IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2023.0322000

Exploring the Synergy of Fog Computing, Blockchain, and Federated Learning for IoT Applications: A Systematic Literature Review

WILSON VALDEZ SOLIS[1], JUAN MARCELO PARRA-ULLAURI[2], ATTILA KERTESZ[1]
[1]University of Szeged, Dugonics tér 13, 6720, Szeged, Hungary (e-mail: wilson@inf.u-szeged.hu, keratt@inf.u-szeged.hu)
[2]Smart Internet Lab, University of Bristol, BS8 1QU Bristol, UK (e-mail: jm.parraullauri@bristol.ac.uk)

Corresponding author: Wilson Valdez Solis (e-mail: wilson@inf.u-szeged.hu).

"The research leading to these results has received funding from the national project TKP2021-NVA-09, supported by the Ministry of Culture and Innovation of Hungary through the National Research, Development and Innovation Fund, under the TKP2021-NVA scheme, and by CERCIRAS COST Action CA19135 funded by the COST Association, along with the University of Szeged Open Access Fund (Grant No. 6858). Additionally, partial sponsorship was provided by the UK GOV DSIT (FONRC) project REASON"

ABSTRACT

The proliferation of Internet of Things (IoT) applications poses formidable challenges in managing data processing, privacy, and security. In response, technologies such as Fog Computing (FC), Blockchain (BC), and Federated Learning (FL) have emerged as promising solutions. Combining these technologies can broaden their scope, and impose novel challenges. This paper conducts a Systematic Literature Review (SLR) to investigate their integration within the IoT domain, systematically evaluating the current state-of-the-art by analyzing 40 papers against 38 extraction criteria, encompassing technical characteristics specific to FC, BC, FL, or their integration. The findings offer insights into the advantages, challenges, opportunities, and limitations of this integration, addressing data processing, privacy, and security concerns in IoT. By filling a research gap and directly examining FC, BC, and FL interoperability across architectural layers, this study contributes to knowledge expansion in the field. This paper proposes a novel framework for implementing FL and BC within FC environments for IoT applications, alongside a comprehensive synthesis of existing literature, distinguishing it from previous research efforts. Furthermore, it offers valuable insights into the current landscape, identifies research needs, and proposes future research directions. The framework and literature synthesis provided allow readers to access customized information on FC-BC-FL integration, aiding in designing and implementing robust IoT solutions.

INDEX TERMS Blockchain, Edge Computing, Federated Learning, Fog Computing, Internet of Things, Systematic Literature Review.

I. INTRODUCTION

THE Internet of Things (IoT) serves as a technological paradigm that supports various application domains (e.g., industry, smart cities, healthcare) through the global telecommunication infrastructure and Cloud Computing (CC) services [1]–[3]. The rapid growth of IoT has led to a significant surge in connected devices, revealing vulnerabilities such as high bandwidth usage, efficiency, latency, security, privacy, and data heterogeneity [2], [3]. Efforts in the distributed systems domain have aimed to enhance both IoT and Cloud paradigms, addressing their flaws. Additionally, novel solutions like Fog Computing (FC), Blockchain (BC), and Federated Learning (FL) have emerged to complement these paradigms [4]–[7], and are the focus of this document.

FC extends CC services closer to devices, thereby reducing latency and alleviating the cloud's workload by minimizing the transmitted data volume [4], [5], [8]. In contrast, BC enhances the security and precision of information management using a distributed ledger technology that records transactions (i.e., data) and employs distributed consensus protocols to govern these repositories [6], [9]. Lastly, FL, an emerging field in Artificial Intelligence (AI), tackles data decentralization by utilizing a distributed computing system to locally train Machine Learning (ML) models on network devices. This approach reduces data processing, bandwidth usage, and reliance on cloud services [7], [10].

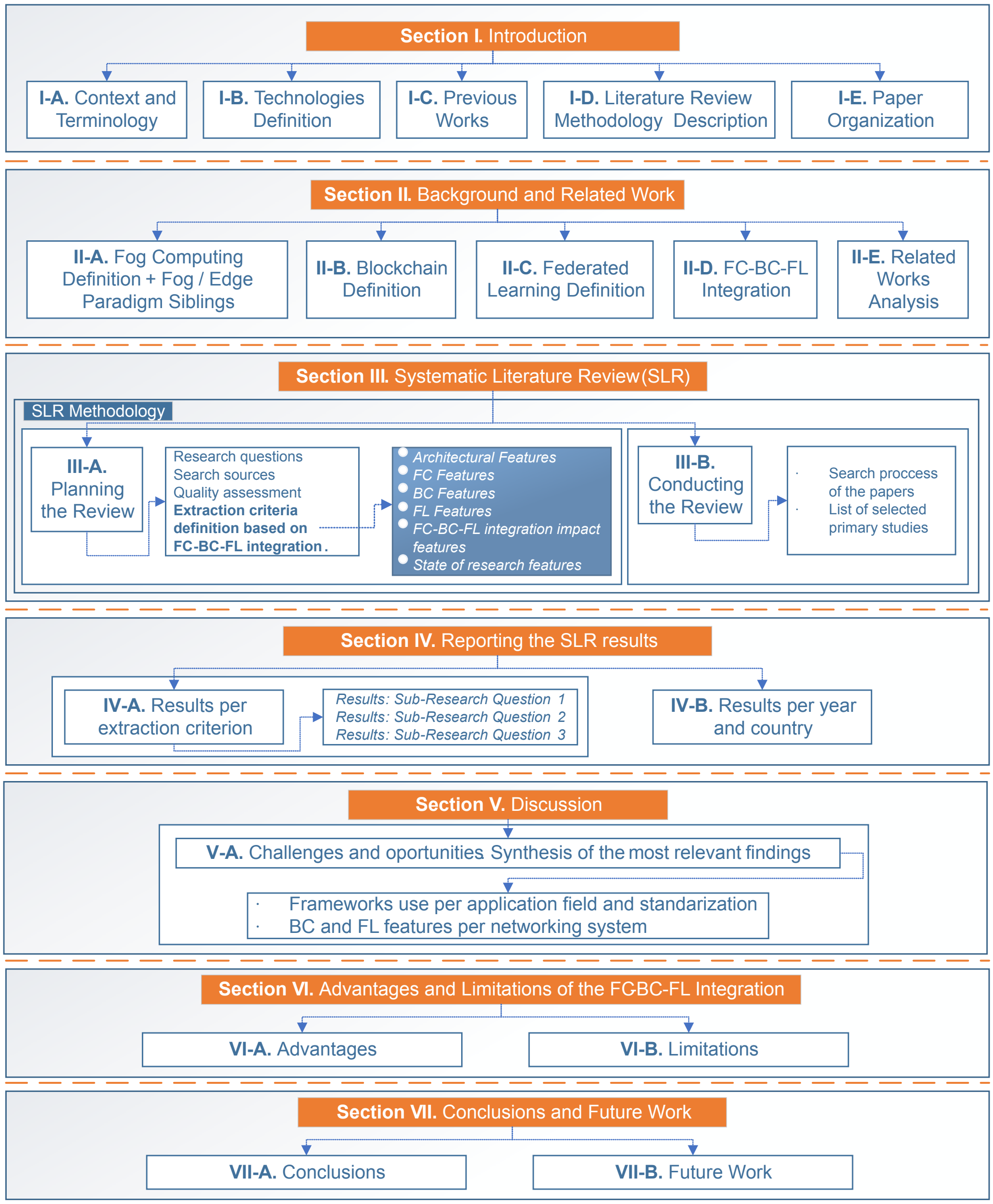

Section I. Introduction
I-A. Context and Terminology
I-B. Technologies Definition
I-C. Previous Works
I-D. Literature Review Methodology Description
I-E. Paper Organization
Section II. Background and Related Work
II-A. Fog Computing Definition + Fog / Edge Paradigm Siblings
II-B. Blockchain Definition
II-C. Federated Learning Definition
II-D. FC-BC-FL Integration
II-E. Related Works Analysis
Section III. Systematic Literature Review(SLR)
SLR Methodology
III-A. Planning the Review
Research questions
Search sources
Quality assessment
Extraction criteria definition based on FC-BC-FL integration.
Architectural Features
FC Features
BC Features
FL Features
FC-BC-FL integration impact features
State of research features
III-B. Conducting the Review
Search proccess of the papers
List of selected primary studies
Section IV. Reporting the SLR results
IV-A. Results per extraction criterion
Results: Sub-Research Question 1
Results: Sub-Research Question 2
Results: Sub-Research Question 3
IV-B. Results per year and country
Section V. Discussion
V-A. Challenges and oportunities Synthesis of the most relevant findings
Frameworks use per application field and standarization
BC and FL features per networking system
Section VI. Advantages and Limitations of the FC-BC-FL Integration
VI-A. Advantages
VI-B. Limitations
Section VII. Conclusions and Future Work
VII-A. Conclusions
VII-B. Future Work

FIGURE 1. Structure of the paper

Numerous secondary studies (or literature reviews) have recognized that despite the collaborative advancement of these technologies within IoT solutions, persistent challenges endure. Yet, these reviews tend to focus primarily on pairs of technologies: BC and FC [11]–[15], FC and FL [16]–[18], or BC and FL [16], [19]–[26]. While primary studies are emerging to explore the direct relationship among all three technologies, there is a noticeable lack of secondary studies summarizing key findings and issues across FC, BC, and FL, highlighting a substantial research gap.

Hence, to comprehensively depict the interplay among these technologies and systematically capture insights within the domain, a Systematic Literature Review (SLR) emerges as an ideal secondary study approach. SLRs enable unbiased identification, evaluation, and interpretation of research queries within a specific domain [27].

This paper conducts an SLR exploring BC, FC, and FL integration in IoT applications. Following Kitchenham & Charters' guidelines in [27], [28], it emphasizes attributes like credibility, contribution, transferability, and compliance [29]. From 2016, when 'Federated Learning' was coined by Google [7], to August 2023, a systematic search for 40 papers was conducted. Additionally, 38 analysis criteria were established to evaluate these papers. These criteria were developed based on an initial assessment of 16 relevant related reviews. Findings reveal a lack of collaborative research in architecture, frameworks, heterogeneity, and standardization among these technologies, offering insights for future research.

As a result of the study, this paper introduces systematical categorization and classification of criteria for the FC-BC-FL integration for IoT applications. It provides a structured framework for understanding the synergies and interactions among these three technologies, aiding researchers and practitioners in navigating the complexities of their integration and implementation. The systematic categorization or criteria is organized into distinct dimensions, including architectural approaches, methodological strategies, and application domains, offering a comprehensive overview of how these technologies can be combined to tackle IoT challenges.

The structure of this paper is outlined in Figure 1. Section II explores the background of the implied technologies, including Fog/Edge Computing, Blockchain, and Federated Learning, with an analysis of their integration. Additionally, it compares similar survey/review papers to highlight the contribution of this SLR. Section III adapts Kitchenham's SLR guidelines to the context. It elaborates on the core of the SLR, defining the research questions and the extraction criteria under which the evaluation will be performed. This section provides a detailed categorization of the technologies and their integration. Additionally, it describes the process of selecting papers for analysis in the review. Section IV validates the SLR methodology by presenting the results of the analyzed studies, classified under the extraction criteria. It also presents results by year and country. A synthesis of the most relevant studies, along with the challenges and opportunities, forms part of the Section V. Section VI discusses the advantages and limitations of the FC-BC-FL integration based on the SLR results. Finally, Section VII presents the conclusions and outlines avenues for future research.

II. BACKGROUND AND RELATED WORK

This section offers an exhaustive background encompassing fundamental concepts, architectures, and essential technical aspects within the domains of Fog Computing, Blockchain, and Federated Learning technologies. It further examines and contrasts relevant literature reviews concerning these technologies, with a specific emphasis on the interconnections between BC and FC, FC and FL, and BC and FL. This analysis provides insight into the distinctive contributions of our work in comparison to prior research endeavors.

A. FOG COMPUTING

Fog Computing, an architectural concept introduced by Cisco in 2012 [4], represents a paradigm shift that redefines conventional computing structures [5], [30]. This concept involves the decentralization of the traditional Cloud and the extension of its services (i.e., storage, processing, networking) to the network edge. The main goal is to enhance the scalability and performance of applications by distributing the computational load away from centralized clouds [4], [5], [8], [30].

At the core of this architecture lie the Fog Nodes (FNs), which establish connections with an array of counterparts, such as other nodes, end devices, centralized services, and even the cloud. Through these connections, FNs extend computing services and resources in closer proximity to end devices, creating an intricate distributed computing environment [7]. Furthermore, FC addresses common issues encountered in Cloud Computing, including network bandwidth overuse, latency, request-response time reduction, and more [4], [5], [30]. This architectural framework has found a particularly fitting application in the realm of IoT, offering an exceptional fusion of computing, networking, and storage capabilities across a wide array of geographically dispersed devices [4], [5], [8], [30].

1) Fog Computing/Edge Computing, Paradigm Siblings

As we delve into the realm of FC, it becomes imperative to acknowledge the closely related concept of Edge Computing (EC). Similar to FC, EC emphasizes the proximity of computation and storage to data sources. However, these paradigms display distinct characteristics and functionalities [31], [32].

Edge Computing is intrinsically concerned with localized processing, often occurring at the immediate first hop from IoT devices, encompassing smart sensors, smart vehicles, and WiFi access points. It empowers computation, data processing, decision-making, and privacy protection within the confines of edge devices. While EC excels at optimizing local network interactions and reducing latency, its scalability might be constrained by limited resources and potential resource contention among multiple IoT applications [31].

In contrast, FC casts a broader net by extending the edge concept to a hierarchical architecture. This encompasses a diverse range of network edge devices such as RANs, base stations, and edge routers. By integrating cloud-like capabilities into the network edge, FC offers a comprehensive suite of computing, networking, storage, control, and acceleration services spanning from cloud to IoT devices [31], [32]. The architecture envisions a seamless platform that caters to various industries and application domains, fostering interactions between edge devices and providing a holistic infrastructure-level perspective [32]. Overall, while EC emphasizes localized processing at the immediate network edge, FC adopts a more comprehensive approach. It encompasses a hierarchical architecture and a broader array of devices and services, with a focus on seamless integration and the utilization of cloud resources to address scalability and resource contention challenges that EC might encounter.

Figure 2 depicts the FC architecture, which comprises four layers (with EC considered as an integral part of the infrastructure), each serving a distinct role within the data processing hierarchy. The architecture is characterized by:

1) *Devices / Edge Layer:* This base layer comprises physical devices (e.g., sensors, actuators, IoT devices) that generate data by gathering it from the environment. Positioned closest to these devices is the Edge infrastructure, acting as an intermediary connecting them

to upper layers for processing. It conducts initial data preprocessing before relaying it to higher layers such as Fog or Cloud, thus minimizing latency by processing essential data close to its source [5], [8], [12], [31].

2) *Fog Layer:* It serves as an intermediary between the Devices/Edge and Cloud layers, comprising FNs/servers to bolster processing and storage capacities. The Fog layer proves invaluable for applications needing greater computational resources than the Edge layer offers but not requiring the extensive resources of the Cloud. It facilitates real-time analytics, decision-making, and data processing while ensuring low latency [4], [5], [8].

3) *Cloud Layer:* This layer represents the traditional CC infrastructure. It provides vast infrastructural (Infrastructure as a Service (IaaS)), computational, and storage (Software as a Service (SaaS)) resources that can be accessed remotely. While the Cloud layer excels in heavy data analysis, storage, and long-term processing, it might introduce higher latency due to data transmission to and from remote data centers [4], [5], [8], [12].

4) *Application Layer:* The uppermost layer (situated within the Cloud Layer) is designated for the development and deployment of applications and services (Software as a Service (SaaS)). This encompasses user interfaces, web services, data analytics, AI services, and other software components that make use of the processed data. Applications can interact with data across all lower layers, making it possible to harness the benefits of the entire architecture [5], [8], [31].

Taking into account the aforementioned information, the analysis of subsequent stages in this SLR will encompass both FC and pertinent EC studies.

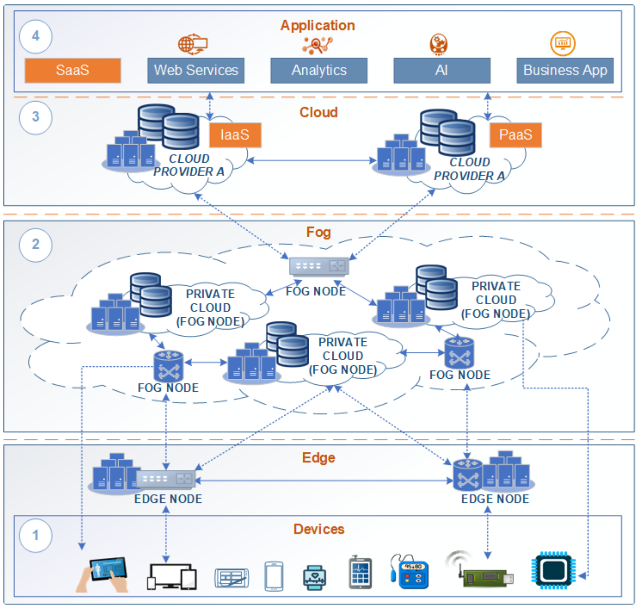


FIGURE 2. A Fog Computing based architecture

B. BLOCKCHAIN

Blockchain, introduced by Satoshi Nakamoto in 2008 is a decentralized technology that forms the essential framework for establishing trustworthy digital currency systems [6].

BC systems encompass distributed computing structures that primarily store and process a Distributed Ledger (DL) within Peer-to-Peer (P2P) networks [9], [33], [34]. Both permissioned and permissionless variants of BC exist, offering authorized or open participation in the system, respectively. The data archived within Blockchain DLs can be public or private, depending on the domain/profile of users capable of requesting or executing tasks within the system [33].

Miners represent the core of the BC ecosystem, undertaking the resolution of computational puzzles or problems (whose complexity is conditionally predefined) to validate blocks and integrate them into the chain. Various consensus algorithms, including proof-based mechanisms (e.g., Proof-of-Work (PoW), Proof-of-Stake (PoS), Proof-of-Authority (PoA)), as well as non-proof-based strategies (e.g., Byzantine Fault Tolerance (BFT)), are employed to achieve consensus on transaction validity [33]. Forking arises when different valid DL versions emerge, each consensus algorithm having a distinct protocol for handling forks, while hashing prior blocks maintains BC's immutability [9], [34].

The potential application of BC in decentralized cloud servers at the network edge has garnered significant attention, leading to the proposition of models like Blockchain-as-a-Service (BaaS) to revolutionize CC across various domains. These models aim to elevate the capabilities of CC by seamlessly integrating BC technology, offering a versatile array of functionalities (i.e., e-voting, authentication, and identity management to trading, reputation management, supply chain management, data management) [9], [33]–[35].

Although BC is originated in cryptocurrencies like Bitcoin [6], its impact extends beyond finance. It enables Smart Contract (SC) creation (e.g., Ethereum) and fosters innovation in decentralized systems, evolving digital interactions and trust [9], [34], [35].

BC-based systems are structured across five layers (see Figure 3) comprising hardware/infrastructure, data, network, consensus (including incentives), and application [9], [33], [34]. These layers are described below:

1) *Hardware / Infrastructure Layer:* This architecture's foundation layer encompasses tangible components that form the backbone of the BC network. Devices (including nodes and miners), establish the network's framework. virtual machines (VMs) and containers create the computational environment for code execution. The communication infrastructure ensures seamless interaction, while messaging mechanisms facilitate data exchange and synchronization [9], [33].

