2PC) and MPC techniques. In 1982, the secure computation was introduced as secure 2PC to address the Millionaires' Problem, also known as a Boolean predicate. In 1986, Yao proposed a novel 2PC to address generally feasible computations [103]. Later, Goldreich, Micali, and Wigderson extended the concept of 2PC with further generalizations to MPC [104], [105]. The majority of MPC protocols utilize secret sharing. Secret sharing enables one party to disseminate a secret with other parties by giving a share of the secret to each participant. In other words, when using secret sharing, inputs are broken into pieces, and the pieces are given to the other participants. Computing is carried out locally, and no single participant can access whole inputs. When the local computation outcomes are combined, a whole, accurate output is shared with all participants. Secret sharing generally takes the form of Additive Secret Sharing or Shamir secret sharing (SSS) [98], [106].

1) *Additive Secret Sharing:* The main concept behind this approach can be discussed using a simple example. Suppose there are three colleagues who want to compute their average salary without sharing individual salary data with one another or a third party. If Colleague A's salary is U.S. $110K, then U.S. $110K would be broken into three random secret shares (i.e., U.S. $40K, $50K, $20K). Colleague A would keep one share ($50K) and then give another share (U.S. $20K) to Colleague B and Colleague C (U.S. $40K). All three colleagues would share their salary information using the same process. When all shares have been distributed, each colleague has one share of each participant's secret. Each individual share gives no useful information because it is incomplete. Secret sharing only generates value when the shares are added together. Each participant adds their shares together to compute a partial result.

When the partial outcomes are added together and then divided by the number of participants, the answer is revealed.

2) *Shamir Secret Sharing:* In this approach, a secret is broken into shares. Revealing a secret using SSS requires that a participant have a minimum number of shares, known as the threshold. SSS is based on the foundational concept that two points can create a line, three points can define a parabola, four points can define a cubic curve, etc. Therefore, it requires k points to define a polynomial of degree $k-1$. For example, if Participant A wants to share a secret ($secret = 6$) with Participant B and Participant C, Participant A chooses secret line f, such that $f(0) = 6$. Two other points on that line are chosen. Participants B and C are each given one point. Participant B knows $f(4) = 4$ and Participant C knows $f(8) = 2$. Having only one share of the secret, neither Participant B nor C could reconstruct the secret using only their shares because of the infinite number of points that can define a line.

4) Differential Privacy: DP is a formalized mathematical method used to manage and quantify privacy risks. This framework has mainly been studied within the context of gathering, analyzing, and sharing aggregated statistics, ranging from simple averages to ML. DP is not necessarily a solitary tool but more of a criterion that tools must satisfy in order to analyze sensitive data. DP supplies a mathematically provable guarantee of privacy in the face of privacy attacks designed to learn individual-specific information from data [23]. DP protects privacy through the addition of noise to personally sensitive data. Generally, DP systems are categorized into local privacy, and global privacy [107]. Global systems include a trusted entity known as the curator that can access raw, private data from many different participants [98]. The curator analyzes the data and injects noise into the output. For example, a researcher wants to determine how many hospital patients have an emerging human stigmatization virus (HSV). A hospital administrator can utilize patient records to count the actual number. When applying global privacy, the administrator picks a random number using a probability distribution familiar to both parties, such as the Laplacian mechanism. The noise is injected into the actual number, and the noisy output is provided to the researcher. While the noisy output is likely close to the actual output, it remains impossible to determine if any particular patient has HSV based on the hospital administrator's response.

Local privacy does not include a curator. On the other hand, each individual is responsible for injecting noise into their privacy-sensitive data prior to sharing it with others. In other words, every participant acts as a curator of their own database. Local DP systems could also include an untrusted aggregator to collect data from everyone at once [98]. For example, a political researcher could ask all residents of a town if they have ever joined the Trump's party. In order to safeguard privacy, the researcher would have each resident secretly flip a coin. If the coin shows head, the participant provides a truthful answer. If the coin initially lands on tails, the coin is flipped again to generate their answer, with heads meaning yes and tails meaning no. Using this random response technique, approximately half of the residents will give truthful answers while the remaining provide random answers. This means that every participant maintains plausible deniability around the veracity of their response, and their privacy is protected. In addition, with enough responses, the researcher can accurately estimate how many in the town support Trump.

The DP model is known for providing the highest level of privacy and the lowest chance of individual data identification. DP sets the boundaries for how much information can be shared with an adversary or third party regarding individual data's presence in a data set. These boundaries are usually set by a parameter ϵ called Privacy Budget/Loss. This parameter measures privacy loss in order to quantify what an adversary can learn about a single individual's data through one query. ϵ-DP can be formally defined by the following equation [108]–[110]:

$$P[A(X) \in S] \leq e^{\epsilon} \cdot P\left[A\left(X'\right) \in S\right]. \tag{1}$$

In the above equation, P stands for probability, and parameters X and X' represent two adjacent (neighbor) data sets (databases). Note that adjacent (neighbor) data sets means that they differ in just one element. A is a protocol/function that runs on data set X, producing some output ($A(X)$). S is all potential output of A that could be predicted. Privacy decreases with persistent queries and epsilon's stack up. If a private query where $\epsilon = 1$ is made twice and results in two differing estimates, it is as if a single query was made with a loss of $\epsilon = 2$ because the answers can be averaged to generate a more accurate estimate [98], [110]–[112].

In DP, a noise-adding mechanism serves to protect data through predefined perturbation mechanisms. Generally, the noise addition mechanisms utilized in DP include the following [113]–[115].

1) *Randomized Response:* This technique was initially proposed by Warner and extensively has been used in structured survey interviews enabling respondents to maintain confidentiality while collecting sensitive issues (e.g., sexuality). This technique which acts very similar to plausible deniability removes evasive answer bias through randomized responses. For instance, randomization can occur by flipping two separate, unbiased coins. An answer is considered true if Coin 1 lands on heads. Otherwise, Coin 2 is flipped. If Coin 2 lands on heads, the answer is yes. If Coin 2 lands on tails, the answer is no. If it is assumed that a biased coin is utilized, the probability of receiving a truthful answer is p (generates an untrue answer, with $1 - p$ probability). The randomized response technique provides DP when

$$P = \frac{e^{\epsilon}}{1 + e^{\epsilon}}. \tag{2}$$

2) *Laplace Mechanism (LM):* Noise is calculated utilizing the Laplacian function. Each data coordinate is perturbed via computed Laplacian noise using LM distribution. The level of sensitivity of the DP function decides the noise addition's scale (i.e., $noise \sim Lap(\Delta f/\epsilon)$) [116].

3) *Gaussian Mechanism:* This mechanism has become essential in some DP algorithms. Like the LM, the Gaussian mechanism generates noise via normal, Gaussian distribution.
4) *Exponential Mechanism:* This method generates DP in the case of nonnumber outputs (categorical variables).
5) *Compression Mechanism:* Information-theory-centered information distortion and compression techniques have also been utilized in the literature to satisfy (ϵ, δ)-DP [117].

V. Convergence of IoT and Blockchain

While blockchain and IoT are very different from one another, the integration of the two creates a new paradigm expected to disrupt systems across diverse industries. Blockchain has the ability to address IoT weaknesses by resolving security faults between intelligent IoT devices connecting in a trustless, public environment. In addition, it can help resolve weaknesses in Cloud IoT client/server models because of its distributed peer-to-peer setup (See Fig. 8) [2], [32]–[36]. The IoT has huge potential, but modern IoT systems use devices with finite resources that can be open to cyberattack. In addition, IoT networks are hard to scale, maintenance is challenging, and single point of failure issues remain a problem. Additionally, data privacy and security are primary concerns, and IoT data are impermanent. Blockchain's decentralized way of generating and maintaining data as well as its consensus mechanism, contribute to mitigating centralized IoT concerns. Multiple use cases have illustrated blockchain's applicability to every piece of the IoT system. Blockchain can be utilized in communication networks to verify and encrypt data, securely handle cloud data and as well as that contained in distributed devices, and manage and store device identifications. It is also capable of lessening risk by using multiple nodes for peer-to-peer data sharing, which makes it almost impossible to corrupt data. Blockchain consensus requirements also prevent a compromised node from becoming part of an IoT network or allowing the network to accept altered data [2].