2) *Data Layer:* It manages the foundational building blocks of BC data. Transactions serve as the bedrock of the data structure, with Digital Signatures ensuring transaction integrity and authenticity. The Merkle Tree structure optimizes data verification, while the Data

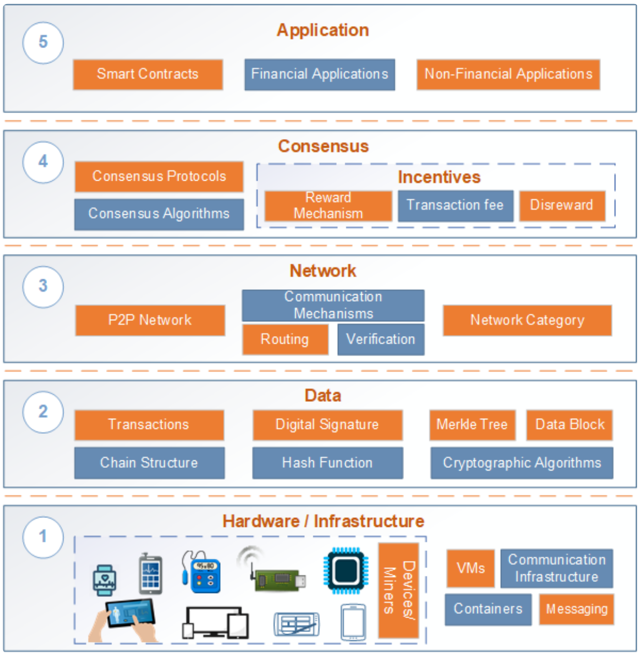


FIGURE 3. An overview of Blockchain propoerties

Block and Chain Structure establish the chronological sequence of transactions. Hash Functions and Cryptographic Algorithms play a pivotal role in data security and encryption within this layer [9], [33].

3) *Network Layer:* It encompasses the communication and connectivity infrastructure that enables node interactions. It includes a P2P Network for direct node-to-node communication, Communication Mechanisms for data exchange, Routing to enhance data propagation efficiency, and Verification mechanisms to authenticate transmitted data [9], [33].
4) *Consensus Layer:* This layer assumes a pivotal role in ensuring network-wide agreement and validation. Consensus Protocols and Algorithms establish rules and mechanisms for collective transaction validation [9], [34]. Moreover, within this layer, the incentives sublayer orchestrates rewards for participants contributing to the consensus process, thereby aligning economic incentives with network integrity [9].
5) *Application Layer:* Positioned as the topmost layer, it accommodates a wide range of both financial and non-financial applications that harness the capabilities of BC technology. Within this stratum, one can find Smart Contracts, Financial Applications, and Non-Financial Applications, representing the diverse spectrum of use cases that BC is capable of addressing [9], [33], [35].

This comprehensive architectural framework underscores the intricate interplay of components and mechanisms, collectively realizing the robust capabilities and security inherent to BC technology, aspects that should be taken into consideration during the development of the SLR.

C. FEDERATED LEARNING

Federated Learning (FL) is a paradigm introduced by Google in 2016 within the ML and data privacy domain [7]. In traditional ML approaches, data is often centralized on a single server for training, giving rise to concerns regarding data security and privacy breaches. Unlike, FL adopts a distributed approach, enabling collaborative model training across a network of decentralized devices or servers without sharing raw data. This decentralized feature not only effectively addresses privacy concerns but also capitalizes on the collective intelligence derived from diverse data sources [7], [10], [17].

At its core, FL offers distinctive technical features that differentiate it from traditional centralized methods. Operating within a decentralized framework, FL safeguards data privacy by retaining information locally on devices, selectively sharing only model updates for collaborative training [10], [17]. This unique approach facilitates a privacy-centric aggregation process, enhances communication efficiency, and seamlessly adapts to the intricacies of diverse data sources. These attributes make it suited for scenarios involving Fog and Edge [16]–[18]. Furthermore, FL extends personalized and fault-tolerant model training capabilities while complying with stringent data protection regulations. This multi-faceted nature positions FL as a robust and privacy-conscious solution with expansive potential across various domains [17].

In practical terms, FL empowers local data training while upholding stringent data privacy. Devices or servers involved in this process autonomously compute model updates using their respective datasets, which are subsequently aggregated to refine the overarching model. This decentralized approach significantly mitigates the risks associated with centralized data storage and transmission, offering advantages that extend beyond privacy. Notably, FL delivers scalability gains, reduces communication overhead, and proves adaptable to resource-constrained real-world scenarios [10], [17].

FL architecture consists of four layers (see Figure 4), each serving a specific role in the decentralized learning process:

1) *Client / Infrastructure Layer:* This layer encompasses both devices and local servers that play an active role in the FL process. Devices contribute their localized data and learning models to the broader system. Within this layer, a sub-layer, referred to as the Data Layer, assumes responsibility for Local Data Management and Local Learning Models. The task of Local Data Management involves overseeing proper data handling and secure storage on individual devices. Meanwhile, the Local Learning Models, which are trained using local data, undergo continuous refinement through updates throughout the FL process [10], [17], [36].
2) *Network Layer:* This layer manages interactions among devices, nodes, local servers, and aggregation servers within the FL system. It uses advanced Channel Coding techniques to robust data transmission over potentially noisy channels, supported by various Communication Protocols (e.g., 6G, 5G, Z-Wave, ZigBee, Wi-Fi) to facilitate data exchange [17], [36], [37].

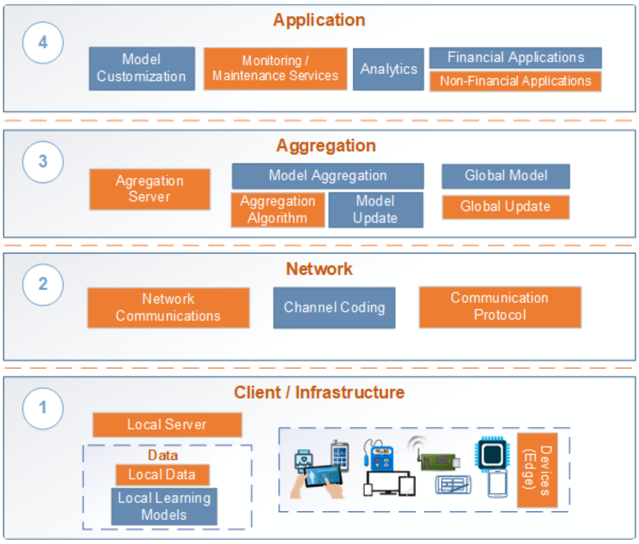


FIGURE 4. A general architecture for Federated Learning

3) *Aggregation Layer:* Situated at the heart of the architecture, This layer assumes a pivotal role in amalgamating local models into a comprehensive global model. The pivotal Aggregation Server oversees the Model Aggregation process. Various Aggregation Algorithms dictate how local model updates integrate into the global model. Model Updates involve transmitting refined models to the Aggregation Server, which then generates the comprehensive Global Model. Regular Global Updates drive the continual refinement of the overarching model [10], [17], [36], [37].
4) *Application Layer:* Positioned as the uppermost layer, this layer customizes the global model to align with specific applications through Model Customization. It also encompasses vital services such as Monitoring and Maintenance, which sustain the optimal performance and integrity of the FL system. Analytics extract valuable insights and patterns from the global model, while diverse applications spanning financial and non-financial domains effectively translate the model's findings into real-world contexts [10], [17].

Therefore, this comprehensive FL Architecture seamlessly integrates layers, fostering a unified, efficient system prioritizing data privacy, operational efficiency, and practical applicability, all considered during SLR development.

D. INTEGRATING FOG COMPUTING, BLOCKCHAIN, AND FEDERATED LEARNING

Building on the theoretical principles discussed earlier, integrating FC, BC, and FL appears highly feasible due to their shared decentralized computing foundations. Their interconnections highlight compatibility in both theory and practice, urging a deeper exploration.

For instance, the FC architecture creates a dynamic framework for distributed and federated computing. Clustering nodes systematically facilitates collaborative interactions, fostering an environment for interconnecting federated services across domains. This seamless interconnectivity not only enhances scalability but also aligns intriguingly with FL's core principles [8]. This alignment gains significance, emphasizing the shared focus on optimizing service distribution for superior application performance.

The potential for synergy becomes more evident when delving into the realm of Blockchain. BC's decentralized nature and focus on secure and transparent data transactions complement the ideals of both FC and FL. The distributed and tamper-resistant nature of BC inherently supports the trust and privacy concerns essential to FL's data aggregation process. This synchronization of objectives between BC and FL creates a foundation for secure, privacy-conscious collaborative learning scenarios within an FC environment.

In essence, the FC-BC-FL integration paints a cohesive picture of decentralized, collaborative, and privacy-conscious data processing and sharing. Despite inherent distinctions, these technologies synergistically tackle challenges, foster innovation, and shape future computing systems across domains while prioritizing security, privacy, and efficiency.

As a first look, Figure 5 visually illustrates the integration and interaction of FC-BC-FL technologies to collaboratively address common challenges by converging within each layer, showcasing their synergistic roles and interactions. This view employs a schema with three core layers: Device/Infrastructure, Network, and Application, acting as a visual guide that emphasizes their harmonious coexistence and collaboration. By aligning layers in this schema, Figure 5 offers insights into the interconnected nature of FC (See Figure 2), BC (See Figure 3), and FL (See Figure 4) architectures.

1) *Application Layer:* In this layer, FC, BC, and FL exhibit distinct functionalities. FC's Layer 3 includes the Cloud Layer with ample computational resources and the Application Layer housing diverse services. BC's Layer 4, the Consensus Layer, centers on network-wide agreement, while Layer 5 serves various applications. FL, within Layer 4, tailors the global model and provides monitoring and maintenance services, while Layer 3 manages aggregation. This layer demonstrates the synergy of FC-BC-FL, offering diverse applications.

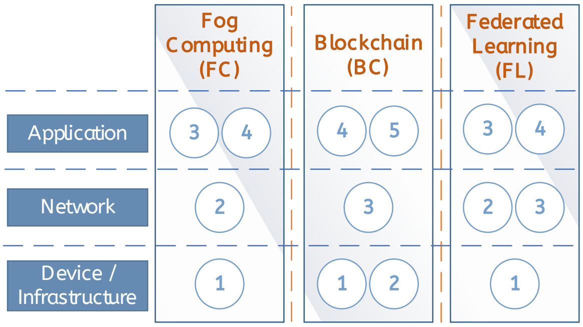


FIGURE 5. FC-BC-FL architectural integration relationship

2) *Network Layer:* This Layer facilitates smooth connectivity and communication. FC's Fog Layer serves as a bridge between Edge and Cloud, while BC's layer manages node interactions. FL's layer orchestrates device-to-server communication, using advanced coding techniques and protocols to ensure efficient data exchange and reliable transmission.
3) *Device/Infrastructure Layer:* At the base, FC's Layer 1 consists of devices and sensors gathering data, with the Edge infrastructure nearby for initial processing. BC's Layers 1 and 2 establish the hardware foundation and manage data using transactions, signatures, and cryptographic elements. FL's Layer 1 includes devices and local servers, handling the data and learning models.

E. RELATED WORK

Over the last five years, numerous primary studies explored integrating these technologies (FC/EC, BC, and FL). These efforts curated influential studies for literature reviews, symbolizing collaborative research progress. This subsection analyzes and compares sixteen of these secondary studies to identify areas needing further exploration in this field.

As Table 1 shows, prevailing studies have predominantly focused on examining combinations of two out of the three technologies: BC-FC [11]–[15], FC-FL [16]–[18], or BC-FL [16], [19]–[26]. A noticeable trend emerges wherein there is a scarcity of literature reviews encompassing the integration of all three technologies comprehensively. The comparison presented in Table 1 highlights the strengths and weaknesses of these studies, specifically in areas such as architectural features, frameworks, integration aspects (privacy, efficiency, performance, security, interoperability, scalability, data management, service levels, trust, heterogeneity, resilience, and access control), as well as the analysis of experimentation features or general data analysis of the papers - detailed further in the subsequent section. These papers excel in their comprehensive analyses of specific intersections among the technologies, elucidating architectures, addressing security concerns, and exploring various application domains. However, their limitations become apparent in the absence of a holistic integration across all three paradigms in several studies. Some prioritize or overlook one of the three technologies, limiting a comprehensive understanding of their collective potential within specific criteria or interest areas. Moreover, while these studies address challenges and solutions within individual frameworks, there is a recurring oversight in considering architectural features, thus failing to capture the entirety of the tripartite integration. Consequently, while these studies offer valuable insights into isolated intersections, they lack a cohesive examination of the FC-BC-FL holistic integration.

Therefore, our review distinguishes itself by not only acknowledging the existing research gaps but also by proactively addressing them through a meticulous examination of the integration of FC/EC-BC-FL. Unlike prior studies which predominantly focus on partial combinations of these technologies, our review takes a pioneering step forward by providing a comprehensive analysis that encompasses all three paradigms. We meticulously explore various criteria, ranging from architectural features to security concerns, ensuring a holistic understanding of the integration's potential. By bridging this crucial gap in the literature, our review offers unparalleled insights that are indispensable for advancing research and practical applications in this field.

TABLE 1. Related Literature Reviews about integrating FC-BC, FC-FL, BC-FL (●: fully addressed; ○: partially addressed)

Paper	Year	Dimension	Architectural Features	Frameworks	FC Features	EC Features	BC Features	FL Features	Privacy	Efficiency	Performance	Security	Interoperability	Scalability	Data management	Service Levels	Trust	Heterogeneity	Resilience	Access Control	Experimental Features	Analysis of the studies
[11]	2022	FC-BC	○	○	●	○	●		●		●	●		●	●		●			●		
[12]	2023	FC/EC-BC	○		●	●	●		○		●	○		○	○			○			○	●
[13]	2019	FC/EC-BC	●	○	○	●	●		○	○	●	●		●	○			○				
[14]	2020	FC-BC	○		●		●		○	○		○			○		○					○
[15]	2019	FC-BC	●	●	○	●	●		●		○	●	○	●	●		●	●	○	●		
[16]	2021	FC-FL/BC-FL	●	●	○	●	●	●	●	○	○	●		○	○	○	○	●				
[17]	2021	FC-FL		●	○	●	●	●	●			●		●				○				
[18]	2020	FC-FL	○	●		●		●	●	○	○	●		○				●				
[19]	2021	BC-FL	○				●	●	●			●	○									
[20]	2022	BC-FL	○	○			●	●	●		●	●		●	●			○				
[21]	2021	BC-FL	●	○			●	●	●			●										
[22]	2021	BC-FL	●	●			●	●	○	●	●	●										○
[23]	2022	BC-FL	●				●	●	●	○	○	●		○			○	●				
[24]	2021	BC-FL	●	●			●	●	●	●	●	●			○		○	○				●
[25]	2022	BC-FL	○	○			●	●	●	○		●					○					
[26]	2024	BC-FL	○	○		○	●	●	●	○		●	○		●		○		●			
Our Review		FC/EC-BC-FL	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

III. SYSTEMATIC LITERATURE REVIEW

This section outlines the Systematic Literature Review (SLR) methodology, covering the research approach, review protocol formulation following Kitchenham's guidelines, study selection, and categorization criteria.

Kitchenham's systematic approach ensures structured, reliable, and repeatable acquisition, evaluation, and interpretation of information [27]. This tailored procedure investigates Federated Learning and Blockchain convergence within Fog Computing for IoT applications.

Figure 6 illustrates the three main phases of Kitchenham's guide: Planning, Conducting, and Reporting. Here, we focus on the initial two phases, with the final phase detailed in the subsequent section.

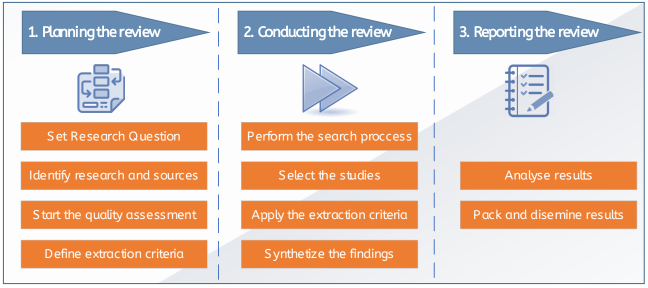

1. Planning the review
Set Research Question
Identify research and sources
Start the quality assessment
Define extraction criteria
2. Conducting the review
Perform the search proccess
Select the studies
Apply the extraction criteria
Synthetize the findings
3. Reporting the review
Analyse results
Pack and disemine results

FIGURE 6. Kitchenham phases for performing SLR

A. PLANNING THE REVIEW

The planning stage structures the SLR by initiating from the definition of the Research Questions, which in turn forms the basis for the SLR protocol statement [27].

1) Research Question Definition

Since the SLR aims to analyze aspects related to the integration of BC and FL in FC to enhance IoT applications, the main Research Question (RQ) and three Sub-Research Questions (SRQs) have been formulated to contribute to specifying the findings of this study (see Table 2).

SRQ1 seeks to comprehend the intersection of FC, FL, and BC technologies in architectures, frameworks, application scenarios, and use cases. It explores leveraging BC and FL techniques in FC to enhance IoT applications, encompassing aspects such as security, privacy, efficiency, performance, interoperability, and data management.

TABLE 2. Research Question and Sub-Research Questions

RQ	*How is the integration of Federated Learning (FL) and Blockchains (BC) in Fog Computing (FC) utilized to support Internet of Things (IoT) applications, and the associated trends, challenges, and opportunities?*
SRQ1	*What are the key motivations, relationships, research approaches, and technical dimensions behind integrating FL, BC, and FC to support IoT applications?*
SRQ2	*How can the integration of BC and FL techniques in FC support and improve IoT applications (in terms of security, privacy, efficiency, performance, interoperability, and data management)?*
SRQ3	*What is the current state of research in the integration of FC, BC, and FL for supporting IoT applications, including trends, methodologies, evaluation approaches, key findings, and overall research status?*

SRQ2 comprehensively analyzes the integration of BC-FL within FC for IoT applications, evaluating performance, efficiency, and security impact while exploring FC interoperability. It examines how FL and BC enhance FC security and efficiency, including BC's role in ensuring privacy. The study explores privacy and security considerations resulting from FC-FL-BC integration, particularly with Smart Contracts. Investigating BC-FL synergy for enhanced data management and security in FC for IoT, it also identifies opportunities to improve FC-BC-FL integration for diverse IoT applications.

SRQ3 encompasses a comprehensive exploration of the following aspects: the type of research conducted, the present state of research including emerging trends, existing drawbacks and limitations, the benefits derived from integration, identified research gaps and challenges, methodologies employed, evaluations of the integration of these three technologies, and potential opportunities for further advancement.

2) Identify Research and Sources

Kitchenham [28] advises extracting information considering population, intervention, comparison, outcomes, and context, as outlined in Table 3.

TABLE 3. Extraction aspects

Aspects	Description
Population	Studies that involve the FC-BC-FL technologies synergy.
Intervention	The study includes a set of technological aspects about integrating FC-BC-FL technologies, that drive the research in this area.
Comparison	The study aims to compare the different technological aspects addressed when integrating FC-BC-FL technologies.
Outcomes	To identify the main aspects addressed in the studies that integrate FC-BC-FL technologies.
Context	This study is performed in a research context, where the experts in the domain present primary studies.

The search process encompasses two stages: Automatic and Manual. In the initial phase, the Advanced Search tools within major digital libraries such as IEEE Xplore, ACM, Springer Link, and Science Direct are utilized to identify relevant papers. Simultaneously, pertinent Conferences and Journals linked to the research domain are explored for a Manual Search on the specified topic. The search strategy begins by outlining key terms as the search string shows: *"[federated learning AND blockchain AND fog computing]"*. It is employed across metadata (i.e., title, abstract, keywords) for articles from all sources, with syntax adjusted according to each library. Variations in terminology, such as "Collaborative Learning" for FL and "Distributed Ledger" for BC, are also taken into account. It is important to note that "Edge Computing," due to its resemblance to Fog Computing, is also incorporated. The search is specifically focused on studies from 2016 onwards, as the Federated Learning emergence [7] is a significant milestone among these technologies.

3) Quality Assessment

Following the initial selection phase, retrieved studies from both automated and manual searches undergo evaluation based on title, abstract, and keywords for inclusion consideration. Discrepancies in the selection are resolved via consensus after thorough paper examination. Studies meeting inclusion criteria, such as presenting novel insights on FL-BC integration in FC, and adhering to English language and at least 5 pages length criteria, are included. Conversely, studies falling under exclusion criteria, like introductory, short, and non-English papers, are excluded. Besides, the quality assessment of the primary studies utilizes a three-point Likert-scale questionnaire encompassing subjective (issues and solutions related to FC-BC-FL integration) and objective (relevance of publication and citation frequency) questions.

Responses to subjective questions vary from +1 "agreement," 0 "partial agreement," to -1 "disagreement." For objective questions about study quality, responses range from +1 "very relevant," 0 "relevant," to -1 "not so relevant." The relevance question considers library ranking and conference tier. Regarding citation frequency, Journal or Google Scholar citation reports assess the study's impact, with responses ranging from +1 "cited by more than five authors," 0 "partially," to -1 "not been cited." Notably, recent publications receive a "partially" score to avoid undue penalties.

4) Extraction Criteria

Extraction criteria are essential for systematically gathering relevant information during the research process. Defining these criteria helps to deepen the understanding of each technology and their initial relationships. In this study, these criteria are established based on information obtained from the studies discussed in the related works subsection and additional existing taxonomies of the implied technologies. This approach ensures that the collected data and the established extraction criteria are significant and align with the overarching research objectives, contributing to a comprehensive analysis of the interactions between the investigated technologies. By employing well-defined extraction criteria, the research process effectively captures and synthesizes essential insights from various sources, thereby enhancing the overall rigor and reliability of the research findings [27], [28].

In the following lines, there are defined 38 extraction criteria that cover a wide range of concepts, addressing each of the three sub-research questions as previously outlined. EC1 to EC18 are utilized to tackle SRQ1 (see Figure 7, 8, and 9), from EC19 to EC33 the SRQ2 (see Figure 11), and from EC34 to EC38 for SRQ3 (see Figure 12).

EC1: Main scope of the study. Indicates the specific related areas covered within the reviewed study: Fog/Edge Computing, Blockchain, and Federated Learning (see Figure 7).

The following criteria (EC2-EC4) delineate characteristics that might be components of architectures as well as the fields of application integrating FC-BC-FL.

EC2: Architecture Features. Encompasses several sub-criteria for analyzing the architectural features (see Figure 7).

1) *Software Architecture Pattern Type.* This sub-criteria defines the architecture patterns that shape the fundamental characteristics of an application [38]. Figure 7 shows architecture patterns that could be employed.
   a) *Layered.* This approach utilizes horizontal layers, each serving distinct functions to foster modular and organized development [38].
   b) *Event Driven.* In this pattern, event processing components are decoupled, managing specific events. It includes two topologies: mediator, which coordinates multiple event steps using a central mediator, and broker, connecting events in a chain without a central mediator [38].
   c) *Microkernel.* It features a core system and plug-in modules for extensibility and isolation [38].
   d) *Microservices.* This pattern involves deploying components individually for straightforward scalability, deployment, and decoupling. The architecture is distributed, with components accessed remotely; it has evolved from layered and service-oriented architecture patterns to tackle scalability and deployment challenges [38].
   e) *Space-based.* Also known as Cloud architecture, it achieves high scalability by utilizing distributed shared memory (tuple space) and replacing the central database with replicated in-memory data grids. Application data resides in memory and is replicated among active processing units, alleviating central database bottlenecks and enabling efficient scaling. Components encompass processing units and virtualized middleware (handling synchronization and communication) [38].
   f) *Other.* Include alternative abstract representations or descriptions of the software system's structure, behavior, and interactions, offering a high-level understanding of system components, relationships, and collaboration for desired functionality.
2) *Architecture Type.* It categorizes architecture types for integrating the FC-BC-FL technologies, based on FL architecture perspectives (See Figure 7) [23], [31].
   a) *Centralized:* This architecture involves a single central node responsible for communication, model aggregation, and deployment for client/edge devices [23], [37].
   b) *Collaborative:* In this architecture, devices create a mesh-like network, connecting either to a central server or nearby devices based on proximity [23]. This approach can be divided into Dispersed Architecture, involving two stages: sub-global model aggregation within device groups and global model computation through centralization or distribution. Two categories exist within this sub-classification: centralized dispersed FL and distributed dispersed FL. Challenges remain concerning client privacy and non-IID data [37].

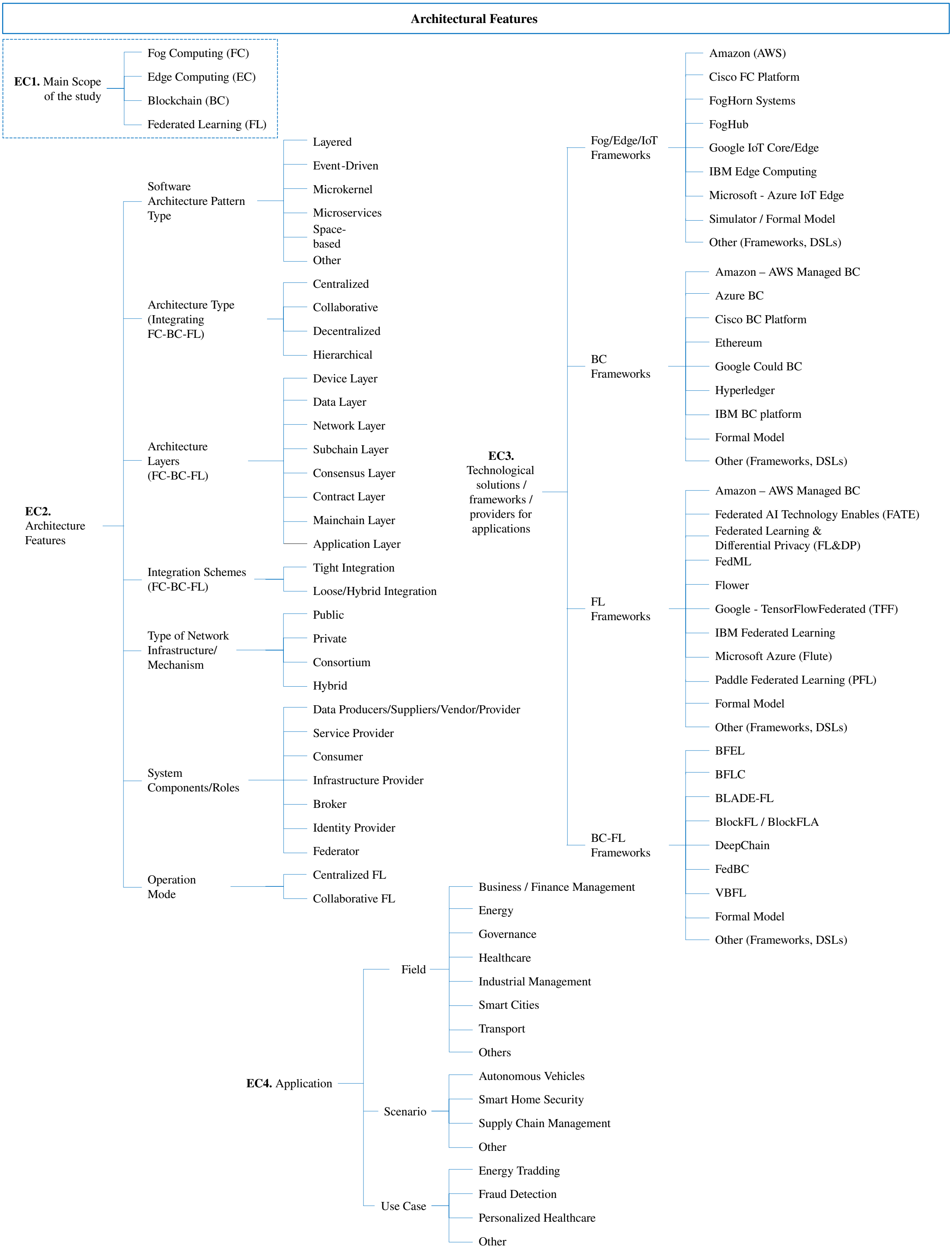


FIGURE 7. SRQ1 Extraction Criteria. Architectural Features (EC2-EC4)

c) *Decentralized (Fully Distributed):* In this architecture, processing shifts to clients or edge nodes, eliminating the need for a third-party entity to aggregate the global model. Clients connect in a P2P or mutual communication manner to exchange local model updates and aggregate the global model [23], [37].

d) *Hierarchical:* It includes regional coordination nodes to manage various edge/fog clusters, thereby reducing the central node's workload [12], [23], [31], [37]. It also considers *regional architecture* as a sub-classification, wherein edge clusters are assigned to regional aggregation nodes, eliminating central aggregation [23].

3) *Architecture Layers.* Considering that IoT applications are commonly presented in Layered architectures, this sub-extraction criteria provides a list of layers that can be utilized, and are common, in IoT applications when integrating FC-BC-FL (See Figure 7) [12], [36], [39].

a) *Device Layer (Infrastructure/Physical):* Consisting of clients participating in the application (e.g., mobile devices, computers, sensors) [12], [39].

b) *Data Layer:* This layer collects, stores, and manages data from the devices in the system [12].

c) *Network Layer:* Operating as a decentralized P2P network, allowing direct resource sharing among peers without intermediaries. Peers can serve various functions and are organized based on support roles like wallets, databases, miners, or routing. It enables distributed resource sharing and removes the need for central authorities [12].

d) *Subchain Layer:* Comprising multiple isolated networks with client, leader replica, and follower replica entities, this architecture operates within a multi-access FC scenario. Here, FNs function as independent replicas for transaction authentication and information exchange. The utilization of a BC consortium ensures compliance with device access control [39].

e) *Consensus Layer:* Verifies the block trustworthiness and maintains accurate ledger copies. However, forks can occur due to malicious nodes, network faults, or communication delays, posing a major challenge for consensus algorithms [12].

f) *Contract Layer:* This layer is responsible for the management of digital currency and the creation and administration of SCs [12].

g) *Mainchain Layer:* This layer is designed for FL tasks and is deployed on distributed FNs. Its purpose is to maintain and verify transactions in a decentralized manner [12], [39].

h) *Application Layer:* Represents The application layer is the topmost layer in the software architecture and is responsible for providing specific functionalities and services to end-users [12], [39].

The BC-FL integration is described and classified by considering techniques and implementation schemas, as presented in [20], [24]. Subsequently, the analysis extends to the integration with FC. To analyze this extraction criterion, selected sub-criteria include integration schemes and the type of network infrastructure/mechanism explained below.

4) *Integration Schemes.* The study [12] suggests two types of integration between IoT and BC applications. As well, this integration schema can be extended to the FC-BC-FL integration (See Figure 7).

a) *Tight Integration:* It mandates all IoT communications through BC, making devices peers and recording interactions for accountability, thus facilitating comprehensive monitoring [12].

b) *Loose / Hybrid integration:* It involves optional interaction recording on BC, optimizing resource usage. It combines decentralized recording with real-time communication for frequent interactions, yet requires careful distance optimization and compromises decentralization security [12].

5) *Type of Network infrastructure/Mechanism.* The studies in [20], [21], [24], [40] outline BC-FL integration mechanisms, which remain consistent even when FC is incorporated. These mechanisms are derived from various BC types. Illustrated in (Figure 7), they comprise:

a) *Public:* In this case, all data is visible to nodes, enabling participation in FL and resource contribution. BC consists of connected blocks with cryptographic hashes, ensuring tamper resistance and immutability. Data privacy relies on encryption or hashes [24], [40].

b) *Private:* Limits BC network to selected participants for data privacy in sensitive scenarios. Write permission is restricted to an organization or group, providing tighter control. Offers flexible configuration, and central governance with controlled mining [20], [24], [40].

c) *Consortium* [1]*:* It merges public and private BCs, using a defined group for block validation. It employs a signature mechanism for approval and spans multiple organizations. Consensus is regulated by pre-authorized nodes, and reading rights can be public or limited. However, controlled consortium BCs risk tampering if nodes permit [20], [21], [24], [26], [40].

d) *Hybrid:* It combines public and private features, offering flexibility and balance in data sharing and control. This versatility suits a range of applications, providing transparency in some areas while preserving privacy in others, thereby offer-

[1] A Consortium BC involves multiple collaborating organizations with restricted access, enabling shared governance and data sharing. Conversely, a Hybrid BC combines public and private elements, allowing both public participation and private data access. The key difference lies in their fundamental structure and participant control.

ing organizations benefits from both BCs within a single solution [20], [24], [40].

6) *System Components/Roles.* This criterion categorizes the roles present in federated system applications, infrastructures, and smart services. The components/roles considered presented in Figure 7 are described below:
   a) *Data Generation:* This entity generates or legally owns/controls the data. It includes sub-roles like Producers, Suppliers, Vendors, and Providers [12], analyzed in EC18 within the FL domain.
   b) *Service Provider:* Offers software and ML models to the ecosystem [12].
   c) *Consumer:* An entity that consumes resources/assets (e.g., data, services) following guidelines and policies provided by Producers/Providers [12].
   d) *Infrastructure Provider:* A participant that supplies computing resources to the ecosystem [12].
   e) *Broker:* Facilitates resource registration and discovery (e.g., infrastructures, services, data sets) via metadata and self-descriptions [12].
   f) *Identity Provider:* Creates and manages participant identity information [12].
   g) *Federator:* Enables and facilitates interaction between providers and consumers [12].
7) *Operation Mode.* Describes the FL operation within the application, based on the aggregation process for the global model [20]. Two types exist (see Figure 7):
   a) *Centralized FL:* In this scenario, global model updates occur on a central server, where local model parameters are aggregated (this process relies on the central server for both the aggregation process and global model updates) [20].
   b) *Collaborative FL:* In this setup, aggregation begins at the end devices. Subsequently, the aggregated model is shared with the central server for final aggregation. Here, devices with limited communication or resources share their model parameters with nearby devices [20].

EC3: Technological solutions/frameworks/providers for applications. This extraction criterion identifies and categorizes technological solution providers or frameworks for each of the selected technologies (See Figure 7):

1) *Fog/IoT/Edge Frameworks.* Categorizes popular frameworks for Fog solutions (including Edge and IoT), including Amazon Web Services (AWS), Cisco FC Platform, FogHorn Systems, and others, alongside self-developed and Domain-Specific Languages (DSLs) [2].
2) *Blockchain Frameworks.* Groups of well-known BC frameworks such as AWS, Azure BC, Ethereum, Hyperledger, and more, supplemented by DSLs and other frameworks.
3) *Federated Learning Frameworks.* Classifies leading Federated Learning frameworks like AWS, FATE, FL&DP, TFF, and more, encompassing self-developed models, DSLs, and other [41], [42].
4) *BC-based FL Frameworks.* Categorizes existing BC and FL integration frameworks, including BFEL, BFLC, Blade-FL, and others, along with self-developed models and additional frameworks [43]–[48], [48], [49].

EC4: Application. This criterion classifies application scenarios, further subdivided to categorize each analyzed primary study's application (see Figure 7) :

1) *Field of application.* It encompasses various typical application fields, including Business/Finance Management, Energy, Governance, Healthcare, Industrial Management, Smart Cities/Smart Homes, Transport, and others [17], [26], [50]–[52] (see Figure 7).
2) *Scenario.* It classifies deployment situations, environments, or contexts where technology or system (refer to Figure 7). It characterizes unique circumstances under which the technology is applied, emphasizing challenges, requirements, or usage conditions affecting implementation or operation. Examples can include Autonomous Vehicles, Data Governance, Smart Home Security, Supply Chain Management, and more.
3) *Use case.* It outlines specific instances or examples that illustrate how a technology or system is utilized to address a particular problem or fulfill a specific need (see Figure 7). These depictions detail interactions, processes, and advantages of practical technology implementation. Examples include Energy Trading, Fraud Detection, Personalized Healthcare, and more.

The following criteria outline the essential technical components and attributes of Fog Computing about the FC-BC-FL integration (See Figure 8). They assist in defining the integration's elements and improving understanding for a more precise categorization of each primary study.

EC5: Cloud features. It describes the cloud structure in two main parameters as shown in Figure 8, and described below:

1) *Type of cloud.* Defines the type of cloud used in the solution [17], [53]. Figure 8 present their classification:
   a) *Private.* Here, services and infrastructure are provided by third-party providers via the Internet. Resources are shared among multiple users. Scalability and cost-effectiveness are key advantages.
   b) *Public.* In this setup, infrastructure and services are tailored for a single organization, offering greater control, security, and customization options, though with increased maintenance and infrastructure requirements.
   c) *Community.* Shared clouds for organizations with common interests, such as government agencies or educational institutions, facilitate resource sharing, collaboration, and security control.

[2] A Domain-Specific Language (DSL) is a specialized programming or specification language designed for a specific industry or task, solving problems within that domain with tailored syntax and semantics, enhancing tasks like scientific computing or financial modeling.