The various layers in a DLT/blockchain stack used with IoT applications include (see Fig. 9) [2], [33], [118].

1) *Device Layer:* This layer includes actuators, smart devices, controllers, edge/FNs, and sensors connected via various wired or wireless communication mechanisms to create the IoT.
2) *Data Layer:* This layer collects IoT data through transactions from the lower layer and encases encrypted data with hashes, digital signatures, and asymmetric cryptographic algorithms.
3) *Network Layer:* The network layer functions as a peer-to-peer network laid on top of the communication layer. A peer-to-peer model is used because of the foundational need for decentralization, which can only be achieved using a network architecture that allows peers to share resources without third-party assistance. Peers are able to function as service providers or requestors and are organized based on support functions, such as wallet, database, miner, or routing.
4) *Consensus Layer:* The consensus layer is responsible for managing the distributed consensus needed to verify a block's trustworthiness and ensuring that every peer has a correct ledger copy. Even so, nodes and agents can have divergent views of a system's status (i.e., forks) because of malicious nodes, network faults, or communication latency. This makes avoiding forks one of the main challenges facing consensus algorithms.
5) *Contract Layer:* This layer handles digital currency as well as the generation and management of smart contracts.
6) *Application Layer:* The application layer gives users services needed across various industrial areas, such as manufacturing, logistics, healthcare, or supply chains.

IoT and blockchain can be used together via tight integration or loose integration (See Fig. 10) [2], [32], [36], [119], [120].

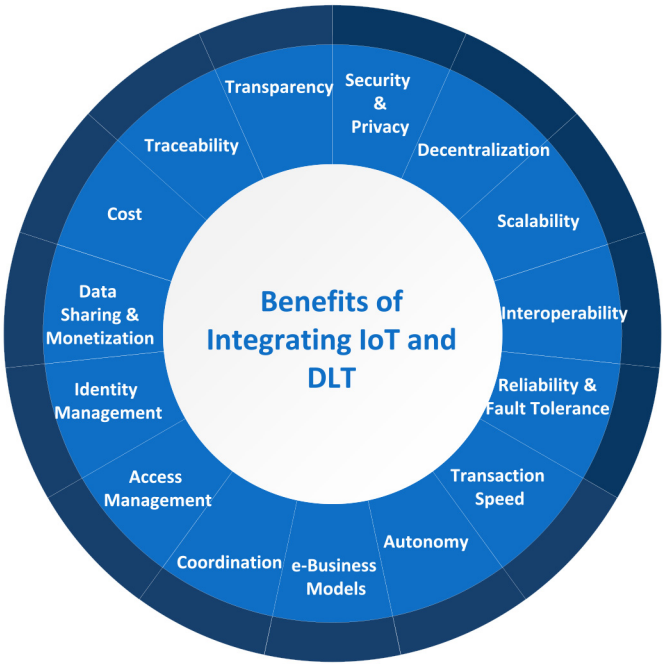


Fig. 8. Benefits of the integration of blockchain and IoT.

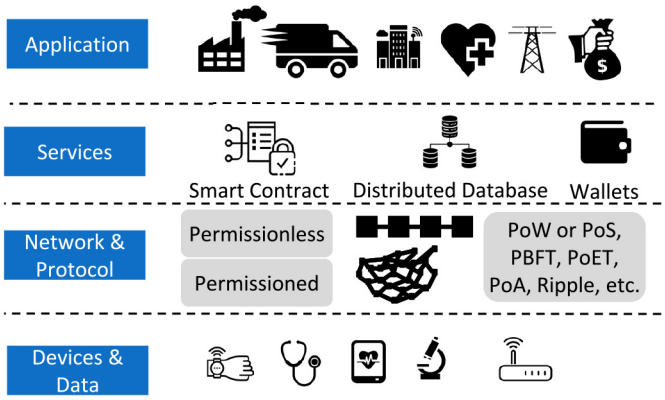


Fig. 9. Blockchain IoT layers.

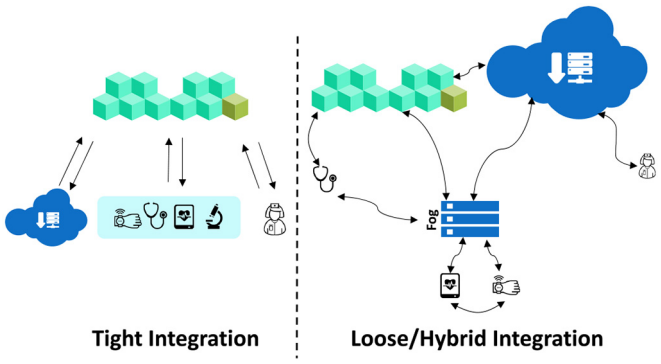


Fig. 10. Blockchain and IoT integration schemes.

1) *Tight Integration*: Tight Integration requires that all participation communications occur through blockchain. This means that every IoT device is a blockchain peer, and every IoT interaction is recorded in the chain to guarantee accountability. Tight integration enables the monitoring of all communication between IoT services and devices.
2) *Loose Integration:* Recording every IoT interaction expands bandwidth and storage needs, which is not always possible in every use case. Therefore, in loose integration, IoT devices with plenty of resources, such as FNs or gateways that register IoT devices, can handle consensus or other heavy power-use interactions. Devices with fewer resources are not required to process or hold blockchain records. IoT devices are allowed to communicate off-chain or through blockchain gateways. While this requires discovery and routing protocols, it ensures low latency between devices and generates an option for recording interactions. In short, all transmitted data does not have to be maintained on the blockchain. The blockchain serves as a control mechanism because smart contracts function as programmable logic when data are sent via peer-to-peer technology. This enables users to choose to use blockchain or a more traditional cloud-IoT protocol to support IoT device interactions. Loose integration combines the best of decentralized recording using blockchain and IoT real-time communication. This kind of integration works best in scenarios with more frequent interaction, low latency, high throughput, and dependable IoT data. However, this level of integration has to optimize the distance between blockchain interactions and those happening in real-time. Additionally, the extent of decentralization achieved through loose integration is not as impregnable as transactions transmitted directly to the blockchain.

VI. Case Study

The convergence of the Internet of Medical Things (IoMT), edge–fog–cloud, AI, and DLTs generated new opportunities for personalized eHealth, transforming the traditional hospital-centric treatment to patient-centric personalized healthcare. The following case study on stress monitoring and management illustrates how the healthcare industry can benefit from this innovative combination. This case study also sheds light on FL as a promising privacy-preserving distributed learning technique to tackle privacy/security challenges of the healthcare domain.

Stress is linked to serious health problems, such as depression, anxiety, obesity, heart disease, Alzheimer's disease, diabetes, gastrointestinal problems, and asthma [121]. The current system of mental healthcare utilizes a reactive approach focused on acute, symptom-centered care. A passive approach to providing mental healthcare is one big disadvantage of such a system because individuals often only become aware of their condition when it has become severe enough that they can see a need for care. In addition, a conventional periodic medical or psychotherapy treatment model is based on the idea that healthcare providers engage with patients at appointments that may be spaced apart by weeks. This framework for mental healthcare puts a heavy burden of responsibility on the provider to integrate a significant amount of data in a short period. It also places a burden on the patient responsible for accurately assessing and reporting their experiences. Moreover, the existing mental healthcare system is also limited by a lack of holistic information about the context of daily life when it comes to small changes in mental health, emotions, and symptoms throughout activities of daily living [121], [122].