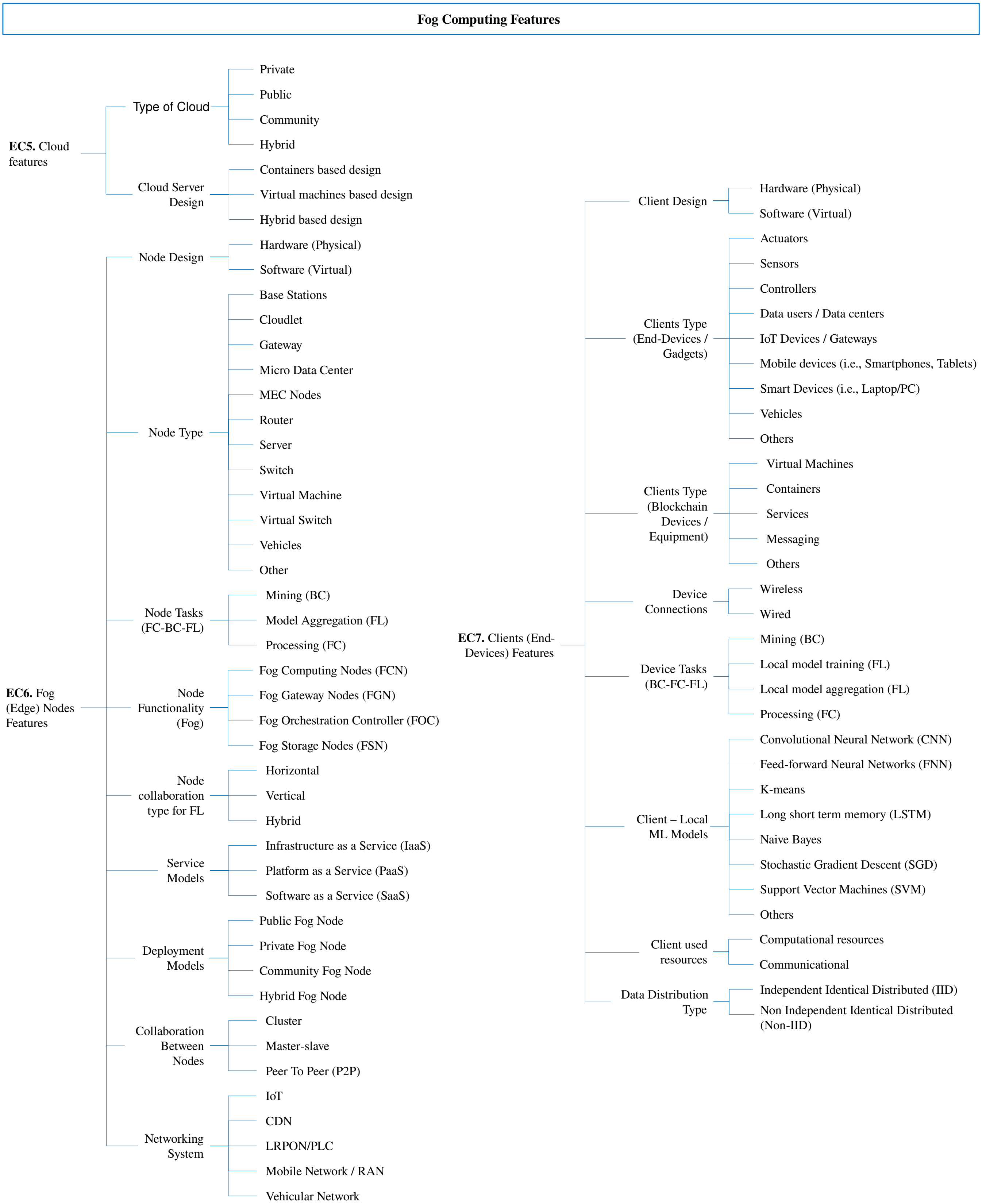


FIGURE 8. Fog Computing Features (EC5-EC7)

   d) *Hybrid.* It combines public and private clouds, allowing organizations to leverage the benefits of both. It offers flexibility, scalability, and the ability to handle varying workloads efficiently.

2) *Cloud server design.* Define FL aggregation server types for cloud implementation [17] (see Figure 8).
   a) *Containers based design.* This design offers lightweight, scalable deployment with faster startup times [17].
   b) *Virtual Machines based design.* provide stronger isolation and compatibility.
   c) *Hybrid based design.* Describes a combination of container-based and virtual machines.

EC6. Fog (Edge) nodes features. The FN serves as a pivotal element within the FC architecture [8]. This criterion assists in delineating and recognizing the attributes of this element within the review of primary studies, utilizing the subsequent sub-criteria as depicted in Figure 8:

1) *Node Design.* It encompasses both hardware (Physical) and Software (Virtual) aspects [8], [17].
2) *Node Type.* This criterion outlines different FN configurations, encompassing base stations, cloudlets, gateways, micro data centers, MEC nodes, routers, servers, switches, virtual machines, virtual switches, vehicles, and other customized designs. Each node type corresponds to specific functionalities and roles within the FC architecture, contributing to the ecosystem's overall efficiency and performance [8], [17], [50], [54].
3) *Node Tasks.* This sub-extraction criterion provides insights into the tasks that a node can undertake within an FC-BC-FL solution (see Figure 8).
   a) *Mining (BC).* The node engages in mining activities for BC-related operations.
   b) *Model Aggregation (FL).* The node facilitates FL processes by aggregating models of end devices.
   c) *Processing.* The node handles conventional FC tasks, such as data storage and processing.
4) *Node Functionality (Fog).* It outlines the designated role of the FN [8], [12] (see Figure 8).
   a) *Fog Gateway Node (FGN).* A node functioning as an intermediary, connecting IoT devices at the edge with the Fog Computing infrastructure [12].
   b) *Fog Orchestration Controller (FOC).* A node responsible for establishing a control layer in the Fog, overseeing resources, and coordinating communication. FOCs manage tasks such as task offloading, scheduling, and resource allocation while considering factors like communication cost and latency [12].
   c) *Fog Computing Node (FCN).* A node comprising one or multiple physical devices endowed with processing and sensing capabilities. These devices empower the Fog to execute tasks assigned by FOCs [8], [12].
   d) *Fog Storage Node (FSN).* A node locating a distributed database/repository [12].
5) *Node Collaboration type for FL.* The classification of collaboration among FNs in the FL process depends on the nature of the application and the available communication resources. Training an FL model for a massive number of IoT devices incurs communication resource overhead. Collaborating between nodes helps alleviate this process [17], [33] (see Figure 8).
   a) *Horizontal.* FNs at the same level collaborate and share resources to collectively handle tasks and provide services [17]. Collaboration between end devices and edge/fog servers [33].
   b) *Vertical.* FNs at different levels collaborate with higher-level nodes providing support and offloading tasks to lower-level nodes. Collaboration occurs between Edge/Fog and the Cloud [17], [33].
   c) *Hybrid.* Merges horizontal and vertical collaboration, enabling flexible and adaptive resource sharing and task distribution in FNs [17], [33].
6) *Service Models.* Similar to the Cloud, the FN offers service models that encompass IaaS, PaaS, and SaaS (see Figure 8) [8].
7) *Deployment Models.* Just like the Cloud, the FN has the following deployment models (see Figure 8) [8]:
   a) *Public FN.* A node provisioned for open use by the general public [8].
   b) *Private FN.* A node dedicated to a single organization with multiple consumers [8].
   c) *Community FN.* A node provisioned exclusively for use by a specific community of consumers from organizations that have shared concerns [8].
   d) *Hybrid FN.* A complex node formed by combining private, community, and public nodes [8].
8) *Collaboration Between Nodes.* Are the methods for coordinating collaborative interactions among diverse FNs within the edge network [54] (see Figure 8):
   a) *Cluster.* Nodes collaborate by forming clusters based on homogeneity or location, considering load balancing and functional development [54].
   b) *P2P.* In FC, node P2P collaboration is common, either hierarchical or flat order. P2P collaboration can be home, local, or non-local based on proximity. It enables the sharing of processed output and virtual computing instances but raises concerns about reliability and access control [54].
   c) *Master-slave.* It is a master FN that controls the functionalities of slave nodes. This approach, along with cluster and P2P interactions, can create a hybrid collaborative network in FC [54].
9) *Networking System.* There exist several computing paradigms in different networking systems where FC has been integrated (see Figure 8), including:
   a) *IoT.* Networking system for device-to-device interaction, categorized as industry or home-based

execution environment, stated in various forms like wireless sensors/actuators, Cyber-Physical Systems, and embedded system networks [54].

b) *Content Distribution Network (CDN).*Networking system composed of distributed proxy servers that provide content to end-users ensuring high performance and availability [54].

c) *Long-Reach Passive Optical Network (LRPON) / Power Line Communication (PLC).* FC integrated with LRPON optimizes home, industry, and wireless backhaul network design, and additionally integrates into electric power distribution [54].

d) *Mobile Network (MN) / Radio Access Network (RAN).* FC in networking systems in mobile networks, particularly in 5G, as well as in other mobile networks like 3G and 4G. Besides, the Radio Access Network (RAN) facilitates communication of individual devices with other entities of a network through radio connections [54].

e) *Vehicular Network (VN).* Networking systems connecting vehicles conformed by computational and networking capabilities [54].

EC7. Clients (End Devices) features. Clients play a crucial role in FC-BC-FL integration, interacting with central servers or distributed networks for service, data, or resource requests. This criterion helps identify client attributes via the sub-criteria in Figure 8.

1) *Client Design.* The client design can either be Hardware (Physical) or Software (Virtual) (see Figure 8) [17].

The clients represent the configuration and functionality of the devices. Two types of clients are presented:

2) *Type of Clients (End-Devices/Gadgets).* They encompass a variety of categories, including actuators, sensors, controllers, data users/data centers, IoT devices/gateways, and mobile devices. The latter category includes devices with mobility within the network, such as cell phones, tablets, and smartwatches. Additionally, there are smart devices with high computing capabilities (e.g., laptops, PCs, Raspberry Pi). Besides, vehicles and other categories (see Figure 8) [12], [17], [55].

3) *Type of clients (BC Devices/Equipment).* Client types for BC include virtual machines, containers, services, messaging, and others (see Figure 8) [12], [23], [24].

4) *Device Connections.* The clients can be connected both wireless or wired (see Figure 8) [12].

5) *Device Tasks (FC-BC-FL).* Describes the possible tasks that the end-device performs in a FC-BC-FL solution, including (see Figure 8):

a) *Mining (BC).* Device performs BC mining tasks.

b) *Local Model Training (FL).* End device participates in FL processes to train models on-device.

c) *Local Model Aggregation (FL).* The end device contributes to FL tasks involving model aggregation within the end devices.

d) *Processing.* End device undertakes conventional FC functions (e.g., data storage, management).

6) *Client - Local ML models.* Describe the ML models potentially executed by clients (see Figure 8) [17], [20]. These models encompass Convolutional Neural Networks (CNN), Feed-Forward Neural Networks (FNN), K-means, Long short-term memory (LSTM), Naive Bayes, Stochastic Gradient Descent (SGD), Support Vector Machines (SVM), and others.

7) *Clients used resources.* The clients can use the following resources when performing (see Figure 8)

a) *Computational Resources.* Refer to the computing power, processing capabilities, and storage capacity of devices or systems. These resources are essential for performing data processing, running algorithms, and executing tasks efficiently [17].

b) *Communicational Resources.* Represent the network infrastructure, bandwidth, and communication protocols that enable data exchange and connectivity between devices or systems. These resources facilitate data transfer, and communication between nodes, and support the information flow within a network [17].

8) *Data Distribution Type.* It describes how data is distributed within the clients' datasets, especially in the ML and data analysis context (see Figure 8) [56], [57].

a) *Independent Identically Distributed (IID).* Here, each data point within the client's dataset is independent of others, and these data points are drawn from the same underlying distribution [56], [57].

b) *Non-IID.* Are datasets within the client where data points are not independent and can come from different underlying distributions. Such situations often occur in real-world contexts due to variations in data sources or devices [56], [57].

The following extraction criteria highlight the Blockchain key attributes to consider during the primary studies review (see Figure 9). These attributes are organized following the layered architecture, covering from the contract to the physical layer (The application layer was previously analyzed).

EC8. BC Contract Layer. This criterion concerns the contract layer, encompassing script codes, algorithms, and SCs embedded in the BC to execute complex business rules. These contracts automatically trigger predefined actions or transactions upon meeting specific conditions agreed upon by network nodes [13], [23], [33], [37]. The subsequent subcriteria (see Figure 9) outline the analyzable features.

1) *Type of Contract.* Defines the types of contacts that can be part of the BC system [13], [58] (See Figure 9).

a) *Scripts.* The BC's contract layer introduces programmable features, enabling advanced scripting. Scripts are basic code pieces used for simple transaction validation in BC networks [58].

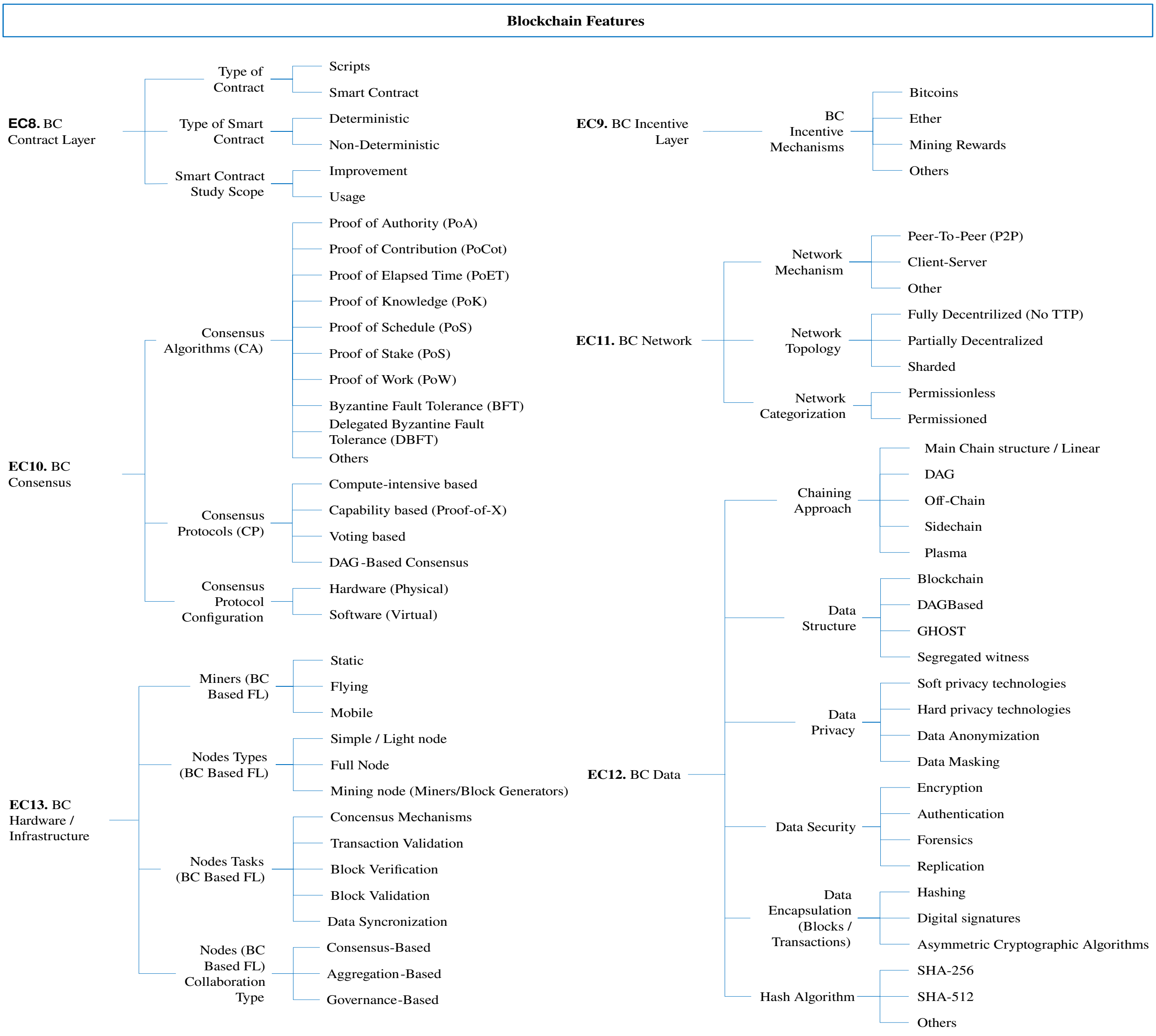


FIGURE 9. Blockchain Features (EC8-EC13)

b) *Smart Contracts.* A SC is a more powerful form of contract that enables the automation of complex agreements and business logic on the BC [13]. It is a data and code collection, also referred to as functions and states, which is deployed using cryptographically signed transactions on the BC. Examples of platforms with SCs include Ethereum's SCs and Hyperledger Fabric's chain code [34].

2) *Type of Smart Contracts.* (see Figure 9).

a) *Deterministic Smart Contracts.* These contracts execute actions based solely on predefined conditions and do not require any external input or off-chain data. They are entirely self-executing and deterministic in nature [59].

b) *Non-Deterministic Smart Contracts.* These contracts rely on external input or off-chain data to execute actions. They may involve human intervention or external systems to trigger certain actions or decisions within the contract [59].

3) *Smart Contracts Study Scope.* (see Figure 9).

a) *Improvement.* If the primary study proposes alternative methods to enhance the SC functionality verification. These methods can be Modeling-driven or Optimization-driven [58].

b) *Usage.* If the primary study demonstrates the utilization of SCs across various domains. This usage can be either resource-driven or driven by cross-organizational collaboration [58].

EC9. BC Incentive Layer. This layer, economically rewards specific nodes, motivating their active block verification and decentralization maintenance (See Figure 9). It ensures incentive issuance and distribution, encouraging node participation in the consensus process [23], [60].

1) *BC Incentive Mechanisms.* They encompass a variety of approaches designed to motivate and reward participants (nodes) within a BC network for their contributions to maintaining the network's security, consensus, and overall functionality [23]. These mechanisms are categorized as follows: Bitcoins, Ether, Mining Rewards, and others (e.g., Governance Participation, Staking Rewards, Transaction Fees, ZCash).

EC10. BC Consensus Layer. This extraction criterion outlines the Consensus Layer in BC, detailing the protocols that network participants follow to establish consensus on valid transactions and ensure system security. This layer can employ diverse algorithms for achieving decentralized agreement [23], [33]. The attributes of this layer help define the core of this extraction criterion (See Figure 9), including:

1) *Consensus Algorithms (CA).* In BC, the CAs ensure ledger integrity, security, and efficiency among untrusted nodes in the P2P network. Their primary goal is to achieve agreement on adding new blocks to the ledger. Different CAs are used in BC systems, each with its strengths and weaknesses [20], [22], [33], [34], [40], [61] (See Figure 9). The CA categorization is:
   a) *Proof of Authority (PoA).* It relies on trusted validators who are authorized to create new blocks and validate transactions based on their recognized identity within the network.
   b) *Proof of Contribution (PoCot).* It rewards nodes based on their tangible contributions to the network, such as providing computational resources, storage, or services that improve functionality.
   c) *Proof of Elapsed Time (PoET).* It ensures fairness by having nodes wait for a randomly assigned time, and the first node to complete its waiting time gets the right to propose a new block.
   d) *Proof of Knowledge (PoK).* It requires nodes to prove their possession of specific knowledge or information before they can participate in consensus. It emphasizes using knowledge-based proofs.
   e) *Proof of Schedule (PoS).* It selects block creators based on a predefined schedule, often determined by factors like the node's age, wealth, or a combination of both. This aims to provide a deterministic way of choosing validators.
   f) *Proof of Stake (PoS).* Similar to PoW, it chooses validators based on the cryptocurrency they hold and are willing to "stake" as collateral. This reduces energy consumption compared to PoW and encourages active participation from those who hold more stake in the network.
   g) *Proof of Work (PoW).* It entails solving complex mathematical problems (using significant computational power). Miners compete to solve the problem, and the first one to solve it gets to create the next block.
   h) *Byzantine Fault Tolerance (BFT).* It addresses faulty nodes in a network, ensuring consensus despite malicious behavior or failures. It guarantees agreement even with a specific number of nodes behaving maliciously or failing, enhancing security and consistency in untrustworthy or unreliable node scenarios. The algorithm employs a voting or agreement process among nodes to determine transaction validity.
   i) *Delegated BFT (DBFT).* Extends BFT by involving trusted nodes or delegates in consensus. These nodes validate transactions through agreement. Trusted delegates streamline consensus, speeding transaction confirmations, especially in BC networks with predetermined validators or delegates responsible for consensus.
   j) *Others*. Covering extra CAs such as Proof of *"X"* (Authentication, Capacity, Importance, Learning, Retrievability, Space, Storage, Training Quality, Verification), Delegated Proof of Stake (DpoS), Crash Fault Tolerance (CFT), and more.

2) *Consensus Protocols (CP).* CPs underpin BC security and performance. CPs achieve decentralized consensus on a shared transaction ledger, defining how nodes exchange messages and make decisions. Design choices impact transaction capacity, scalability, and fault tolerance [33], [62]–[64]. The general CP types are:
   a) *Compute-intensive based.* CPs based on compute-intensive algorithms are characterized by their high energy consumption during the mining process. It primarily focuses on PoW [62].
   b) *Capability based (Consensus based on Proof-of-X).* This category extends beyond energy consumption and considers non-computing capabilities. These protocols factor in various aspects, such as the amount of cryptocurrency owned by a miner, their contribution to the community, trustworthiness, or the storage they possess [62]. It includes all "proof-of-X" except PoW [63], [64].
   c) *Voting based.* They use voting systems to elect a miner for block generation, addressing energy consumption and wealth dominance concerns. They tolerate Byzantine faults, ensuring consensus despite node failures or malicious behavior. This category is divided into BFT-based consensus, achieved by following Byzantine algorithms [63], [64], and CFT-based. BFT-based prevents failing and malicious nodes, while CFT-based addresses only failing/crashing nodes using protocols like Raft and Federated [62].

d) *DAG-Based Consensus.* This category covers CPs that use DAG-Based algorithms [64].

Note that CA defines the overall strategy for consensus, while CP specifies the detailed rules and mechanisms for communication among nodes to achieve consensus.

3) *Consensus Protocol Configuration.* The protocol configuration plays a crucial role in a BC network, impacting security and scalability [40].

a) *Security.* For security purposes, Nakamoto consensus systems employ strategies such as awaiting a specific number of blocks (X-Block confirmation) or implementing checkpoints to mitigate double-spending risks (Checkpointing).

b) *Scalability.* It can be improved by adjusting block size or mining difficulty to increase transaction processing rate (e.g., adjusting original block size and frequency or increasing block size/reducing mining size). However, this might lead to more frequent forks and longer waiting times for confirmation blocks by users.

EC11. BC Network. This BC layer defines the networking implementation within the BC system [13]. This criterion is analyzed by the following sub-criteria (see Figure 9).

1) *Network Mechanism.* This aspect's primary goal is to distribute data generated by the data layer. The network mechanisms include (see Figure 9):

a) *Peer-to-peer (P2P).* In decentralized networks, each participant functions as a peer [13].

b) *Client-Server.* While P2P networks are common in BC to ensure decentralization and eliminate central intermediaries, BC-FL integration might involve other network architectures depending on specific BC implementation requirements [13].

2) *Network Topology.* Referring to the structure and organization of nodes and their communication [13], [33], [40], different network topology types are used in BC systems, categorized as follows (see Figure 9):

a) *Fully decentralized.* his topology lacks a central authority [13], [33], [40].

b) *Partially decentralized.* Some centralization elements exist in this topology [13], [33], [40].

c) *Sharded.* This topology divides into small subsets called shards to achieve scalability [13], [33].

3) *Network Categorization.* BC networks are categorized based on permission models, outlining authorized entities for maintenance [33], [34] (see Figure 9).

a) *Permissionless.* Here, anyone can participate (as the public internet). Permissionless BC networks are decentralized ledgers open for anyone to publish blocks without requiring permission. Users can read/write to the ledger, but malicious users might try to undermine the system [33], [34].

b) *Permissioned.* This network type requires authorization for block publishing, controlling read and transaction access. They can be open for reading to anyone or restrict access to authorized individuals. These networks use consensus models for block publishing, being faster and less computationally intense than the permissionless [33], [34].

EC12. BC Data. The significance of data in BC is relevant. Consequently, This criterion highlights the attributes pertaining to data within a BC environment (refer to Figure 9).

1) *Chaining Approach.* This encompasses the structure of authenticated data from the consensus layer storage (see Figure 9). It includes:

a) *Main chain structure (Linear).* It is the traditional BC, where blocks are linked in a linear chain, and each block contains the hash of the previous block (e.g., Bitcoin) [13], [33].

b) *DAG chain.* In this case, the blocks are connected in a non-linear way, forming a graph structure without cycles. Each block references multiple previous blocks, increasing scalability and transaction parallelism (e.g., IoTa, PriFob) [33].

c) *Off-chain.* In this chaining approach, the transactions are processed outside the main BC, and only the final outcome is recorded on the main chain. This approach reduces on-chain congestion and enhances privacy [13], [33].

d) *Sidechain.* An independent BC that is interoperable with the main BC, allowing specific functions or applications to operate autonomously while maintaining a connection for enhanced security and flexibility [13].

e) *Plasma chain.* This represents a hierarchical structure comprising child chains linked to the main BC, utilizing merkleized proofs to facilitate rapid and cost-effective transactions. This architecture enhances the scalability of BC by reducing the workload on the main chain and enabling parallel processing. [13].

2) *Data Structure.* In BC, it defines how the information is stored and linked together to form a chain of blocks. This data structure allows for the secure and efficient storage and retrieval of data, enabling the decentralized and distributed nature of BC technology [40]. The BC systems use several data structures (see Figure 9):

a) *Blockchain.* It is the traditional d data structure formed by interconnected blocks. When conflicting blocks arise, network participants typically opt for the longest chain as the valid one [40].

b) *GHOST (Greedy Heaviest-Observed Sub-Tree).* By using the GHOST protocol, miners modify the data structure by including competing independently mined blocks (uncle blocks) in their chain, adding weight to their chain for selection as the main chain. This enhances network efficiency and throughput by recognizing concurrent work and incorporating uncle blocks in the consensus [40].

c) *BlockDAG.* Moving from a linear chain to a Directed Acyclic Graph (DAG) permits the integration of non-conflicting transactions from uncle blocks into the primary chain. Selection criteria can favor the longest chain or the heaviest subtree, determined by block length or collective difficulty. Additionally, customization of the internal block structure is achievable. [40].

d) *Segregated witness.* It is a proposed solution in the Bitcoin community that separates transaction signatures (witnesses) from the transaction data, reducing their impact on block size. This approach improves scalability by decreasing storage requirements since all transactions are replicated on every node in a BC network [40].

3) *Data Privacy.* It is the protection of sensitive and personal information on the BC, ensuring authorized access and restricting unauthorized disclosure [12], [40]. The data privacy can be achieved by (see Figure 9):

a) *Soft privacy technologies.* Are techniques that protect privacy without fundamentally altering data, allowing for data processing while safeguarding sensitive information.

b) *Hard privacy technologies.* Are strong measures involving irreversible data transformation or encryption, offering enhanced privacy assurances while potentially constraining specific data processing capabilities.

c) *Data Anonymization.* Refers to the process of removing or modifying personally identifiable information from datasets to prevent individual identification while enabling useful data analysis.

d) *Data Masking.* It is the Technique of concealing sensitive data with modified or randomized values to protect privacy while maintaining data realism for specific purposes like testing or development.

4) *Data Security.* Ensure data integrity, confidentiality, and availability through encryption, access controls, and digital signatures. The techniques include Encryption, Authentication, Forensics, and Replication [12], [40], [52] (see Figure 9).

5) *Data Encapsulation.* In the data layer, the data is collected through transactions from the physical layer. Then, the first step is to encrypt or encapsulate the data [12]. This sub-criterion describes the data encapsulation alternatives for data (see Figure 9).

a) *Hashing.* One-way function, converts data to fixed-size hash and verifies data integrity in BC.

b) *Digital signatures.* They use asymmetric cryptography, provide authenticity and non-repudiation, and verify the sender and data integrity.

c) *Asymmetric Cryptographic Algorithms.* These algorithms employ a pair of keys—a public key and a private key. The public key is utilized for encryption, while the private key is used for decryption. Data encrypted with the public key can solely be decrypted using the corresponding private key, ensuring secure communication between parties without requiring a shared secret.

6) *Hash Algorithms.* It is a mathematical function that transforms an input (often termed a message) into a fixed-size string of characters, known as a hash value or code. This output typically provides a distinct representation of the input data, with minor changes yielding significantly different hash values [65]. This criterion categorizes Hash Algorithms into SHA-256, SHA-512, and Others (e.g., Ethash, ring signatures, cunningham chain) as presented in Figure 9.

EC13. BC Hardware / Infrastructure. This extraction criterion describes the BC physical structure (with special analysis in the BC-based FL) and their relevant elements which are detailed in Figure 9, and analyzed below:

1) *Miners (BC based FL).* Miners are important elements in the BC domain. They validate transactions, solve complex puzzles to add blocks and secure the network while earning rewards. In the BC-based FL context, miners are responsible for the secure and trustable exchange of learning model parameters in a distributed manner [17], [33]. They can be classified into (see Figure 9):

a) *Static.* These stationary devices play a crucial role in the BC network, contributing their high computational power to validate transactions, generate new blocks, and ensure the security of FL. Static miners exhibit the highest computational power, the lowest forking probability, and the highest block propagation capability [17], [33].

b) *Flying.* Are Unmanned Aerial Vehicles (UAVs) that provide computational resources to validate transactions and generate blocks. This miner has low computational power, high forking probability, and the highest block propagation [17], [33].

c) *Mobile.* Similar to autonomous cars, movable devices actively participate in mining. They use computational power and connectivity to contribute to the BC network, ensuring secure parameter exchange in learning models. These miners boast high computational power, low forking probability, and low block propagation [17], [33].

2) *Node Types (BC based FL).* The nodes are the network participants [62]. The node types are (see Figure 9):

a) *Simple/Light Node.* A network node that can only send and receive transactions, without storing a copy of the ledger or validating transactions [62].

b) *Full Node.* A node that stores a copy of the entire ledger and can validate transactions [62].

c) *Mining node (Miners/Block Generators).* A full node within the network that possesses the mining capability, which involves creating new blocks and adding them to the BC [62].

3) *Node Tasks (BC based FL).* The nodes in a BC-based FL can perform tasks such as:
   a) *Consensus Mechanisms:* Nodes collaborate to reach a consensus on the validity and order of transactions. Examples include Proof of Work (PoW), Proof of Stake (PoS), and Practical Byzantine Fault Tolerance (PBFT) [60], [62].
   b) *Transaction Validation.* Nodes collaborate to validate transactions by verifying their integrity, authenticity, and adherence to the predefined rules or smart contracts [52], [60], [62].
   c) *Block Propagation.* Nodes collaborate to propagate newly created blocks across the network. This involves broadcasting the block to other nodes for verification and inclusion in their BC local copy [60], [62].
   d) *Block Verification.* Nodes collaborate to verify the correctness of blocks by performing cryptographic operations, checking signatures, and validating transactions within the block [60], [62].
   e) *Block Validation.* The nodes collaborate to validate the entire BC, ensuring its consistency and integrity. They reach a consensus on the BC state and agree on the next block to be added [60], [62].
   f) *Data Synchronization.* Nodes collaborate to synchronize their local copies of the BC, ensuring that they have the same version of the ledger. This involves sharing and updating the BC data between nodes [60], [62].
4) *Node Collaboration Type (BC-based FL).* It defines three types in a BC-based FL solution [33], [44], [62]:
   a) *Consensus-Based Collaboration.* Nodes in the BC network collaborate to reach a consensus on the validity and ordering of federated learning updates. Through consensus algorithms such as Proof-of-Work (PoW) or Proof-of-Stake (PoS), nodes collectively validate and agree on the updates contributed by participating devices or entities. This collaboration ensures the integrity and consistency of the FL process [44].
   b) *Aggregation-Based Collaboration.* Nodes collaborate to perform aggregation of local model updates contributed by participating devices. Each node receives the individual model updates, aggregates them using specific aggregation algorithms (e.g., FedAvg), and generates a new global model. The collaboration among nodes enables the merging of knowledge from different devices while preserving data privacy [44], [60], [66].

These node collaboration types are crucial for the successful operation of BC network-based FL. They enable secure, decentralized coordination among nodes, ensure trustworthy updates and model aggregation, and provide a framework for collective decision-making and governance.

The following extraction criteria outline the main Federated Learning features to be considered when reviewing the primary studies. The features are established based on the layered architecture, starting from the global model layer to the data layer (The application and physical layers were previously analyzed) (see Figure 10).

EC14. FL Global Model. It describes the Global Model layer characteristics by the sub-criteria shown in Figure 10.

1) *Federated Optimization Schemes.* In FL, they aim to minimize the global loss function (i.e., the overall performance measure used to evaluate the accuracy and quality of the global model) [17]. Two types exist:
   a) *Single Task.* In this case, the global federated learning model is trained only for a single task (e.g., FedAvg, FedProx, and q-FedAvg) [17].
   b) *Multi task.* In this case, it involves the training of multiple models for different tasks (e.g., Federated Multitask Learning (FML)) [17].
2) *FL Algorithms.* They enable devices to collaboratively train a shared global model without sharing raw data. Local models' updates are aggregated iteratively, preserving privacy and decentralizing data. Within the realm of FL, various algorithmic approaches have been developed, each tailored to address distinct objectives and challenges. These include Auditable FL (AFL), Communication-Efficient FL (CEFL), Clustered FL (CFL), Chain FL (CFL-Chain), Fine-Grained FL (FGFL), Incentive-aware FL (IFL), Reliability-aware FL (RAFL), Reward FL (RFL), Reputation-Aware FL (RFL), and others [24], [39] (see Figure 10).

EC15. FL Incentive Layer. This category contains the incentive mechanisms employed within the FL domain. These mechanisms serve as strategies or tools intended to incentivize and reward participants, including nodes/miners and end devices, within an FL network. These incentives aim to encourage their active engagement in the learning process and the sharing of updates from their local models while upholding data privacy and security [17], [23]. Some of the incentive mechanisms include (see Figure 10): Auctions, Blockchains, Contact theory, Reinforcement Learning, Game-theoretic, Matching theory, Shapely value, Stackelberg Game [17], [20], [23], and others.

EC16. FL Aggregation. Aggregation is the merging of local model updates from multiple devices to create a global model while preserving data privacy [10], [36]. This criterion helps in identifying important FL aggregation facts (see Figure 10).

1) *FL Aggregator Type (For FL-FC/IoT applications).* An aggregator is the device that performs the aggregation process which consists of initiating, combining local model updates, and constructing new global models. The aggregator types include Cloud Servers, Data Centers, Data Servers, Data Workers, FL Servers, Security Gateway, and Others [10].

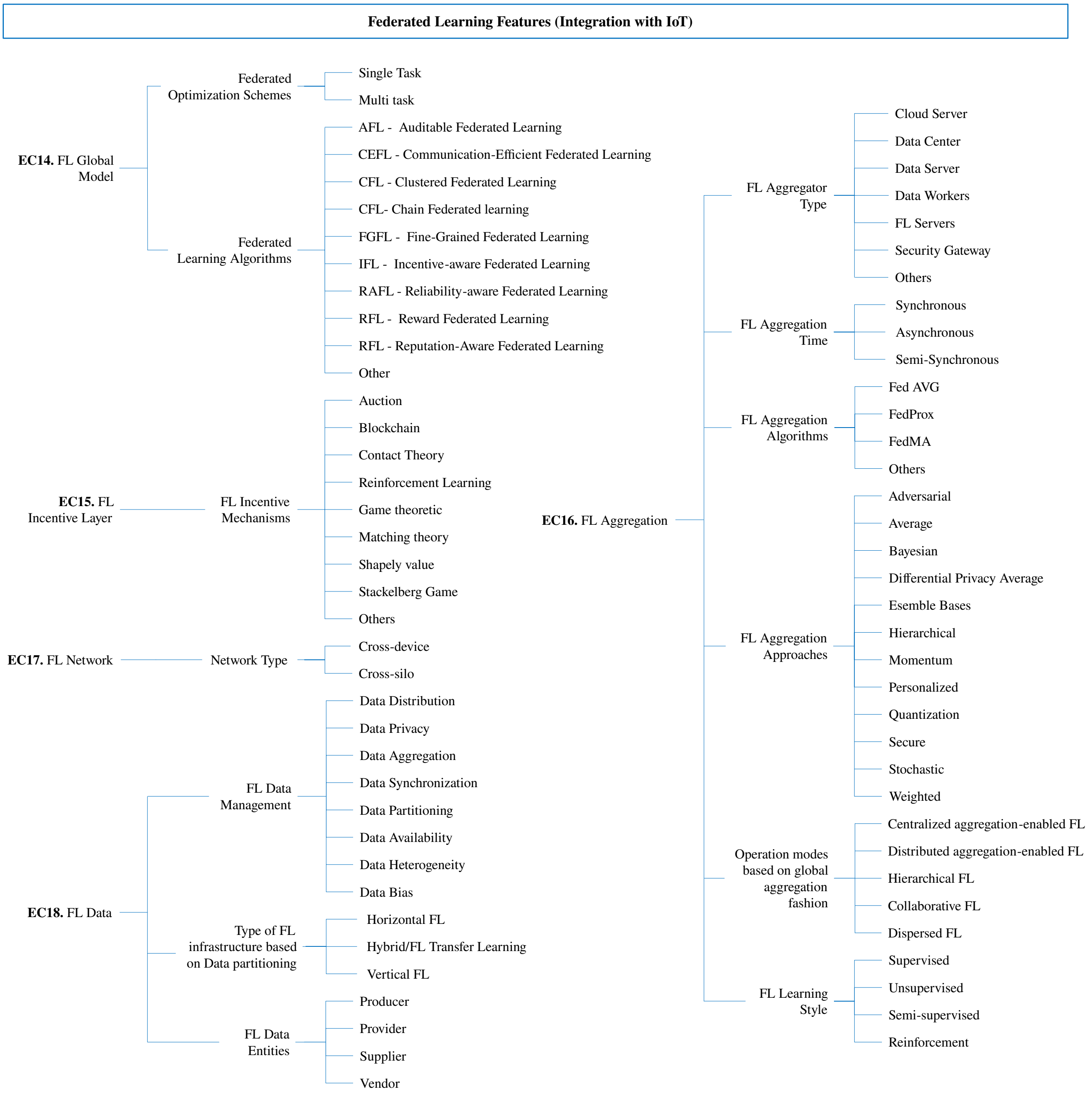


FIGURE 10. Federated Learning Features (EC14-EC18)

2) *FL Aggregation Time.* The FL methods are categorized into three synchronization schemes based on when global model aggregation occurs [23] (see Figure 10).

a) *Synchronous.* This scheme entails simultaneous training of all active devices, resulting in idle time for high computational devices and slow convergence due to the slowest device [23].

b) *Asynchronous.* It allows devices to update the global model separately, improving convergence but taking more communication resources [23].

c) *Semi-Synchronous.* This aggregation time strikes a balance between synchronous and asynchronous processes, enabling local training until synchronization points. It cuts communication costs and optimizes resource utilization, enhancing model convergence, especially for devices with diverse computational capabilities [23].

3) *FL Aggregation Algorithms.* An aggregation algorithm combines outcomes from training individual models on clients' devices with their data, updating the global model. Algorithms include (see Figure 10) Fed AVG, FedProx, FedMA, and others [10], [17], [20], [36], [42].

4) *FL Aggregation Approaches.* As known, the aggregation algorithms in FL are crucial for updating global models. Various approaches are used based on goals like privacy protection, convergence rate improvement, and fraud prevention. Each has pros and cons, and some suit specific contexts better [36] (see Figure 10).
   a) *Adversarial.* It identifies and mitigates the impact of malicious clients or outlier model updates [36].
   b) *Average.* It averages the client updates [36].
   c) *Bayesian.* It is employed for aggregating model updates while considering uncertainty [36].
   d) *Differential Privacy Average.* Introduce random noise to the model updates before aggregation to guarantee privacy [36].
   e) *Ensemble Based.* Merge model updates from various models trained on diverse data subsets [36].
   f) *Hierarchical.* It performs the aggregation process at different levels of a hierarchical structure [36].
   g) *Momentum.* It incorporates a momentum factor into the model updates before aggregation to enhance the speed of convergence [36].
   h) *Personalized.* Takes clients' unique characteristics into account [36].
   i) *Quantization.* Decrease the bit representation of model updates before sending them [36].
   j) *Secure.* It ensures privacy with techniques like homomorphic encryption or secure multi-party computation, safeguarding data during computation and transmission [36].
   k) *Stochastic.* Solutions in this category utilize randomness or probabilistic methods during the model update aggregation process. It aims to enhance privacy, improve convergence, and mitigate malicious or outlier model updates influence [39].
   l) *Weighted.* Assign weights to clients' contributions based on their performance or other criteria [36].