Recent technological advancements in AIoT and the embracement of smart and connected health (SCH) solutions, promoted by the predictive, preventative, personalized, and participatory (P4) model, offer novel approaches capable of addressing such limitations by integrating individual lifestyle choices and genetics to develop customized interventions. In this context, IoMT and Wearable Internet of Wearable Things (WIoT) offer a ubiquitous healthcare surveillance system based on a rich set of smart IoT medical devices and wearables enabling individuals to monitor their stress-related data anytime, from anywhere. Thanks to the sovereign data sharing/exchanging technologies, cloud platform, and blockchain, healthcare providers and patients whether they are traveling on the road or relaxing at home, always have access to the collected mental health data. WIoT, IoMT, and SCH naturally produce vast amounts of data that are too big for humans to handle. Fortunately, AI can tame the data deluge and wring insight while growing ever more intelligent about what they ingest [28].

A. Data Set

We utilized the wearable stress and affect detection (WESAD) data set as a benchmark to illustrate customized privacy-aware healthcare data analysis using FL [123]. The data set is publicly available on the UCI ML Repository [124]. It contains multimodal physiological signals measured by two types of devices: 1) RespiBAN, a wearable device placed on the chest, and 2) Empatica E4, a wristband. RespiBAN measures ECG, EDA, respiratory signal, EMG, accelerometer data, and temperature sampled at a rate of 700 Hz. Empatica E4 measures EDA, BVP, accelerometer data, and temperature. For

demonstration purposes, we used only the ECG signal in our study. The data set includes data recorded from 17 participants, with each wearing a RespiBAN and an Empatica E4. Two participants were excluded due to low data quality. All subjects performed a series of tasks, where each was associated with a stress level. Stress levels studied in the experiment are stress, amusement, and the baseline. The stress condition was induced by performing public speaking and mental arithmetic tasks according to the trier social stress test (TSST) [125]. The amusement condition was achieved by watching funny video clips. The baseline was measured when the participants were sitting, standing, or reading magazines. The order of the groups of tasks was as illustrated in [123], with a meditation interval between each condition session. The devices recorded a total of 100-min data from each participant.

B. Setup and Evaluation Metrics

Computation Platform: We performed the FedAvg algorithm on a server equipped with two NVIDIA TITAN V, each with 12-GB memory and two GeForce RTX 2080 Ti, each with 11-GB memory. We segmented raw ECG based on R-peaks and converted each segment to an image of size. 70% of the samples were used as the training data. The training samples were further partitioned to each client based on their originality. We utilized the convolutional neural network (CNN) architecture proposed in our previous work in [121] as the local client model. The client models were trained with a batch size of 32 for five epochs in each communication round between the client and the central server. We performed 80 communication rounds during the training. Based on [121], we utilized cross-entropy loss for training the CNN. Stochastic gradient descent (SGD) optimizer with a learning rate of 10–5 was applied. All the above-mentioned hyperparameters were chosen based on the random search. All the experiments were implemented using PyTorch.

C. Performance Evaluation

1) Stress Evaluation Using FedAvg: To investigate the performance of FL, we implemented the most widely used and effective algorithm FedAvg [126], utilizing all 15 clients with their associated data. We trained the same CNN architecture in a conventional setting for comparison. The classification performance in terms of accuracy and F1 score is presented in Table I. The generalized model using conventional training method achieved an accuracy of 94.26% and the FedAvg model using 15 clients obtained an accuracy of 83.9%. We see that both the accuracy and the F1 score of FedAvg are around 10% lower than performance in the non-FL setting. This difference is due to data heterogeneity. In FL, the distribution of the data from each client may differ due to individual physiological patterns. After averaging parameters from all local client models, the central model may not yield the best parameters for all clients, resulting in a drop in the performance of the stress level classification. However, in the conventional model training, all the data are treated as from a single distribution. The model is thus able to generalize better compared to the model using FedAvg.

TABLE I
PERFORMANCE COMPARISON BETWEEN CONVENTIONAL AND FL TRAINING

	Accuracy (%)	F1 score
Non-FL	94.26	0.943
FedAvg	83.9	0.847

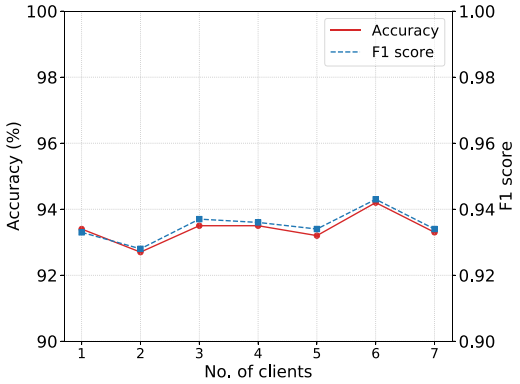


Fig. 11. Performance of FedAvg using data partitioned to a different number of clients. Each client holds data from multiple subjects, e.g., when client = 1, the client holds data from all 15 subjects; when client = 3, each client holds data from five subjects.

2) Effect of Partitioning Data to a Different Number of Clients: Conventional training can lead to a generalized model with high performance. However, the FedAvg model trained on individuals caused degradation in task performance due to data heterogeneity. To maintain a balance between the two schemes, we propose to group the subjects to achieve customization with high performance. We randomly partitioned data from all 15 participants to n clients, where $n \in [1, 7]$, i.e., one client holds data from multiple participants. The stress evaluation performance using FedAvg with data grouped into a different number of clients is as shown in Fig. 11.

We observe both the accuracy and the F1 score fluctuate as the number of clients increases. This pattern is due to the different partitioning of the data. As mentioned earlier in the section, data heterogeneity degrades classification performance. We see all data partitioning presented in Fig. 11 yield performances around 94%, close to the accuracy obtained in the non-FL setting. By comparing Fig. 11 with Table I, we notice data grouping significantly increases the FedAvg performance by 10%, compared with using individual 15 clients. The results suggest when having a client with skewed data distribution, aggregating data from two subjects is sufficient to minimize the effect of data heterogeneity. According to Fig. 11, the most effective way of grouping is to partition the data into six clients, achieving an accuracy of 94.2%.

We next report the local client training and aggregation time in one communication round in FedAvg shown in Fig. 12. The aggregation time increases as the number of clients increases. The fluctuation of the client training time can be attributed to the noise from uncontrollable processes running on the platform.

3) Effect of the Number of Participating Clients: We have examined the impact of data partitioning on stress-level classification performance. We next investigate the effect of utilizing

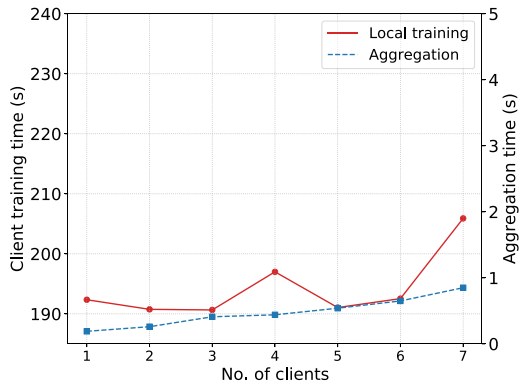


Fig. 12. Runtime of FedAvg with a different number of clients in one communication round.

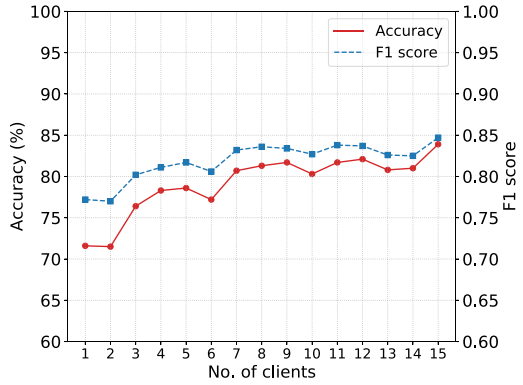


Fig. 13. Performance of FedAvg using a different number of participating clients. All clients hold their own data, e.g., when client = 1, the client holds data from one subject; when client = 3, each client holds data from one subject.

a different number of participating clients on classification performance, i.e., all 15 clients hold their own data. In each communication round, we randomly selected n clients and evaluated the performance in terms of accuracy and $F1$ score, where $n \in [1, 15]$.

The accuracy and $F1$ score using a different number of participating clients are as shown in Fig. 13. We see in the figure that the performance increases when the number of clients increases. This trend is due to the increased data from the new client fed to the model during the training. We see degradation in performance when increasing five clients to six and increasing nine clients to ten. The degradation is caused by the difference between the distributions of the data from the new client and the existing data. However, increased data samples still positively affect the model, resulting in enhanced overall performance.