5) *FL Operation modes based on global aggregation fashion.* The global aggregation fashion refers to how local learning models from several devices or servers are combined to create a global model in FL [17]. Thus, based on the mentioned information, the FL operation modes are (see Figure 10):
   a) *Centralized aggregation-enabled FL.* The aggregations take place at distributed servers without using a centralized aggregation [17], [20].
   b) *Distributed aggregation-enabled FL.* The aggregation occurs at distributed servers without relying on a central aggregation point. End devices or servers share their model updates with each other. The aggregations perform distributedly [17].
   c) *Hierarchical FL.* In hierarchical FL, the local learning models are aggregated at edge servers before global aggregation at the cloud [17].
   d) *Collaborative FL.* The devices with limited communication resources send their models to nearby devices with better resources, and the aggregated models are sent to a centralized server [17], [20].
   e) *Dispersed FL.* It is a novel approach where sub-global FL models are computed within different groups, then transferred between groups, and finally aggregated iteratively until a desirable global FL accuracy is achieved [17].

6) *FL Learning Style.* Before the aggregation process, the models are trained following a specific learning style. It refers to the approach or methodology used by an algorithm or model to process and learn from data. It defines how the model generalizes from the provided data to make predictions or decisions on new, unseen data [36]. These styles include:
   a) *Supervised.* Use labeled data to train a model, where input-output pairs are provided, allowing the model to learn to make predictions or classifications on new data [36].
   b) *Unsupervised.* Deal with unlabeled data, finding patterns or structures within it, without explicit target outputs, often used for clustering or dimensionality reduction [36].
   c) *Semi-supervised.* Combine labeled and unlabeled data, leveraging the unlabeled data to enhance model performance with limited labeled data [36].
   d) *Reinforcement.* Involve an agent interacting with an environment, learning from feedback in the form of rewards or penalties to make optimal decisions and actions to reach a specific goal [36].

EC17. FL Network. This extraction criterion classifies the network features in FL. While the criteria discussed in EC6 and EC13 have described important network characteristics of FC and BC, there is one remaining feature that is crucial in the FL domain and in the integration of these three technologies which is the number of participants in the network.

1) *Network Type (FL Scale).* Since FL can occur in a range of types [23], [41]. This extraction criterion defines the scales of FL networks regarding the number of participants (See Figure 10):
   a) *Cross-device.* It involves a large number of client devices, like IoT devices and smartphones, with limited data size and intermittent network connectivity. The challenge is to effectively utilize contributions from diverse devices to collaboratively train a global model [23], [41], [42], [67].
   b) *Cross-silo.* It involves a small number of clients (tens to hundreds), like data centers or organizations, working together with large data sets, reliable connectivity, and powerful computing resources. The security and performance of the FL system depend on the network type and other complex criteria related to ML objectives and privacy constraints [23], [41], [42], [67].

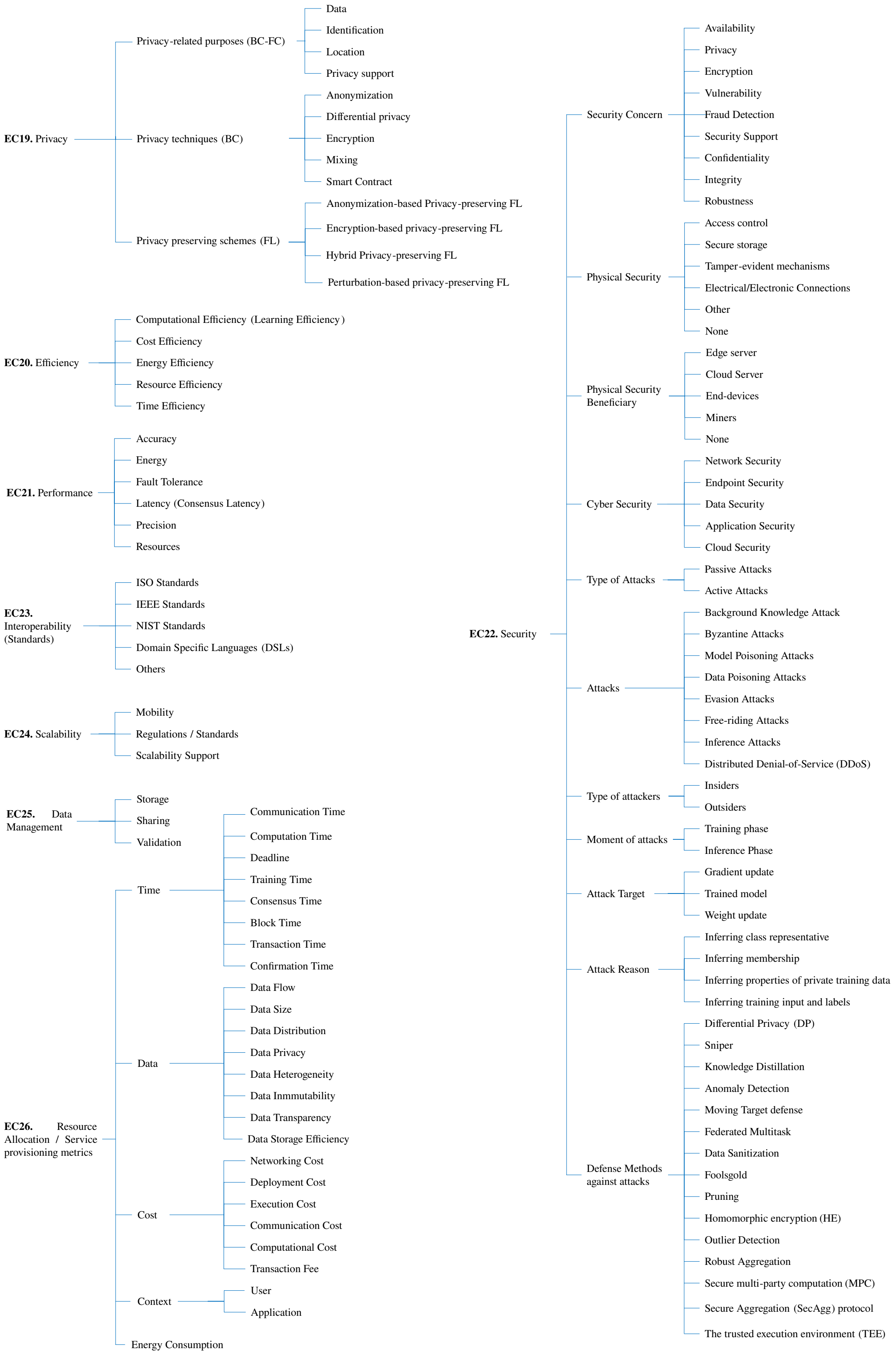
EC18. FL Data. It classifies some of the considerations in the data layer within the FL domain (See Figure 10).

1) *FL Data Management.* The data layer in FL involves various features regarding data that are essential for the FL system functionality, including key data management aspects (see Figure 10).
    a) *Data Distribution.* The data layer deals with how data is distributed across multiple devices, clients, or nodes participating in the FL process. Data can be heterogeneous, non-IID, and may have different features at each device [10], [36], [39].
    b) *Data Privacy.* Privacy preservation is crucial in FL, as data is kept decentralized and sensitive information remains on individual devices. Techniques like differential privacy and encryption are employed to protect data privacy during the FL process [10], [23], [26], [41], [52].
    c) *Data Aggregation.* The data layer manages how local model updates from different devices are aggregated to create a global model. Aggregation methods need to be efficient and secure while considering communication resources [10], [36].
    d) *Data Synchronization.* FL systems need to handle the synchronization of data updates across devices with intermittent connectivity. The data layer ensures timely and reliable synchronization of local model updates [23].
    e) *Data Partitioning.* Data partitioning involves dividing data across devices or clients for training local models. Different partitioning strategies (vertical, horizontal, transfer learning) are employed based on the data distribution and learning objectives [10], [12], [20], [42], [67].
    f) *Data Availability.* The data layer addresses issues related to data availability, as some devices may be offline or inaccessible at times. Data availability mechanisms must ensure that devices can effectively contribute to the FL process.
    g) *Data Heterogeneity* FL systems often deal with data heterogeneity, where devices have different data formats, types, and distributions. The data layer manages these variations to create a coherent global model [12], [17], [23].
    h) *Data Bias.* The data layer handles data bias, ensuring that biased data from specific devices does not negatively impact the global model's performance. Techniques to address bias and fairness are incorporated [36].
2) *Type of FL infrastructure based on data partitioning.* The FL is used in scenarios where data is distributed across multiple devices with different variants of clients, usage data, and applications [20], [37], [42]. This diversity allows ML models to have better generalization capabilities through continuous updates. Data partitioning in FL includes (see Figure 10):
    a) *Vertical FL.* In this case, IoT devices from different clusters with shared data interests train ML models collaboratively without relying on central authority [20], [24], [42].
    b) *Horizontal FL.*In this case, clients with similar data features share their data for collaborative learning [20], [24], [42].
    c) *Transfer Learning.* In this case, pre-trained models are shared among devices to train local ML models, providing better results compared to training from scratch [20], [24], [42].
3) *FL Data Entities.* These are entities, organizations, or businesses that generate or legally own/control the data [12]. The classification is (See Figure 10):
    a) *Data producer.* It is an entity, organization, or business that generates or creates data. They are the source of the data and have legal ownership or control over it. They can be individuals, companies, sensors, devices, or any entity that generates data as part of its operations or activities [12].
    b) *Data Supplier.* It is an entity or organization that supplies or provides data to others. The term "supplier" is used interchangeably with "producer," referring to entities that generate or own data [12].
    c) *Data Vendor.* It is an entity or business that sells or trades data as a product or service. Vendors act as intermediaries between data producers and consumers, aggregating and offering data from multiple sources [12].
    d) *Data Provider.* These entities mediate between data producers and the data ecosystem. They collect data from various producers and offer it to the ecosystem on behalf of the producers, making data access more convenient for consumers [12].

The extraction criteria below outline the impact of FC-BC-FL integration in primary studies across domains like privacy, efficiency, security, interoperability, scalability, data management, resource allocation, service metrics, trust, resilience, access control, heterogeneity, and more (See Figure 12).

EC19. Privacy. It entails the protection of sensitive data and user information during the collaborative model training and chaining process [20], [41], [68], [69]. This criterion defines several categories to analyze the privacy impact in the FC-BC-FL integration across primary studies (see Figure 11).

1) *Privacy-related purposes (BC-FL).*
    a) *Data privacy.* In the BC-FL integration is addressed through accurate data provenance and decentralized access control mechanisms, using spatiotemporal chaotic models and encryption for IoT data protection [68].
    b) *Identification privacy.* BC-based identity management systems with access control and self-certified cryptography ensure secure authentication, confidentiality, and auditability, safeguarding identities in IIoT data [68].



(a)

FIGURE 11. SRQ2 Extraction Criteria (EC19 - EC26)

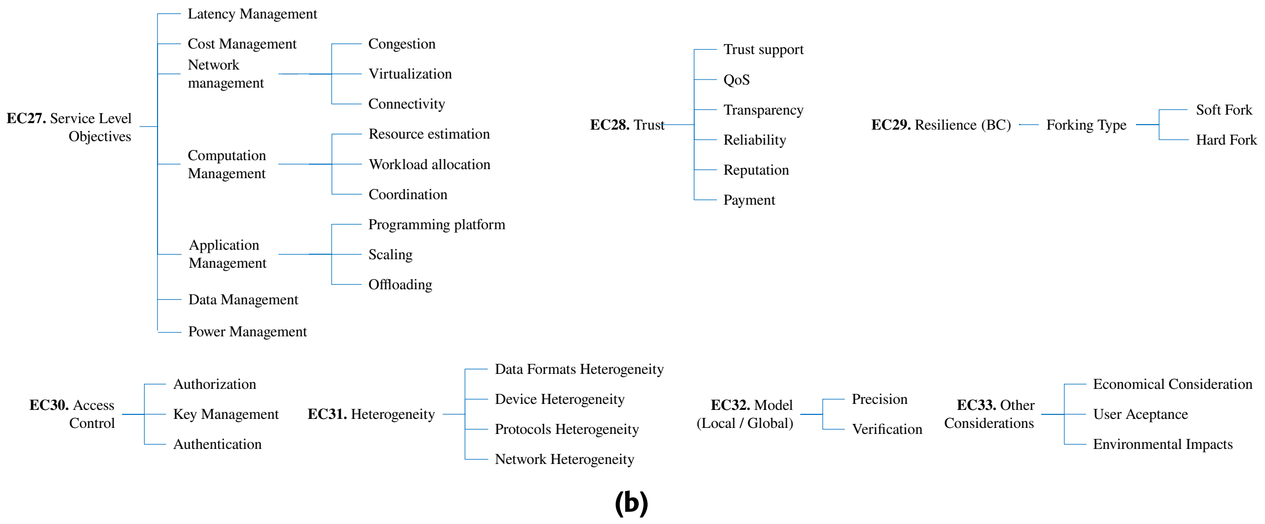


FIGURE 11. SRQ2 Extraction Criteria (EC27 - EC33)

   c) *Location privacy.* BC-enabled vehicular FC and pseudonym systems ensure anonymous verification, secure pseudonym management, and reliable data sharing while protecting the location information of nodes [26], [68].
   d) *Privacy support.* BC integration in FC enhances privacy, reducing the need for centralized third parties through features like Consortium BC and TLSP, preserving data security and privacy [68].
2) *Privacy Techniques.* This criterion classifies the privacy techniques which are methods to safeguard sensitive data, such as encryption, data anonymization, access control, and secure communication, ensuring confidentiality and protection from unauthorized access [20].
   a) *Anonymization.* This method is utilized to safeguard sensitive data by eliminating or encrypting personally identifiable information (PII) from datasets. It guarantees that individuals' identities cannot be associated with specific data entries, preserving privacy and confidentiality. Anonymization finds widespread use in various domains, such as healthcare, finance, and IoT, to thwart unauthorized access and shield user information from potential threats [20].
   b) *Differential privacy.* This privacy protection technique adds controlled noise during query evaluation to safeguard data. It is applied in diverse fields, including BC-enabled IoT. For instance, in healthcare, perturbation-based differential privacy adds noise to patient records, preventing privacy attacks [20], [42].
   c) *Encryption.* Encryption is a widely used privacy technique in BC-enabled IoT and other networks. It ensures secure communication using public key encryption and protects sensitive data. Applications include securing vehicular networks and wearable health devices. However, encryption requires significant computation and may increase communication overhead [20].
   d) *Mixing.* This privacy technique in IoT involves encrypted transactions sent to trusted third-party servers, which then mix and forward them to transmitter nodes, ensuring privacy. To decentralize the process and protect user privacy, CoinShuffle techniques leverage the BC network.
   e) *Smart Contract.* Are programmable codes with condition statements executed when conditions are met. In BC-enabled smart contracts, IoT information is stored in code and deployed when conditions are satisfied [20].
3) *Privacy Preserving FL (PPFL) schemes.* The PPFL aims to achieve a balance between data privacy and data utility when applying privacy-preserving techniques to FL frameworks. The goal is to enable collaborative model training across multiple devices while protecting the privacy of individual data owners [69]. This extraction criterion classifies four PPFLs:
   a) *Anonymization-based Privacy-Preserving FL.* These methods prioritize privacy and data utility by applying anonymization schemes like k-anonymity to protect private data during collaborative model training. They defend against malicious attacks and have shown better privacy preservation and model performance compared to differential privacy-based FL methods [69].
   b) *Encryption-based Privacy-Preserving FL.* These methods employ cryptographic techniques for privacy preservation, and these methods can be subcategorized into homomorphic encryption-based, secret sharing–based, and secure multiparty computation-based PPFL methods [52], [69].

c) *Hybrid Privacy-Preserving FL.* These methods strike a balance between data privacy and utility by combining cryptographic and perturbation-based techniques. They address computation and communication overheads while preserving privacy and data accuracy. Various approaches integrate homomorphic encryption, secret sharing, differential privacy, and secure multiparty computation to ensure data privacy without sacrificing the accuracy of FL models [69].

d) *Perturbation-based Privacy-Preserving FL.* This method adds intentional noise to data for privacy while enabling collaborative model training. Noise obfuscates individual data, preserving privacy. Four subcategories include global and local differential privacy-based, additive, and multiplicative perturbation-based PPFL methods. These techniques balance data privacy and utility in FL scenarios [69].

EC20. Efficiency. This criterion helps to analyze the efficiency of the solutions presented in the primary studies as a critical aspect of the FC-BC-FL integration. It refers to the system's ability to perform tasks with minimal resource consumption and optimal performance. The following subcriteria support the analysis of efficiency (see Figure 11).

1) *Computational Efficiency (Learning Efficiency).* Refers to how effectively the integrated system performs model training using available computational resources [10], [39], [68].
2) *Cost Efficiency.* Examines the overall expenses associated with implementing and maintaining the integrated system compared to the benefits it provides [10], [42].
3) *Energy Efficiency.* Focuses on how well the system utilizes energy resources during model training and communication processes [68].
4) *Resource Efficiency.* Evaluates the utilization of resources such as bandwidth, memory, data sharing, and storage in the integrated system [10], [26], [68].
5) *Time Efficiency.* Assesses how quickly the system can complete tasks, enabling real-time or near-real-time decision-making [68].

EC21. Performance. This extraction criterion helps to analyze the performance of the solutions presented in the primary studies. Performance in the context of FC-BC-FL integration refers to how well the system functions and delivers results in terms of its overall effectiveness, speed, and accuracy [10], [68]. It encompasses several aspects related to the system's capabilities and achievements (see Figure 11), including:

1) *Accuracy.* The correctness and reliability of the model's predictions and results [68].
2) *Energy.* The amount of power or energy consumed during the integration process [68].
3) *Fault Tolerance.* The system's ability to maintain functionality despite failures or errors [68].
4) *Latency (Consensus Latency).* The time taken to reach a consensus in the BC network [68].
5) *Precision.* The level of detail and accuracy in model training and inference [39], [68].
6) *Resources.* The utilization and allocation of computing, memory, and network resources [39], [68].

EC22. Security. This extraction criterion assesses the security-related aspects of the solutions presented in the primary studies. Security in FC-BC-FL integration aims to protect the system from unauthorized access, data breaches, and malicious attacks. It guarantees data confidentiality, integrity, and availability during collaborative model training, secures the BC network and smart contracts, and prevents data tampering and unauthorized modifications. Trusted entities are permitted to participate in FL, ensuring controlled access to sensitive information [68] (see Figure 11).