We present the runtime of local client training and the aggregation time in one communication round in Fig. 14. The aggregation time positively correlates with the number of participating clients. The client training time fluctuates but still follows an overall increasing trend. The fluctuation is caused by the difference in the amount of data each participating client holds. As mentioned earlier in the section, we randomly selected 70% of the data for training. The training data from each client are therefore unbalanced. Some clients have fewer

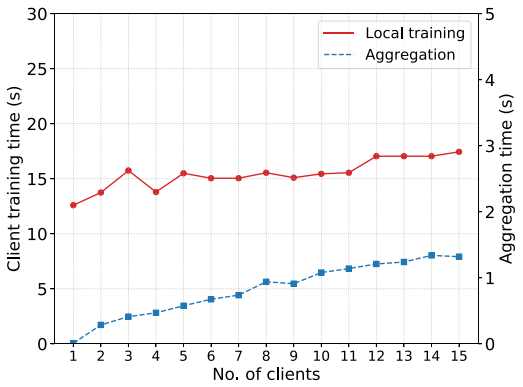


Fig. 14. Runtime of FedAvg using a different number of participating clients in one communication round.

data samples than other clients. When selecting a client with a larger data size, the local training time increases. When selecting all 15 clients, we recorded the local runtime of the client with the most abundant data, resulting in the longest local training time.

In addition, we consider the latency of transferring model parameters between the client and the central server. To compute the latency, we measured the upload and download bandwidths of Wi-Fi using TestMy.net [127]. We computed the size of the model parameters to be 1151 MB, assuming all parameters are 32-bit floats. The latency of updating a client model is thus 6.4 min.

We present Fig. 15 to show a clear pattern between the FedAvg accuracy and total runtime. The total runtime is composed of local client training, global server aggregation, and model parameter transmission, multiplied by 80 communication rounds, as indicated earlier in the section. We observe that the FedAvg total runtime slowly increases as the number of clients increases. Based on our experiment, the main factor for the overall runtime is the model parameter transmission. Although we see more obvious increasing trends in Figs. 13 and 14, they have smaller contributions to the total runtime. For future experiments, we propose to optimize neural network architecture to obtain a smaller number of parameters to reduce the overall runtime.

VII. Conclusion

Although IoT, AI, edge–fog–cloud, and DLT/blockchain have evolved independently as four distinct technologies, they are often complementary topics in technology, and one should not be discussed without the other. As these technologies increasingly converge over time to complement each other, new paradigms emerge, leading to unprecedented opportunities for the development of new kinds of platforms and innovative services and products disrupting whole industries. edge–fog–cloud computing collaborates with IoT technology to provide efficient data processing, elastic storage, cost-effective scalability, data-service integration, and pay-as-you-go features. With the continuous growth of IoT devices, DLT/blockchain can be utilized to help with the security challenges of IoT and offer a decentralized alternative to conventional centralized IoT solutions. The confluence of

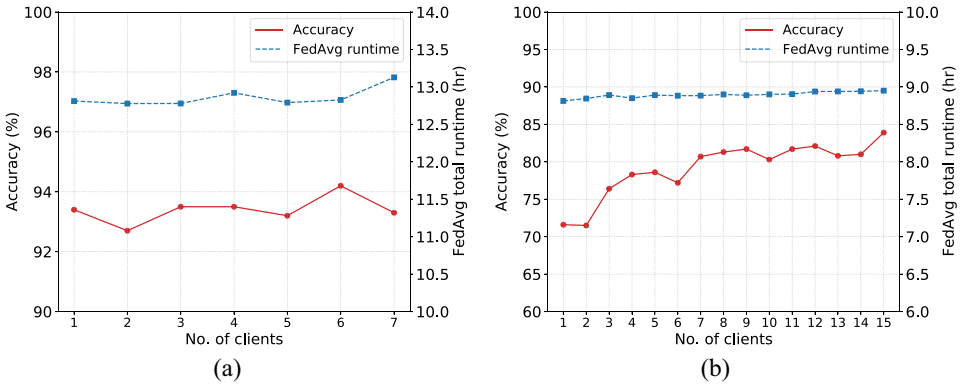


Fig. 15. FedAvg accuracy and total runtime. (a) FedAvg accuracy and runtime with data partitioned to a different number of clients. (b) FedAvg accuracy and runtime with a different number of participating clients.

IoT and AI gives rise to AI of Things and Augmented Intelligence of Things, becoming a catalyst for a new era of technological change. The AIoT systems solve a wide range of complex problems by offering a better insight into data and adding a smart layer of intelligence on top of connected devices, redefining the way industries, businesses, and economies function. In line with these developments, this article comprehensively discussed all critical aspects, opportunities, challenges, applications, solutions, and existing architectures of the fusion of IoT, AI, edge–fog–cloud, and DLT/blockchain. Furthermore, a holistic case study on privacy-preserving IoT-based stress monitoring was presented to address better the role of federated analysis, MPC, and data sharing in the era of IoT, AI, edge–fog–cloud, and DLT/blockchain convergence.

REFERENCES

[1] F. Firouzi, B. Farahani, and A. Marinšek, "The convergence and interplay of edge, fog, and cloud in the AI-driven Internet of Things (IoT)," *Inf. Syst.*, vol. 107, Jul. 2022, Art. no. 101840.

[2] B. Farahani, F. Firouzi, and M. Luecking, "The convergence of IoT and distributed ledger technologies (DLT): Opportunities, challenges, and solutions," *J. Netw. Comput. Appl.*, vol. 177, Mar. 2021, Art. no. 102936.

[3] F. Firouzi, B. Farahani, M. Weinberger, G. DePace, and F. S. Aliee, "IoT fundamentals: Definitions, architectures, challenges, and promises," in *Intelligent Internet of Things*. Cham, Switzerland: Springer, 2020, pp. 3–50.

[4] P. Sandner, J. Gross, and R. Richter, "Convergence of blockchain, IoT, and AI," *Front. Blockchain*, vol. 3, Sep. 2020, Art. no. 522600.

[5] "Internet of Things—Number of Connected Devices Worldwide 2015–2025." [Online]. Available: https://www.statista.com/ (Accessed: Oct. 15, 2021).

[6] "oneM2M System Enhancement to Support Privacy Data Protection Regulations (eDPR)." [Online]. Available: https://onem2m.org/ (Accessed: Oct. 15, 2021).

[7] F. Firouzi and B. Farahani, "Architecting IoT cloud," in *Intelligent Internet of Things*. Cham, Switzerland: Springer, 2020, pp. 173–241.

[8] I. Martinez, A. S. Hafid, and A. Jarray, "Design, resource management, and evaluation of fog computing systems: A survey," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2494–2516, Feb. 2021.

[9] A. Yousefpour *et al.*, "All one needs to know about fog computing and related edge computing paradigms: A complete survey," *J. Syst. Archit.*, vol. 98, pp. 289–330, Sep. 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1383762118306349

[10] M. Aazam, S. Zeadally, and K. A. Harras, "Offloading in fog computing for IoT: Review, enabling technologies, and research opportunities," *Future Gener. Comput. Syst.*, vol. 87, pp. 278–289, Oct. 2018.

[11] R. K. Naha *et al.*, "Fog computing: Survey of trends, architectures, requirements, and research directions," *IEEE Access*, vol. 6, pp. 47980–48009, 2018.

[12] P. Bellavista, J. Berrocal, A. Corradi, S. K. Das, L. Foschini, and A. Zanni, "A survey on fog computing for the Internet of Things," *Pervasive Mobile Comput.*, vol. 52, pp. 71–99, Jan. 2019.

[13] A. Aljumah and T. A. Ahanger, "Fog computing and security issues: A review," in *Proc. 7th Int. Conf. Comput. Commun. Control (ICCCC)*, 2018, pp. 237–239.

[14] F. A. Salaht, F. Desprez, and A. Lebre, "An overview of service placement problem in fog and edge computing," *ACM Comput. Surv.*, vol. 53, no. 3, pp. 1–35, 2020.

[15] F. Firouzi, B. Farahani, M. Ibrahim, and K. Chakrabarty, "Keynote paper: From EDA to IoT eHealth: Promises, challenges, and solutions," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 12, pp. 2965–2978, Dec. 2018.

[16] A. Botta, W. De Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and Internet of Things: A survey," *Future Gener. Comput. Syst.*, vol. 56, pp. 684–700, Mar. 2016.

[17] E. Elmroth, P. Leitner, S. Schulte, and S. Venugopal, "Connecting fog and cloud computing," *IEEE Cloud Comput.*, vol. 4, no. 2, pp. 22–25, Mar./Apr. 2017.

[18] X. Masip-Bruin, E. Marín-Tordera, G. Tashakor, A. Jukan, and G.-J. Ren, "Foggy clouds and cloudy fogs: A real need for coordinated management of fog-to-cloud computing systems," *IEEE Wireless Commun.*, vol. 23, no. 5, pp. 120–128, Oct. 2016.

[19] V. Stantchev, A. Barnawi, S. Ghulam, J. Schubert, and G. Tamm, "Smart items, fog and cloud computing as enablers of servitization in healthcare," *Sens. Transducers*, vol. 185, no. 2, pp. 121–128, 2014.

[20] F. Firouzi, B. Farahani, F. Ye, and M. Barzegari, "Machine learning for IoT," in *Intelligent Internet of Things*. Cham, Switzerland: Springer, 2020, pp. 243–313.

[21] F. Shi *et al.*, "Recent progress on the convergence of the Internet of Things and artificial intelligence," *IEEE Netw.*, vol. 34, no. 5, pp. 8–15, Sep./Oct. 2020.

[22] M. J. Kaur, V. P. Mishra, and P. Maheshwari, "The convergence of digital twin, IoT, and machine learning: Transforming data into action," in *Digital Twin Technologies and Smart Cities*. Cham, Switzerland: Springer, 2020, pp. 3–17.

[23] F. Firouzi, B. Farahani, M. Barzegari, and M. Daneshmand, "AI-driven data monetization: The other face of data in IoT-based smart and connected health," *IEEE Internet Things J.*, vol. 9, no. 8, pp. 5581–5599, Apr. 2022.

[24] C. Resende *et al.*, "TIP4.0: Industrial Internet of Things platform for predictive maintenance," *Sensors*, vol. 21, no. 14, p. 4676, 2021.

[25] T. Zonta, C. A. da Costa, R. da Rosa Righi, M. J. de Lima, E. S. da Trindade, and G. P. Li, "Predictive maintenance in the industry 4.0: A systematic literature review," *Comput. Ind. Eng.*, vol. 150, Dec. 2020, Art. no. 106889.

[26] S. H. Dolatabadi and I. Budinska, "Systematic literature review predictive maintenance solutions for SMEs from the last decade," *Machines*, vol. 9, no. 9, p. 191, 2021.

[27] K. Rabah, "Convergence of AI, IoT, big data and blockchain: A review," *Lake Inst. J.*, vol. 1, no. 1, pp. 1–18, 2018.

[28] B. Farahani, F. Firouzi, V. Chang, M. Badaroglu, N. Constant, and K. Mankodiya, "Towards fog-driven IoT eHealth: Promises and challenges of IoT in medicine and healthcare," *Future Gener. Comput. Syst.*, vol. 78, pp. 659–676, Jan. 2018.

[29] O. Debauche, S. Mahmoudi, S. A. Mahmoudi, P. Manneback, and F. Lebeau, "A new edge architecture for AI-IoT services deployment," *Procedia Comput. Sci.*, vol. 175, pp. 10–19, Jan. 2020.

[30] L. Xiao, X. Wan, X. Lu, Y. Zhang, and D. Wu, "IoT security techniques based on machine learning: How do IoT devices use AI to enhance security?" *IEEE Signal Process. Mag.*, vol. 35, no. 5, pp. 41–49, Sep. 2018.

[31] X. Liu, B. Farahani, and F. Firouzi, "Distributed ledger technology," in *Intelligent Internet of Things*. Cham, Switzerland: Springer, 2020, pp. 393–431.

[32] M. S. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli, and M. H. Rehmani, "Applications of blockchains in the Internet of Things: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1676–1717, 2nd Quart., 2018.

[33] H.-N. Dai, Z. Zheng, and Y. Zhang, "Blockchain for Internet of Things: A survey," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8076–8094, Oct. 2019.

[34] W. Viriyasitavat, L. D. Xu, Z. Bi, and D. Hoonsopon, "Blockchain technology for applications in Internet of Things—Mapping from system design perspective," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8155–8168, Oct. 2019.

[35] X. Wang *et al.*, "Survey on blockchain for Internet of Things," *Comput. Commun.*, vol. 136, pp. 10–29, Feb. 2019.

[36] Z. Xiong, Y. Zhang, N. C. Luong, D. Niyato, P. Wang, and N. Guizani, "The best of both worlds: A general architecture for data management in blockchain-enabled Internet-of-Things," *IEEE Netw.*, vol. 34, no. 1, pp. 166–173, Jan./Feb. 2020.

[37] F. Firouzi, K. Chakrabarty, and S. Nassif, *Intelligent Internet of Things: From Device to Fog and Cloud*. Cham, Switzerland: Springer, 2020.

[38] "A Guide to Data Monetization." [Online]. Available: https://blog.bosch-si.com/business-models/a-guide-to-data-monetization/ (Accessed: Oct. 15, 2021).

[39] K. Fan, S. Wang, Y. Ren, H. Li, and Y. Yang, "Medblock: Efficient and secure medical data sharing via blockchain," *J. Med. Syst.*, vol. 42, no. 8, p. 136, 2018.

[40] Q. Xia, E. B. Sifah, K. O. Asamoah, J. Gao, X. Du, and M. Guizani, "MeDShare: Trust-less medical data sharing among cloud service providers via blockchain," *IEEE Access*, vol. 5, pp. 14757–14767, 2017.

[41] X. Liang, J. Zhao, S. Shetty, J. Liu, and D. Li, "Integrating blockchain for data sharing and collaboration in mobile healthcare applications," in *Proc. IEEE 28th Annu. Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, 2017, pp. 1–5.

[42] Q. Xia, E. B. Sifah, A. Smahi, S. Amofa, and X. Zhang, "BBDS: Blockchain-based data sharing for electronic medical records in cloud environments," *Information*, vol. 8, no. 2, p. 44, 2017.

[43] S. Wang, Y. Zhang, and Y. Zhang, "A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems," *IEEE Access*, vol. 6, pp. 38437–38450, 2018.

[44] H. Shafagh, L. Burkhalter, A. Hithnawi, and S. Duquennoy, "Towards blockchain-based auditable storage and sharing of IoT data," in *Proc. Cloud Comput. Security Workshop*, 2017, pp. 45–50.

[45] T. Sultana, A. Almogren, M. Akbar, M. Zuair, I. Ullah, and N. Javaid, "Data sharing system integrating access control mechanism using blockchain-based smart contracts for IoT devices," *Appl. Sci.*, vol. 10, no. 2, p. 488, 2020.

[46] "GAIA-X." [Online]. Available: https://www.gaia-x.eu/ (Accessed: Oct. 15, 2021).

[47] "International Data Space." [Online]. Available: https://internationaldataspaces.org/ (Accessed: Oct. 15, 2021).

[48] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[49] J. Cao, Q. Zhang, and W. Shi, *Edge Computing: A Primer*. Cham, Switzerland: Springer, 2018.

[50] F. Firouzi, B. Farahani, E. Panahi, and M. Barzegari, "Task offloading for edge-fog-cloud interplay in the healthcare Internet of Things (IoT)," in *Proc. IEEE Int. Conf. Omni-Layer Intell. Syst. (COINS)*, 2021, pp. 1–8.