1) *Security Concern.* While traditional FC systems face significant security vulnerabilities due to their location between end devices and cloud data centers [54], in the FC-BC-FL integration, security concerns aim to protect the system from unauthorized access, ensure data confidentiality and integrity, prevent disruptions in availability, and safeguard sensitive information during collaborative model training [17], [54], [68]. Existing studies may address security concerns related to:
   a) *Availability.* Ensuring that the system remains accessible and functional to entitled users [17], [68].
   b) *Privacy.* To Implement privacy-preserving mechanisms to protect individual data while allowing collaborative model training in a federated learning environment [17], [54].
   c) *Encryption.* Applying strong encryption techniques to protect data during transmission and storage, preventing unauthorized access to sensitive information [17], [54].
   d) *Vulnerability.* Address potential weaknesses in the embedded system to mitigate attacks that may disrupt system functionality [17], [54], [68].
   e) *Fraud Detection.* Detecting and preventing fraudulent activities within the integrated system [68].
   f) *Security Support.* Implementing measures to support secure and reliable operations [68].
   g) *Confidentiality.* Safeguarding sensitive data from unauthorized access or disclosure [42], [68].
   h) *ntegrity.* Ensuring the accuracy and consistency of data and models [42], [68].
   i) *Robustness.* Enhancing the system's resilience to withstand attacks, failures, or adverse conditions, keeping its operation and security [17], [54], [68].
2) *Physical Security.* In the context of integrating FC, BC, and FL, physical security entails protecting hardware (e.g., FC, BC nodes), against tampering, theft, and unauthorized access. Physical security encompasses access control, secure storage, tamper-evident mechanisms, electrical/electronic connections, and others.

3) *Physical Security Beneficiary.* In the context of presenting physical security in the primary study, the beneficiaries of security measures can be the edge server, cloud server, end devices, miners, or none.
4) *Cyber Security Beneficiary.* Cybersecurity defends against digital threats, ensuring data confidentiality, integrity, and availability during collaborative model training [17], [23], [68]. Since the subsequent extraction criteria analyze specific cyber-security concerns, this extraction criterion classifies the beneficiaries of the implemented cyber security within the solutions:
    a) *Network Security.* Securing the communication channels and infrastructure to prevent unauthorized access and data breaches.
    b) *Endpoint Security.* Securing individual devices and endpoints (e.g., computers, mobile devices) to prevent malware and unauthorized access.
    c) *Data Security.* Protecting sensitive data from unauthorized access, modification, or theft, both during transmission and storage.
    d) *Application Security.* Securing software applications and systems from vulnerabilities and exploits to prevent potential cyber-attacks.
    e) *Cloud Security.* Securing data and cloud-hosted applications hosted in cloud environments, to prevent data breaches and unauthorized access.
5) *Type of Attacks.* An attack represents deliberate and malicious actions or attempts to exploit vulnerabilities or weaknesses in a system, network, or application. The primary goal of attacks is to gain unauthorized access, disrupt normal operations, steal sensitive information, manipulate data, or cause harm to the targeted entity [69]. Below are classified the general attack types that could be part of FC-BC-FL integration developments:
    a) *Passive Attack.* Are attempts to observe computations and data during collaborative model training without directly altering the system [69].
    b) *Active Attack.* Are deliberate actions to influence the training process and manipulate model parameters to achieve adversarial objectives [69].
6) *Attacks.* This criterion classifies the possible attacks analyzed within the primary studies. The classified attacks are common in the areas of FL and BC.
    a) *Background Knowledge Attack.* It is a major privacy-oriented attack originating from the device's local data and global model updates received from the central authority. In FL, this attack exploits data and global model updates to potentially leak privacy information. Collusion attacks, a specific form of this attack, involve multiple parties sharing knowledge, and disclosing more sensitive information due to the freedom of devices to join and leave FL systems [25].
    b) *Byzantine Attacks.* These attacks disrupt FL system model convergence. A resilience strategy employs stochastic quantization, outlier detection, and secure model aggregation. Large-scale FL is susceptible due to diverse, low-powered edge devices, making prevention challenging [23], [25].
    c) *Poisoning Attacks.* Are significant threats that aim to manipulate training data and global models to mislead the learning output [25]. They cover data and model Poisoning attacks [18], [23], [41], [42].
    d) *Model Poisoning Attacks.* In FL, these attacks aim to compromise the global model directly by manipulating its updates or learning rules, often proving more effective than data poisoning. These attacks employ techniques like gradient manipulation or altering training rules to undermine the global model's performance. Preventing model poisoning in FL is challenging given the large number of participants, requiring innovative defense strategies for effective detection and mitigation of sophisticated attacks [18], [23], [41], [42].
    e) *Data Poisoning Attacks.* In FL, these attacks compromise the integrity of training data to mislead the model's performance such as label flipping, watermarking, perturbation, and backdoor insertion, which intentionally fool the model and reduce overall accuracy. These attacks call for urgent solutions to protect the global model from being poisoned by malicious participants, requiring careful defense strategies like FoolsGold and model evaluation to detect and mitigate Sybil-based data poisoning attacks [18], [23], [41], [42].
    f) *Evasion Attacks.* These attacks deceive the target model with adversarial samples during prediction, causing inaccurate results when the global model is deployed on end devices [23].
    g) *Free-riding Attacks.* Attacks where participants benefit from the global model without actively contributing to training. This behavior can undermine system fairness and efficiency, and various approaches, including BC-based solutions, aim to detect and mitigate such attacks [18], [23], [42].
    h) *Inference Attacks.* In FL, these attacks aim to extract sensitive information about participants, training data, and labels; compromising privacy and impacting performance. They include membership inference attacks (e.g., Confidence score-based, label-based), data properties inference attacks, data samples attacks, labels inference attacks, and model inversion attacks (e.g., Class representation) [23], [25], [41], [42], [69].
    i) *Distributed Denial of Service (DDoS).* In large-scale FL systems, DDoS involves thousands of edge devices participating in the learning task, potentially causing communication channel over-occupation and computational resource overload. This can lead to high latency, physical failure of infrastructure, and denial of service [25].

7) *Type of Attackers.* This criterion describes the potential adversaries in the FC-BC-FL integration. Including:
    a) *Insiders.* These attackers, including clients and the central server responsible for model aggregation in FL (i.e., malicious clients and servers), can gain access to intermediate training updates and the final model, posing privacy and information inference risks [42], [69].
    b) *Outsiders.* This type of attacker are outsider who can probe private data through the final model or query results (e.g., consumers and eavesdroppers). Moreover, these attackers can exploit access to the final model, while eavesdroppers may steal intermediate training updates, both causing significant privacy risks [42], [69].
8) *Moment of Attacks.* This extraction criterion defines the phases where the attacks occur. These moments are:
    a) *Training phase.* In this phase, model updates are vulnerable to privacy leakage as adversaries can access local gradients, model weights, aggregated updates, and the final model, with eavesdroppers intercepting communication between participants and the aggregator, posing additional risks [69].
    b) *Inference Phase.* In this phase, privacy risks primarily arise from the released final model, enabling attacks based on model parameters and queries. Adversaries can perform inference attacks to extract sensitive information about training datasets or infer membership [69].
9) *Attack Target.* This extraction criterion, Attack Target, describes the specific targets where attacks can occur, which encompass weight updates, gradient updates, and trained models [69].
10) *Attack Reason.* The attacks aim to compromise privacy and extract valuable information about the training data. Potential triggers for these attacks include:
    a) *Inferring class representative.* Generate representative samples for studying training datasets [69].
    b) *Inferring membership.* Determine if a data sample was used for model training [42], [69].
    c) *Inferring properties of private training data.* Gain information about the datasets [69].
    d) *Inferring training input and labels.* Reconstruct the original data and labels [69].
11) *Defense Methods against attacks.* Defines and classifies several methods that could have been considered as methods against attacks. These include:
    a) *Differential Privacy (DP).* These methods aim to enhance data privacy by injecting noise into input data, making it difficult to distinguish individual entries with a high degree of certainty. DP safeguards FL model parameters against information leakage and defends against attacks such as backdoor attacks while maintaining privacy in federated analysis models. DP mechanisms strike a balance between preserving data privacy and maintaining accuracy during FL [41], [67].
    b) *Sniper.* A defense against distributed poisoning attacks in FL, able to recognize legitimate users and reduce poisoning attack success rates even with multiple attackers [42].
    c) *Knowledge distillation.* A model compression technique sharing knowledge instead of model parameters to enhance FL client data security [42].
    d) *Anomaly detection.* Utilizes statistical methods to identify deviations from normal behavior in FL, useful for detecting various attacks such as data poisoning and trojan threats [42].
    e) *Moving target defense.* A proactive defense architecture that continuously changes to increase the cost and complexity for attackers, protecting against intrusion at different levels in FL [42].
    f) *Federated MultiTask Learning.* Extends FL to collaboratively train personalized but shared models among devices, addressing communication cost and fault tolerance challenges [42].
    g) *Data Sanitization.* Filters out suspicious data points as an anomaly detection technique to defend against data poisoning attacks in FL [42].
    h) *Foolsgold.* A defense against compromised clients in FL, efficient against Sybil-based, label flipping, and backdoor poisoning attacks [42].
    i) *Prunning.* A technique to minimize the size of ML models in FL, reducing complexity and improving accuracy to address communication and computation limitations on client devices [42].
    j) *Homomorphic encryption (HE).* This method enables mathematical operations on encrypted data without decryption. The output remains encrypted and decrypting it yields the result of the operations on the plaintext [67].
    k) *Outlier Detection.* This method is a defense strategy that aims to identify and deny malicious influence. Approaches such as “reject on negative impact” measure test error and reject updates that don’t improve the global model; or the TRIM method removes outliers with high residuals to minimize global objective loss, proving effective against various poisoning attacks [41].
    l) *Robust Aggregation.* It involves combining model updates from multiple participants while addressing challenges like noisy or malicious data, network instabilities, and attacks. The goal is to maintain accuracy and reliability by mitigating outliers and adversarial behaviors, ensuring the collaborative learning process integrity [41].
    m) *Secure Multi-Party Computation (MPC).* It is a cryptographic protocol enabling users to perform computations with private inputs. Data owners send encrypted data to servers for model training or analysis [42], [67].

n) *Secure Aggregation (SecAgg) protocol.* This method uses secure MPC, allowing untrusted parties to evaluate functions on hidden inputs. Trusted execution environments (e.g., Intel SGX) are used to protect computations within secure enclaves. SecAgg protocol uses MPC and involves data owners performing local training and sending encrypted model weights to the aggregator for gradient aggregation. [41], [67].

o) *The trusted execution environment (TEE).* These are trustable computational environments that ensure the code and data security within them. It can enhance the central server credibility [42], [67].

EC23. Interoperability (Standards). Standards are foundational documents that offer guidance across various domains, ensuring optimal performance and simplifying the use of information technology. They provide precise specifications for system interactions, like network protocols, and conceptual blueprints for software development, such as software architecture. Adhering to standards enhances interoperability, security, and efficiency, promoting seamless information exchange and technology adoption based on common formats and criteria [70]. This criterion assesses whether primary studies integrate standardization in their FC-BC-FL integration. Subcategories encompass prominent standard organizations: ISO Standards provide international guidelines for optimal performance in defined scopes [71]. IEEE Standards ensure consistency and interoperability in various technology fields [55]. NIST Standards offer guidance in cybersecurity, technology, and measurement [72]. Consider Domain Specific Languages (DSL), is relevant due to their specialization for specific application domains, streamlining tasks, and improving efficiency, although achieving broader interoperability often requires standardization [73]. Additional standards and guidelines are included in Others (see Figure 11).

EC24. Scalability. When integrating FC-BC-FL, scalability issues arise due to the different mechanisms in each technology. For instance, some solutions have limitations in scalability and power requirements, while others sacrifice security, privacy, or decentralization [68]. This extraction criterion helps in identifying key scalability parameters that could have been addressed in the primary studies to improve the scalability of these integrated systems (see Figure 11):

1) *Scalability Support.* It refers to implementing architectures and techniques to address scalability challenges arising from diverse mechanisms in each technology. This involves designing systems to efficiently manage numerous devices and transactions, improving real-world performance (e.g., via smart contracts, secure data management platforms, integrated frameworks, distributed SDN controllers, and scalable public BC with two-chain structures) [68].
2) *Mobility.* It refers to the ability to transfer data and perform tasks efficiently and securely across mobile and distributed devices [68].
3) *Regulations/Standards.* It refers to the use of BC and SCs to establish transparent rules for transactions and IoT devices, ensuring compliance and secure resource management [68]. It also involves setting guidelines and policies to govern the sharing and privacy of data among participating devices or clients in FL to ensure compliance with legal and ethical requirements.

EC25. Data Management. FC-BC-FL integration leads to data management issues arising due to the differences in the mechanisms of handling data, as well as the heterogeneity and distributed nature of IoT devices/nodes in each technology [54]. This criterion helps in identifying key data management parameters that could have been addressed in the primary studies to improve these integrated systems.

1) *Storage.* In the FC-BC-FL integration, storage refers to the management of data from IoT devices. Transitory fog storage allows fast data model updates. Besides, various protocols and techniques, including BC capability, regeneration coding, and encryption, ensure secure and efficient data storage [54].
2) *Sharing.* It involves securely and efficiently sharing data among participants, ensuring trust, and maintaining integrity through methods such as consensus and encryption. Techniques like storing hash values in blockchain and cross-chain sharing ensure reliable data exchange in diverse IoT/FC systems [50], [54].
3) *Validation.* It ensures data accuracy, integrity, and authenticity before storage or access. Techniques like digital signatures, smart contracts, and timestamping on the BC ledger ensure secure validation in the FL process, maintaining data integrity across entities. These mechanisms preserve privacy, security, and performance in the BC-FC integration [52], [54].

EC26. Resource Allocation / Service provisioning metrics. Resource allocation and service provisioning metrics in FC-BC-FL integration are performance indicators used to assess the efficient utilization of resources for executing tasks (e.g., FL tasks) [68]. They ensure sufficient resource allocation, optimize system performance, and leverage the combined capabilities of BC, FC, and FL technologies. This extraction criterion categorizes various resource allocation and service provisioning metrics that could have been addressed in the primary studies, considering several domains:

1) *Time.* Time is a key metric for resource allocation and service provisioning in FC-BC-FL integration, measuring task efficiency, response times, and system performance. It evaluates the timely allocation of resources to meet application and service demands [54]. There are sub-criteria to consider:
   a) *Communication.* It indicates the time for exchanging model updates or data elements between the central server and clients or between FC/Mining nodes, impacting training efficiency, communication overhead, and node selection [54].

  b) *Computation.* Measures task execution efficiency, helping in resource and power management [54].
  c) *Deadline.* Set the maximum service delivery delay, and detail latency-sensitive applications [54].
  Below, are described some BC-FL Time metrics.
  d) *Training Time.* It is the time taken to train the ML models in the FL process. This metric indicates the efficiency of the training process across multiple clients or nodes in a federated environment.
  e) *Consensus Time.* In BC-based systems, the time required to reach consensus on new transactions or blocks. This metric affects the speed and scalability of the BC network.
  f) *Block Time.* The time interval between the creation of consecutive blocks in a BC. This metric determines the speed at which new transactions are added to the BC.
  g) *Transaction Time.* The time taken to process and validate a single transaction in the BC network. This metric influences the responsiveness and efficiency of the BC system.
  h) *Confirmation Time.* In BC networks using consensus mechanisms (e.g., PoW), the confirmation time for new block validity impacts transaction security and finality.

2) *Data.* The metrics concerning data are:

  a) *Data Flow.* It defines data transmission patterns (event-driven or real-time) and influences resource allocation and service provisioning [54].
  b) *Data Size.* The volume of data processed through BC, FC, or FL, that impacts resource provisioning and computational space requirements [54].
  Below, are described some BC-FL Data metrics.
  c) *Data Distribution.* Describes the distribution of data across participating clients, which can impact the model's performance and convergence.
  d) *Data Privacy.* Measures the level of privacy protection during data sharing and model updates to ensure compliance with privacy regulations and prevent data leakage.
  e) *Data Heterogeneity.* Evaluates the diversity and variation of data among clients, affecting the generalization and robustness of the FL model.
  f) *Data Immutability.* Refers to the property of data being tamper-resistant once recorded on the BC, ensuring the integrity and trustworthiness of data.
  g) *Data Transparency.* Measures the degree of visibility and accessibility of data stored on the BC to promote accountability and auditability.
  h) *Data Storage Efficiency.* Evaluates the efficiency of storing data on the BC, considering factors like data size, storage cost, and scalability to optimize resource utilization.

3) *Cost.* The metrics about cost-impacting resource and service provisioning in the FC-BC-FL integration are:

  a) *Networking Cost.* It includes bandwidth expenses, uploading, and inter-nodal data sharing costs [54].
  b) *Deployment Cost.* Refers to infrastructure placement expenses, considering Fog/Mining node positioning and virtual computing instances [54].
  c) *Execution Cost.* It refers to the computational expenses of Fog/Mining nodes while processing tasks, calculated based on task completion time and resource usage cost [54].
  Below, are considered some BC-FL Cost metrics.
  d) *Communication Cost.* The expenses incurred in transmitting model updates or gradients between the central server and individual clients during the FL process [42].
  e) *Computation Cost.* The computational expenses involved in training and updating the models on the client devices during the FL process [42].
  f) *Transaction Fee.* The cost paid by users for each transaction executed on the BC network or FL training. This fee is essential to incentivize miners or validators to include the transaction in a block and secure the network (i.e., FL or BC Incentives).

4) *Context.* It refers to the situation or condition of entities in various circumstances. They include:

  a) *User Context.* It comprises features, usage history, and feedback, impacting resource allocation [54].
  b) *Application Context.* Enfolds operational requirements like processing and networking needs [54].

5) *Energy Consumption.* It is a crucial concern in FC-BC-FL integration. Studies prioritize energy-related issues for FC-BC-FL resources and services provisioning, optimizing consumption, and considering end devices' energy constraints [54].

EC27. Service Level Objectives (SLOs). An SLO is a measurable performance goal set to ensure the quality and reliability of a service. It represents the desired level of service performance and is used to monitor and maintain service quality. Meeting SLOs is essential for delivering a satisfactory user experience [54]. The SLOs include (see Figure 11).

1) *Latency management.* To optimize communication between FC nodes, BC ledger, and FL clients, minimizing service delivery time and achieving low latency while meeting QoS requirements [54].
2) *Cost management.* It is to strategically deploy Fog nodes and utilize cost-effective FC-BC-FL infrastructure configurations to minimize costs in Fog nodes/miners for resource hosting, ensuring cost-efficient provisioning of resources and services [54].
3) *Network management.* Enabling flexible, virtualized network structures to ensure seamless connectivity among FNs, BC miners, and FL clients. Designing an architecture for efficient resource discovery and communication across FC-BC-FL environments, particularly catering to the highly distributed IoT devices. This

entails defining SLOs targeting Congestion, Virtualization, and Connectivity [54].

4) *Computation management.* It entails SLOs oriented to:
   a) *Resource estimation.* Optimizing resource allocation in FC-BC-FL integration is vital, considering user characteristics, Quality of Experience (QoE), and device features. This ensures efficient resource allocation for FL tasks, maintaining the desired Quality of Service (QoS) and accurately determining service prices [54].
   b) *Workload allocation.* It focuses on optimizing resource utilization, minimizing idle periods, and ensuring balanced load distribution among various components. Effective workload allocation involves distributing computational tasks efficiently among Fog nodes, clients, and the BC network. Scheduling-based policies play a key role in achieving these objectives and improving the overall QoE in the system [54].
   c) *Coordination.* Efficient Fog resource coordination is vital for FL. Using a directed graph-based model optimizes communication and computation, ensuring effective resource use across Fog nodes and federated entities. [54].
5) *Application management.* It entails SLOs oriented to:
   a) *Programming platform.* Platforms using simplified programming models for large-scale applications in Fog computing. A distributed data flow platform facilitates application development. In FC-BC-FL integration, an efficient and compatible programming platform is vital [54].
   b) *Scaling.* Are scalable techniques to optimize resource utilization and enhance QoE for BC mining and FL tasks [54].
   c) *Offloading.* It distributes tasks efficiently among federated entities, considering resource availability and improving FL performance [54].
6) *Data management.* It includes data management, such as data analytics, resource allocation, and low-latency data aggregation, in the SLOs of the studies.
7) *Power management.* It has power consumption considerations (e.g., miner node energy, cloud energy) as part of the SLOs in the studies [54].