[51] B. Tang, Z. Chen, G. Hefferman, T. Wei, H. He, and Q. Yang, "A hierarchical distributed fog computing architecture for big data analysis in smart cities," in *Proc. ASE BigData SocialInform.*, 2015, pp. 1–6.

[52] K. Intharawijitr, K. Iida, and H. Koga, "Analysis of fog model considering computing and communication latency in 5G cellular networks," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, 2016, pp. 1–4.

[53] E. Saurez, K. Hong, D. Lillethun, U. Ramachandran, and B. Ottenwälder, "Incremental deployment and migration of geo-distributed situation awareness applications in the fog," in *Proc. 10th ACM Int. Conf. Distrib. Event-Based Syst.*, 2016, pp. 258–269.

[54] S. Tomovic, K. Yoshigoe, I. Maljevic, and I. Radusinovic, "Software-defined fog network architecture for IoT," *Wireless Pers. Commun.*, vol. 92, no. 1, pp. 181–196, 2017.

[55] *OpenFog Reference Architecture for Fog Computing, Architecture Working Group*, OpenFog Consortium, Fremont, CA, USA, 2017.

[56] S. Tuli, R. Mahmud, S. Tuli, and R. Buyya, "FogBus: A blockchain-based lightweight framework for edge and fog computing," *J. Syst. Softw.*, vol. 154, pp. 22–36, Aug. 2019.

[57] H. Ning, X. Ye, A. B. Sada, L. Mao, and M. Daneshmand, "An attention mechanism inspired selective sensing framework for physical-cyber mapping in Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 9531–9544, Dec. 2019.

[58] I. Stypsanelli, O. Brun, S. Medjiah, and B. J. Prabhu, "Capacity planning of fog computing infrastructures under probabilistic delay guarantees," in *Proc. IEEE Int. Conf. Fog Comput. (ICFC)*, 2019, pp. 185–194.

[59] A. Brogi, S. Forti, and A. Ibrahim, "Deploying fog applications: How much does it cost by the way?" *Small*, vol. 1, no. 2, p. 20, 2018.

[60] M. Noreikis, Y. Xiao, and A. Ylä-Jaäiski, "QoS-oriented capacity planning for edge computing," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2017, pp. 1–6.

[61] K. Poularakis, J. Llorca, A. M. Tulino, I. Taylor, and L. Tassiulas, "Joint service placement and request routing in multi-cell mobile edge computing networks," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, 2019, pp. 10–18.

[62] Z. Nezami, K. Zamanifar, K. Djemame, and E. Pournaras, "Decentralized edge-to-cloud load balancing: Service placement for the Internet of Things," *IEEE Access*, vol. 9, pp. 64983–65000, 2021.

[63] T. He, H. Khamfroush, S. Wang, T. La Porta, and S. Stein, "It's hard to share: Joint service placement and request scheduling in edge clouds with sharable and non-sharable resources," in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2018, pp. 365–375.

[64] M. Taneja and A. Davy, "Resource aware placement of IoT application modules in fog-cloud computing paradigm," in *Proc. IFIP/IEEE Symp. Integr. Netw. Serv. Manag. (IM)*, 2017, pp. 1222–1228.

[65] F. Faticanti, M. Savi, F. De Pellegrini, P. Kochovski, V. Stankovski, and D. Siracusa, "Deployment of application microservices in multi-domain federated fog environments," in *Proc. Int. Conf. Omni-Layer Intell. Syst. (COINS)*, 2020, pp. 1–6.

[66] T. Subramanya, D. Harutyunyan, and R. Riggio, "Machine learning-driven service function chain placement and scaling in MEC-enabled 5G networks," *Comput. Netw.*, vol. 166, Jan. 2020, Art. no. 106980.

[67] Z. Zhang, L. Ma, K. K. Leung, L. Tassiulas, and J. Tucker, "Q-placement: Reinforcement-learning-based service placement in software-defined networks," in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2018, pp. 1527–1532.

[68] J.-Y. Baek, G. Kaddoum, S. Garg, K. Kaur, and V. Gravel, "Managing fog networks using reinforcement learning based load balancing algorithm," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2019, pp. 1–7.

[69] L. Huang, S. Bi, and Y.-J. A. Zhang, "Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks," *IEEE Trans. Mobile Comput.*, vol. 19, no. 11, pp. 2581–2593, Nov. 2020.

[70] M. Tang and V. W. S. Wong, "Deep reinforcement learning for task offloading in mobile edge computing systems," *IEEE Trans. Mobile Comput.*, vol. 21, no. 6, pp. 1985–1997, Jun. 2022.

[71] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 869–904, 2nd Quart., 2020.

[72] C. Jiang, X. Cheng, H. Gao, X. Zhou, and J. Wan, "Toward computation offloading in edge computing: A survey," *IEEE Access*, vol. 7, pp. 131543–131558, 2019.

[73] J. Almutairi and M. Aldossary, "A novel approach for IoT tasks offloading in edge-cloud environments," *J. Cloud Comput.*, vol. 10, no. 1, pp. 1–19, 2021.

[74] H. Wu, Y. Fan, Y. Wang, H. Ma, and L. Xing, "A comprehensive review on edge caching from the perspective of total process: Placement, policy and delivery," *Sensors*, vol. 21, no. 15, p. 5033, 2021.

[75] I. H. Sarker, A. I. Khan, Y. B. Abushark, and F. Alsolami, "Internet of Things (IoT) security intelligence: A comprehensive overview, machine learning solutions and research directions," *Mobile Netw. Appl.*, to be published.

[76] B. K. Mohanta, D. Jena, U. Satapathy, and S. Patnaik, "Survey on IoT security: Challenges and solution using machine learning, artificial intelligence and blockchain technology," *Internet Things*, vol. 11, Sep. 2020, Art. no. 100227.

[77] M. Kuzlu, C. Fair, and O. Guler, "Role of artificial intelligence in the Internet of Things (IoT) cybersecurity," *Discov. Internet Things*, vol. 1, no. 1, pp. 1–14, 2021.

[78] S. S. Gopalan, A. Raza, and W. Almobaideen, "IoT security in healthcare using AI: A survey," in *Proc. Int. Conf. Commun. Signal Process. Appl. (ICCSPA)*, 2021, pp. 1–6.

[79] H. HaddadPajouh, R. Khayami, A. Dehghantanha, K.-K. R. Choo, and R. M. Parizi, "AI4SAFE-IoT: An AI-powered secure architecture for edge layer of Internet of Things," *Neural Comput. Appl.*, vol. 32, no. 20, pp. 16119–16133, 2020.

[80] Z. Lv, L. Qiao, A. K. Singh, and Q. Wang, "AI-empowered IoT security for smart cities," *ACM Trans. Internet Technol.*, vol. 21, no. 4, pp. 1–21, 2021.

[81] H. Wu, H. Han, X. Wang, and S. Sun, "Research on artificial intelligence enhancing Internet of Things security: A survey," *IEEE Access*, vol. 8, pp. 153826–153848, 2020.

[82] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on IoT security: Application areas, security threats, and solution architectures," *IEEE Access*, vol. 7, pp. 82721–82743, 2019.

[83] 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*, vol. 107, no. 8, pp. 1738–1762, Aug. 2019.

[84] S. Teerapittayanon, B. McDanel, and H.-T. Kung, "BranchyNet: Fast inference via early exiting from deep neural networks," in *Proc. 23rd Int. Conf. Pattern Recognit. (ICPR)*, 2016, pp. 2464–2469.

[85] Y. Deng, M. M. Kamani, and M. Mahdavi, "Adaptive personalized federated learning," 2020, *arXiv:2003.13461*.

[86] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Inference attacks against collaborative learning," 2018, *arXiv:1805.04049*.

[87] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Security*, 2015, pp. 1310–1321.

[88] N. Rieke *et al.*, "The future of digital health with federated learning," *NPJ Digit. Med.*, vol. 3, no. 1, pp. 1–7, 2020.

[89] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–19, 2019.

[90] K. Bonawitz *et al.*, "Towards federated learning at scale: System design," 2019, *arXiv:1902.01046*.

[91] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[92] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," 2016, *arXiv:1610.05492*.

[93] Y. Liu, Z. Ai, S. Sun, S. Zhang, Z. Liu, and H. Yu, "FedCoin: A peer-to-peer payment system for federated learning," in *Federated Learning*. Cham, Switzerland: Springer, 2020, pp. 125–138.

[94] A. G. Roy, S. Siddiqui, S. Pölsterl, N. Navab, and C. Wachinger, "BrainTorrent: A peer-to-peer environment for decentralized federated learning," 2019, *arXiv:1905.06731*.

[95] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," *Future Gener. Comput. Syst.*, vol. 115, pp. 619–640, Feb. 2021.

[96] J. Jiang, L. Hu, C. Hu, J. Liu, and Z. Wang, "BACombo—Bandwidth-aware decentralized federated learning," *Electronics*, vol. 9, no. 3, p. 440, 2020.

[97] E. Diao, J. Ding, and V. Tarokh, "HeteroFL: Computation and communication efficient federated learning for heterogeneous clients," 2020, *arXiv:2010.01264*.

[98] A. Nadian-Ghomsheh, B. Farahani, and M. Kavian, "A hierarchical privacy-preserving IoT architecture for vision-based hand rehabilitation assessment," *Multimedia Tools Appl.*, vol. 80, pp. 31357–31380, Feb. 2021.

[99] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes: Theory and implementation," *ACM Comput. Surv.*, vol. 51, no. 4, pp. 1–35, 2018.

[100] I. Damgard, M. Geisler, and M. Kroigard, "Homomorphic encryption and secure comparison," *Int. J. Appl. Cryptogr.*, vol. 1, no. 1, pp. 22–31, 2008.

[101] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Trans. Inf. Forensics Security*, vol. 13, pp. 1333–1345, 2018.

[102] P. Li *et al.*, "Multi-key privacy-preserving deep learning in cloud computing," *Future Gener. Comput. Syst.*, vol. 74, pp. 76–85, Sep. 2017.

[103] A. C. Yao, "Protocols for secure computations," in *Proc. 23rd Annu. Symp. Found. Comput. Sci. (SFCS)*, 1982, pp. 160–164.

[104] O. Goldreich. "Secure Multi-Party Computation." 1998. [Online]. Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.11.2201&rep=rep1&type=pdf (Accessed: Oct. 15, 2021).

[105] A. Ben-David, N. Nisan, and B. Pinkas, "FairplayMP: A system for secure multi-party computation," in *Proc. 15th ACM Conf. Comput. Commun. Security*, 2008, pp. 257–266.

[106] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[107] M. Yang, L. Lyu, J. Zhao, T. Zhu, and K.-Y. Lam, "Local differential privacy and its applications: A comprehensive survey," 2020, *arXiv:2008.03686*.

[108] A. Friedman and A. Schuster, "Data mining with differential privacy," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2010, pp. 493–502.

[109] M. Abadi *et al.*, "Deep learning with differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2016, pp. 308–318.

[110] C. Dwork, "Differential privacy: A survey of results," in *Proc. Int. Conf. Theory Appl. Models Comput.*, 2008, pp. 1–19.

[111] Z. Ji, Z. C. Lipton, and C. Elkan, "Differential privacy and machine learning: A survey and review," 2014, *arXiv:1412.7584*.

[112] M. Al-Rubaie and J. M. Chang, "Privacy-preserving machine learning: Threats and solutions," *IEEE Security Privacy*, vol. 17, no. 2, pp. 49–58, Mar./Apr. 2019.

[113] X. Xiong, S. Liu, D. Li, Z. Cai, and X. Niu, "A comprehensive survey on local differential privacy," *Security Commun. Netw.*, vol. 2020, p. 29, 2020. [Online]. Available: https://doi.org/10.1155/2020/8829523

[114] N. Saleheen *et al.*, "mSieve: Differential behavioral privacy in time series of mobile sensor data," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2016, pp. 706–717.

[115] J. Zhao, Y. Chen, and W. Zhang, "Differential privacy preservation in deep learning: Challenges, opportunities and solutions," *IEEE Access*, vol. 7, pp. 48901–48911, 2019.

[116] M. U. Hassan, M. H. Rehmani, and J. Chen, "Differential privacy techniques for cyber physical systems: A survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 746–789, 1st Quart., 2020.

[117] A. D. Sarwate and L. Sankar, "A rate-disortion perspective on local differential privacy," in *Proc. 52nd Annu. Allerton Conf. Commun. Control Comput. (Allerton)*, 2014, pp. 903–908.

[118] Q. Zhu, S. W. Loke, R. Trujillo-Rasua, F. Jiang, and Y. Xiang, "Applications of distributed ledger technologies to the Internet of Things: A survey," *ACM Comput. Surv.*, vol. 52, no. 6, pp. 1–34, 2019.

[119] M. Naz *et al.*, "A secure data sharing platform using blockchain and interplanetary file system," *Sustainability*, vol. 11, no. 24, pp. 1–24, 2019.

[120] N. B. Truong, K. Sun, G. M. Lee, and Y. Guo, "GDPR-compliant personal data management: A blockchain-based solution," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1746–1761, 2020.

[121] S. Jiang, F. Firouzi, K. Chakrabarty, and E. B. Elbogen, "A resilient and hierarchical IoT-based solution for stress monitoring in everyday settings," *IEEE Internet Things J.*, vol. 9, no. 12, pp. 10224–10243, Jun. 2022.

[122] A. M. Rahmani *et al.*, "Personal mental health navigator: Harnessing the power of data, personal models, and health cybernetics to promote psychological well-being," 2020, *arXiv:2012.09131*.

[123] P. Schmidt, A. Reiss, R. Duerichen, C. Marberger, and K. Van Laerhoven, "Introducing WESAD, a multimodal dataset for wearable stress and affect detection," in *Proc. 20th ACM Int. Conf. Multimodal Interact.*, 2018, pp. 400–408.

[124] "UC Irvine Machine Learning Repository." [Online]. Available: https://archive.ics.uci.edu/ml/index.php (Accessed: Oct. 15, 2021).

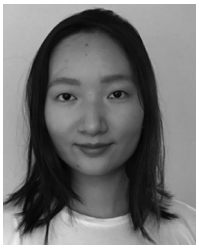
[125] C. Kirschbaum, K.-M. Pirke, and D. H. Hellhammer, "The 'trier social stress test'—A tool for investigating psychobiological stress responses in a laboratory setting," *Neuropsychobiology*, vol. 28, nos. 1–2, pp. 76–81, 1993.

[126] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Stat.*, 2017, pp. 1273–1282.

[127] "TestMy.net Internet Speed Test." [Online]. Available: https://testmy.net (Accessed: Oct. 15, 2021).

Farshad Firouzi is an Adjunct Assistant Professor with the Electrical and Computer Engineering Department, Duke University, Durham, NC, USA. He has authored more than 60 conference/journal papers. He is a top-producing expert and technical leader with more than ten years experience offering strong performance in all aspects of AI/ML, smart data, computer architecture, VLSI, and IoT, including research and development, consulting services, strategic planning, and technology solutions, across vertical industries, such as semiconductor, automotive, finance, manufacturing, logistics, and eHealth.

Dr. Firouzi has served as a Guest/Associate Editor for several well-known journals, such as the IEEE INTERNET OF THINGS JOURNAL, IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, *Future Generation Computer Systems* (Elsevier), *Microprocessors and Microsystems* (Elsevier), *Journal of Network and Computer Applications* (Elsevier), and *Information Systems* (Elsevier) as well as the chair for more than ten international conferences on AI/IoT/eHealth.



Shiyi Jiang received the B.A. degree (*magna cum laude*) from Boston University, Boston, MA, USA, in 2018, and the M.Sc. degree in electrical and computer engineering with data analytics and machine learning concentration from Duke University, Durham, NC, USA, in 2020, where she is currently pursuing the Ph.D. degree.

Her current research focuses on techniques to preserve privacy, enhance, and personalize healthcare data analysis in the IoT system.

Krishnendu Chakrabarty (Fellow, IEEE) received the B.Tech. degree from the Indian Institute of Technology Kharagpur, Kharagpur, India, in 1990, and the M.S.E. and Ph.D. degrees from the University of Michigan, Ann Arbor, MI, USA, in 1992 and 1995, respectively.

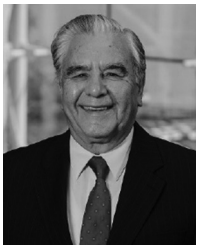
He is currently the John Cocke Distinguished Professor and the Department Chair of Electrical and Computer Engineering, and the Professor of Computer Science with Duke University, Durham, NC, USA. His current research projects include: design-for-testability of integrated circuits and systems, especially 3-D integration and system-on-chip; testing of AI accelerators; microfluidic biochips; hardware security; AI for failure prediction and healthcare; and neuromorphic computing systems.

Prof. Chakrabarty is a recipient of the National Science Foundation CAREER Award, the Office of Naval Research Young Investigator Award, the Humboldt Research Award from the Alexander von Humboldt Foundation, Germany, the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS Donald O. Pederson Best Paper Award in 2015, the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS Prize Paper Award in 2021, the *ACM Transactions on Design Automation of Electronic Systems* Best Paper Award in 2017, multiple IBM Faculty Awards and HP Labs Open Innovation Research Awards, and over a dozen best paper awards at major conferences. He is also a recipient of the IEEE Computer Society Technical Achievement Award in 2015, the IEEE Circuits and Systems Society Charles A. Desoer Technical Achievement Award in 2017, the IEEE Circuits and Systems Society Vitold Belevitch Award in 2021, the IEEE-HKN Asad M. Madni Outstanding Technical Achievement and Excellence Award in 2021, the Semiconductor Research Corporation Technical Excellence Award in 2018, and the IEEE Test Technology Technical Council Bob Madge Innovation Award in 2018. He is a Research Ambassador of the University of Bremen, Bremen, Germany and he was a Hans Fischer Senior Fellow with the Institute for Advanced Study, Technical University of Munich, Munich, Germany, from 2016 to 2019. He is a 2018 recipient of the Japan Society for the Promotion of Science Invitational Fellowship in the "Short Term S: Nobel Prize Level" category. He was a Distinguished Visitor of the IEEE Computer Society from 2005 to 2007 and from 2010 to 2012, a Distinguished Lecturer of the IEEE Circuits and Systems Society from 2006 to 2007 and from 2012 to 2013, and an ACM Distinguished Speaker from 2008 to 2016. He has served as the Editor-in-Chief of the *IEEE Design & Test of Computers* from 2010 to 2012, the *ACM Journal on Emerging Technologies in Computing Systems* from 2010 to 2015, and the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS from 2015 to 2018. He is a Fellow of ACM and AAAS, and a Golden Core Member of the IEEE Computer Society.

Bahar Farahani received the Ph.D. degree in computer engineering from the University of Tehran, Tehran, Iran, in 2017, and the Postdoctoral degree in computer engineering from Shahid Beheshti University, Tajrish, Iran, in 2019.

She is an Assistant Professor with Cyberspace Research Institute, Shahid Beheshti University. She has authored several peer-reviewed conference/journal papers as well as book chapters on IoT, big data, and AI.

Dr. Farahani has served as a Guest Editor for several journals, such as the IEEE INTERNET OF THINGS JOURNAL, IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, *Future Generation Computer Systems* (Elsevier), *Microprocessors and Microsystems* (Elsevier), *Journal of Network and Computer Applications* (Elsevier), and *Information Systems* (Elsevier). She has also served for the Technical Program Committee of many international conferences/workshops on AI/IoT/eHealth as well as the Technical Chair for the IEEE COINS Conference.



Mahmoud Daneshmand received the B.S. degree in mathematics and the first M.S. degree in mathematical statistics from the University of Tehran, Tehran, Iran, in 1963 and 1965, respectively, and the second M.S. and Ph.D. degrees in statistics from the University of California at Berkeley, Berkeley, CA, USA, in 1973 and 1975, respectively.

He is the Co-Founder and the Professor with the Department of Business Intelligence & Analytics as well as the Data Science Ph.D. Program, and the Professor with the Department of Computer Science, Stevens Institute of Technology, Hoboken, NJ, USA. He has more than 40 years of Industry and University experience as an Executive Director, an Assistant Chief Scientist, a Professor, a Researcher, a Distinguished Member of Technical Staff, a Technology Leader, a Founding Chair of Department, and the Dean of School with Bell Laboratories, Holmdel, NJ, USA; AT&T Shannon Labs Research, Florham Park, NJ, USA; University of California at Berkeley; University of Texas at Austin, Austin, TX, USA; New York University, New York, NY, USA; Sharif University of Technology, Tehran, Iran; University of Tehran; and Stevens Institute of Technology. He is a Data Scientist, and an expert in big data analytics, artificial intelligence, and machine learning with extensive industry experience, including with the Bell Laboratories as well as the AT&T Shannon Labs Research. He has published more than 400 journal and conference papers, authored/coauthored three books, and has graduated more than 3000 Ph.D. and M.S. students.

Prof. Daneshmand holds key leadership roles in the IEEE Journal Publications, including the IEEE INTERNET OF THINGS JOURNAL and IEEE TRANSACTIONS ON BIG DATA, IEEE major conferences, Industry–IEEE Partnership, IEEE Future Direction Initiatives, and IEEE AI/ML Initiative for Future Networks. He is the Co-Founder and the Vice President of the IEEE Technical Committee on Big Data. He has served as the general chair, keynote chair, panel chair, executive program chair, and technical program chair for many IEEE major conferences. He has given many keynote speeches in major IEEE as well as international conferences.

JaeSeung Song (Senior Member, IEEE) received the B.S. and M.S. degrees in computer science from Sogang University, Seoul, South Korea, in 2000 and 2002, respectively, and the Ph.D. degree from the Department of Computing, Imperial College London, London, U.K., in 2012.

He is an Associate Professor, leading the Software Engineering and Security Lab, with the Computer and Information Security Department, Sejong University. He worked with LG Electronics, Seoul, South Korea, as a Senior Researcher leading a 3GPP SA standard team, from 2002 to 2008. He worked as a Leading Standard Senior Researcher with NEC Europe Ltd., Heidelberg, Germany, from 2012 and 2013. His research interests span the areas of software engineering, software testing, and networked systems and security, with a focus on the design and engineering of reliable IoT/M2M platforms, particularly in the context of semantic IoT data interoperability, secure software patch techniques, blockchain IoT, and edge computing.

Dr. Song holds the position of the *IEEE Communications Standards Magazine* IoT Series Editor, the oneM2M Technical Plenary Vice-Chair, and the TTA IoT/M2M Convergence Special Project Group Chair.

Kunal Mankodiya (Member, IEEE) received the B.E. degree in biomedical engineering from Saurashtra University, Rajkot, India, in 2003, and the M.S. degree in biomedical engineering and the Ph.D. degree in computer science from the University of Lübeck, Lübeck, Germany, in 2007 and 2010, respectively.

He is currently a tenured Associate Professor of Biomedical Engineering and the Director of the Wearable Biosensing Lab, Department of Electrical, Computer, and Biomedical Engineering, The University of Rhode Island, Kingston, RI, USA. He was a Postdoctoral Researcher with Intel Science and Technology Center affiliated with Carnegie Mellon University, Pittsburgh, PA, USA, from 2011 to 2014. He co-founded EchoWear, a digital health startup producing solutions supporting neurodegenerative disorders, such as Parkinson's disease and dementia.

Dr. Mankodiya is a recipient of the TechConnect Defense Innovation Award in 2018, the NSF CAREER Award in 2017, the Innovator-of-the-Year Future Textiles Award, Germany, in 2017, the 40 under 40 Providence Business News Award in 2017, the NSF CRII Award in 2016, and the 2010 SYSTEX Award, Belgium, in 2010. He is a member of ACM and Biomedical Engineering Society and serves for the professional society in various capacities.