EC28. Trust. Represent trust connections between nodes, enabling entities to trust each other for specific activities. Its primary functions are to establish an entity's trustworthiness for others and assess the trustworthiness of other entities. However, trust management can be energy-intensive, which poses challenges for resource-constrained IoT devices [68]. It outlines sub-criteria related to trust (see Figure 11).

1) *Trust support.* Mechanisms enabling secure interactions and cooperation among entities in FC-BC-FL integration [68].
2) *QoS.* It is the level of performance and reliability in FC-BC-FL operations [68].
3) *Transparency.* Are visible decision-making processes and data-handling practices for trust-building [68].
4) *Reliability* It is the Consistency and dependability of accurate results in the integrated system [68].
5) *Reputation.* It is the Assessment of entity trustworthiness based on past behaviors in FC-BC-FL [68].
6) *Payment.* It is the fair and secure handling of financial transactions and incentives among participants [68].

EC29. Resilience (Proper of BC). It refers to the capacity to withstand and recover from challenges and disruptions, adapting and maintaining functionality in adverse situations. It involves proactive planning and the ability to bounce back quickly after setbacks [34]. This extraction criterion classifies a resilience-related sub-criterion in BC (see Figure 11):

1) *Forking*
   a) *Soft Forks.* Backwards-compatible changes to a BC implementation, allowing non-updated nodes to still transact with updated nodes [34].
   b) *Hard Forks.* Non-backwards-compatible changes to a blockchain implementation require all nodes to switch to the updated protocol, resulting in the creation of two independent BC versions [34].

EC30. Access Control. This extraction criterion aids in the analysis of access control, involving the use of countermeasures and tactics to secure access to data [68]. The following sub-criteria assist in addressing this criterion (see Figure 11):

1) *Authorization.* It ensures access for authenticated users, using BC to enable distributed processes among fog nodes. It enhances data sharing, integrity, and security, mitigating centralized storage concerns, and improves privacy for IoT devices through SCs [26], [68].
2) *Key Management.* It involves cryptographic procedures to protect data, requiring encryption and access control. BC-based schemes manage secure keys, enable mutual authentication, and provide efficient key management for secure group channels in Fog-based IoT systems. BC is utilized to ensure data integrity, and message security, and detect malicious nodes in decentralized key management frameworks [68].
3) *Authentication.* It ensures user identity verification and prevents fraudulent communications. Integrated BC security models provide privacy and authentication, enhancing security and privacy for distributed vehicular fog services and smart vehicle systems in FC [68].

EC31. Heterogeneity. Refers to the diverse elements or entities within a system. In integrating FC-BC-FL, managing diverse environments requires seamless interactions and interoperability. This criterion includes sub-criteria such as data format, device, protocol, and network heterogeneity.

EC32. Model (Local / Global). The models are critical elements within FC-BC-FL integration; therefore, they require precision and security to ensure success [39]. This extraction criterion categorizes two important features that could have been addressed in the primary studies to guarantee the effectiveness of the models (see Figure 11). These are:

1) *Model Precision.* It is the accuracy and correctness of locally trained models on FNs ensure they faithfully represent their datasets. In global FL, this extends to the accuracy of the aggregated global model, encapsulating the collective knowledge of all local models [17], [20].
2) *Model Verification.* It assures the authenticity and security of FL models. BC verifies model integrity by storing cryptographic hashes, ensuring reliability and detecting tampering or malicious activities [20], [39].

EC33. Other Considerations. This extraction criterion categorizes additional aspects that could have been evaluated in the primary studies regarding the integration of BC, FC, and FL to enhance IoT applications (see Figure 11):

1) *Economical Considerations.* Entails assessing the financial aspects and cost implications of integrating these technologies, analyzing expenses, cost-effectiveness, and Return on Investment (ROI), and optimizing resource allocation for maximal benefits.
2) *User Acceptance.* Assesses stakeholders' preparedness for the integrated system, addressing user expectations, handling resistance to change, ensuring user-friendly interfaces, and collecting feedback for enhancements.
3) *Environmental Impacts.* Evaluate the integration's environmental impact from energy use, carbon footprint, and resource consumption, while backing sustainability via energy-efficient and eco-friendly methods.

The upcoming criteria (Figure 12) align with SRQ3, aiding in characterizing research phases, validations, relevance, and key aspects, directly extracted from primary studies.

EC34. Phases. In traditional computer science, the standard life cycle includes key phases like analysis, design, implementation, and testing. In an FC-BC-FL integration project, these phases are used as reference points to assess the project's progress. During the review, primary studies are categorized by the specific phase they address, such as Analysis, Design, Implementation, or Testing (see Figure 12).

*EC35. Type of validation.*Assessing research quality is vital across various scientific fields and study levels [74]. This criterion classifies the validation methods that primary studies may utilize to evaluate their solutions [75] (see Figure 12).

1) *Experiment.* Involves strict control and randomization of variables to establish causality.
2) *Quasi-Experiment.* Utilizes controlled variables for causality when full control is not feasible.
3) *Prototyping.* Creates an initial product version for design and functionality testing.
4) *Study Case.* Analyzes a specific case deeply to understand a phenomenon or problem.
5) *Surveys.* Collects data through structured questions to gauge trends or opinions.
6) *Proof of Concept.* Demonstrates idea feasibility, even if it is small or incomplete.
7) *Other*. Another type of validation.

EC36. Experiment Features (Simulation Configurations, Tools). This extraction criterion outlines experiment details in primary studies, including (see Figure 12):

1) *Type of Scenario.* Real or Simulation.

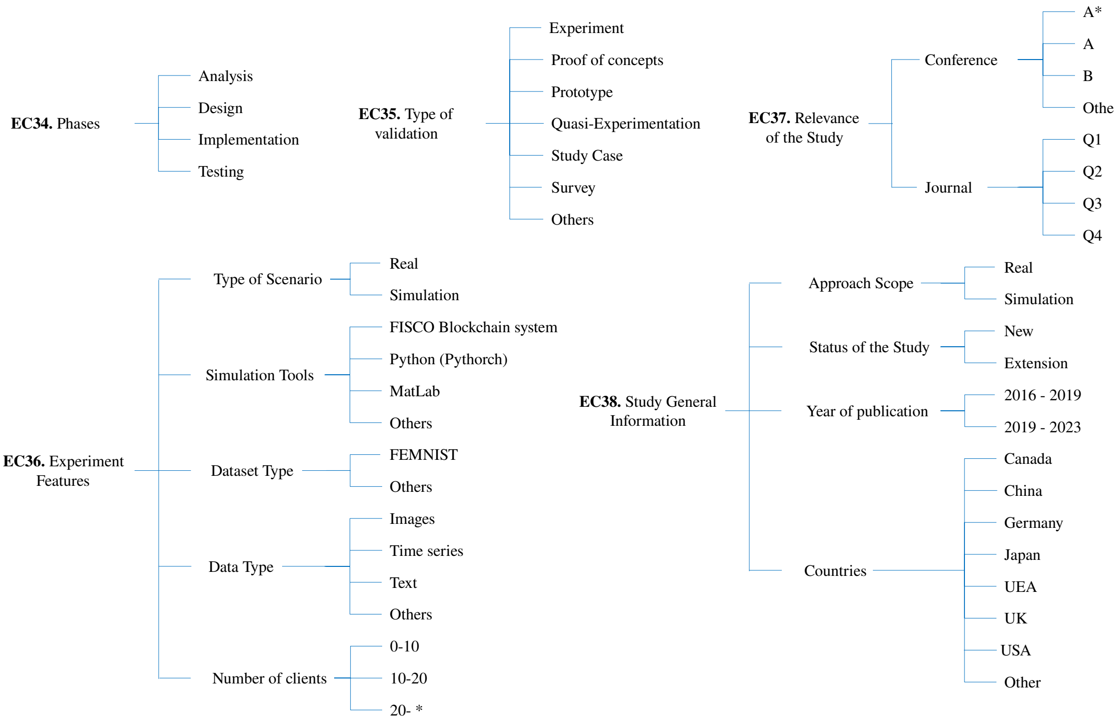


FIGURE 12. SRQ3 Extraction Criteria (EC34 - EC38)

2) *Simulation Tools.* The tools include FISCO Blockchain system, Python (PyTorch), MatLab, among others.
3) *Dataset Type.* There are classified utilized datasets to perform experiments [39] (e.g., FEMNIST).
4) *Data Type.* Types of data used in experiments, including images, time series, text, and more.
5) *Number of clients.* Ranging from 0-10, 10-20, to 20-* in real or simulated scenarios.".

EC37. Relevance of the study. It classifies the study relevance into two categories: Conference Rank (i.e., A*, A, B, Other) or Journal Rank (i.e., Q1, Q2, Q3, Q4) (see Figure 12).

EC38. Study General Information. Defines essential publication details for primary studies (see Figure 12).

1) *Approach scope.* Specifies whether the study was conducted or supported in academia or industry.
2) *Status of the study.* The study is new, or an extension.
3) *Year of Publication.* Starting from the milestone of the FL emergence. The years of publication are categorized into two groups: 2016 – 2019, and 2019 – 2023.
4) *Country.* Specifies the country where the study was conducted (not where it was published).

B. CONDUCTING THE REVIEW

The process for selecting primary studies is illustrated in Figure 13. Initially, 153 papers were retrieved through automatic search following the planning stage and search guidelines. Subsequently, the inclusion/exclusion criteria stated in the quality assessment were applied, resulting in the selection of 32 papers. Additionally, 8 papers were added through a manual search in specialized conferences and journals. Thus, 40 primary studies were selected for the SLR (see Table 4).

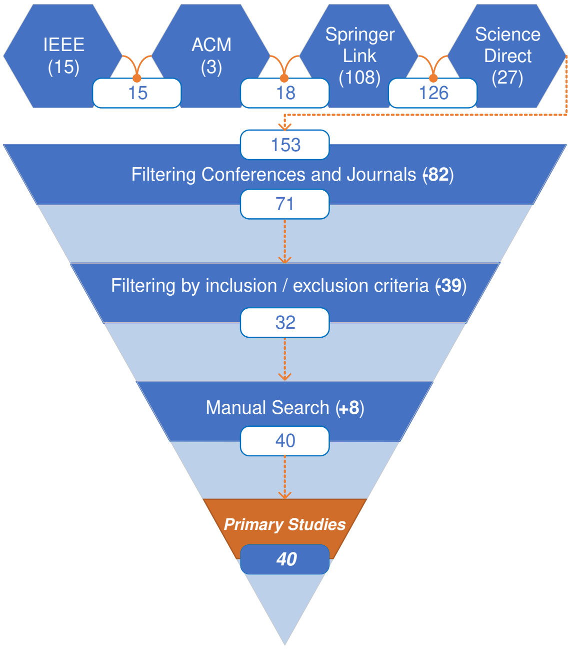


FIGURE 13. Selection of the primary studies for performing the SLR

TABLE 4. Selected primary studies addressing FC-BC-FL integration

Source	Paper	Year	Paper ID
IEEE	[39]	2021	P01
	[76]	2022	P02
	[77]	2020	P03
	[78]	2022	P04
	[79]	2022	P05
	[80]	2023	P06
	[81]	2023	P07
	[82]	2020	P08
	[83]	2022	P09
	[84]	2023	P10
	[85]	2022	P11
ACM	[86]	2020	P12
	[87]	2021	P13
Springer Link	[88]	2021	P14
	[89]	2023	P15
	[90]	2020	P16
	[91]	2023	P17
	[92]	2023	P18
Science Direct	[93]	2022	P19
	[94]	2022	P20
	[95]	2022	P21
	[96]	2020	P22
	[97]	2022	P23
	[98]	2023	P24
	[99]	2023	P25
	[100]	2021	P26
	[101]	2023	P27
	[102]	2022	P28
	[103]	2023	P29
	[104]	2023	P30
	[105]	2023	P31
	[106]	2023	P32
Conference	[107]	2023	P33
Journal	[108]	2022	P34
	[109]	2022	P35
	[110]	2023	P36
	[111]	2020	P37
	[112]	2023	P38
	[113]	2022	P39
	[114]	2023	P40

IV. REPORTING THE SLR RESULTS

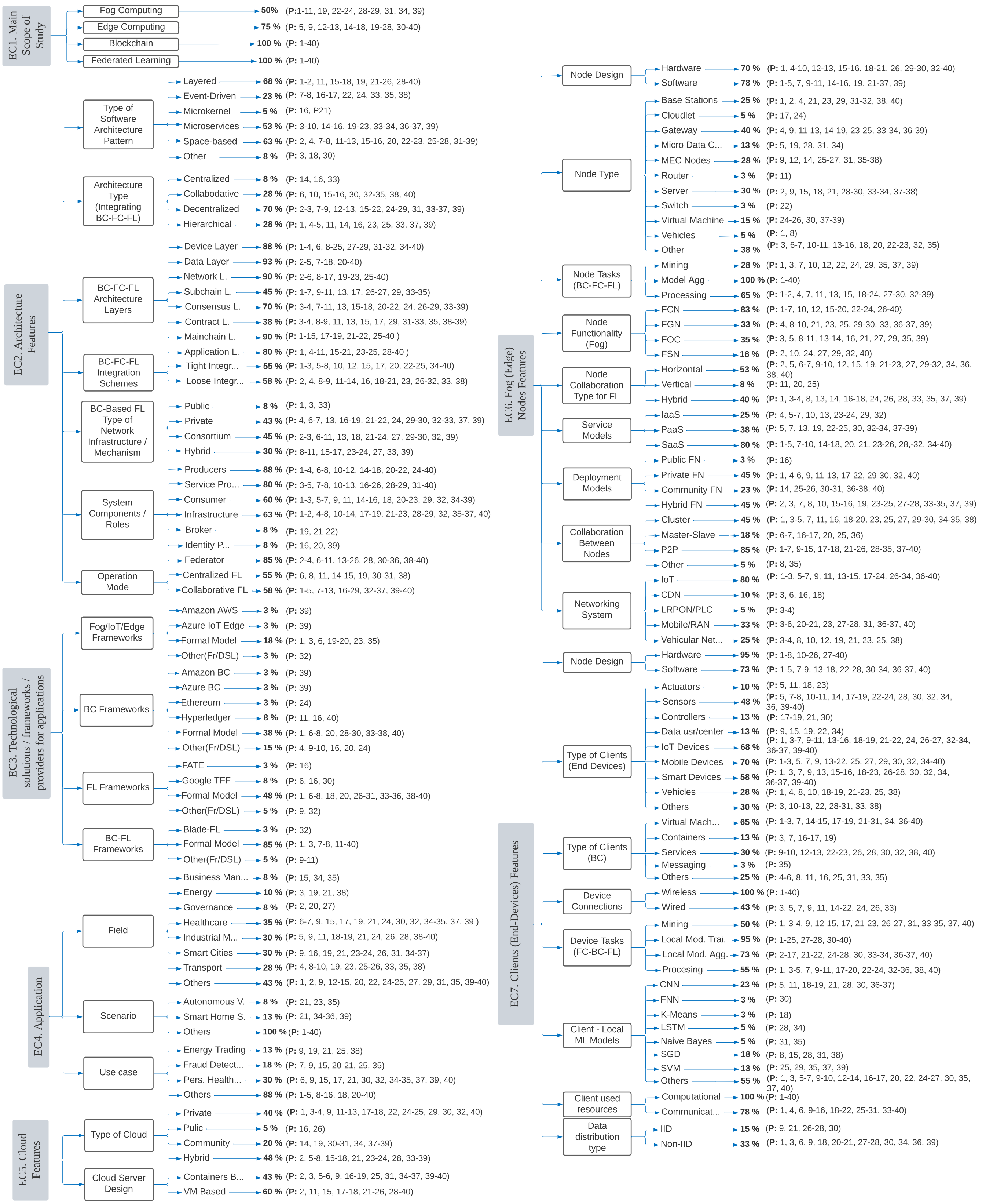
This section presents SLR results. The reporting is segmented into two parts: first, results per extraction criterion; second, findings concerning age and year of publication.

A. RESULTS PER EXTRACTION CRITERION

The results for each extraction criterion are presented and discussed in this sub-section. These findings are presented in percentages, corresponding to the number of studies out of 40 that cover or consider these criteria while integrating FC-BC-FL, grouped by sub-research question (i.e., SRQ1, SRQ2, SRQ3). All results with values greater than 0 are considered; however, those below this threshold have been excluded.

1) SRQ1 results

Figure 14 (a) and Figure 14 (b) display the outcomes of SRQ1, all of the selected studies focus on BC and FL technologies (100%). Additionally, 75% of these studies delve into the edge, while 50% explore research including FC. In this context, the primary motivations driving this integration encompass enhanced data privacy (highlighted in 93% of the studies)


EC1. Main Scope of Study
Fog Computing 50% (P:1-11, 19, 22-24, 28-29, 31, 34, 39)
Edge Computing 75 % (P: 5, 9, 12-13, 14-18, 19-28, 30-40)
Blockchain 100 % (P: 1-40)
Federated Learning 100 % (P: 1-40)
EC2. Architecture Features
Type of Software Architecture Pattern
Layered 68 % (P: 1-2, 11, 15-18, 19, 21-26, 28-40)
Event-Driven 23 % (P: 7-8, 16-17, 22, 24, 33, 35, 38)
Microkernel 5 % (P: 16, P21)
Microservices 53 % (P: 3-10, 14-16, 19-23, 33-34, 36-37, 39)
Space-based 63 % (P: 2, 4, 7-8, 11-13, 15-16, 20, 22-23, 25-28, 31-39)
Other 8 % (P: 3, 18, 30)
Architecture Type (Integrating BC-FC-FL)
Centralized 8 % (P: 14, 16, 33)
Collabodative 28 % (P: 6, 10, 15-16, 30, 32-35, 38, 40)
Decentralized 70 % (P: 2-3, 7-9, 12-13, 15-22, 24-29, 31, 33-37, 39)
Hierarchical 28 % (P: 1, 4-5, 11, 14, 16, 23, 25, 33, 37, 39)
BC-FC-FL Architecture Layers
Device Layer 88 % (P: 1-4, 6, 8-25, 27-29, 31-32, 34-40)
Data Layer 93 % (P: 2-5, 7-18, 20-40)
Network L. 90 % (P: 2-6, 8-17, 19-23, 25-40)
Subchain L. 45 % (P: 1-7, 9-11, 13, 17, 26-27, 29, 33-35)
Consensus L. 70 % (P: 3-4, 7-11, 13, 15-18, 20-22, 24, 26-29, 33-39)
Contract L. 38 % (P: 3-4, 8-9, 11, 13, 15, 17, 29, 31-33, 35, 38-39)
Mainchain L. 90 % (P: 1-15, 17-19, 21-22, 25-40)
Application L. 80 % (P: 1, 4-11, 15-21, 23-25, 28-40)
BC-FC-FL Integration Schemes
Tight Integr... 55 % (P: 1-3, 5-8, 10, 12, 15, 17, 20, 22-25, 34-40)
Loose Integr... 58 % (P: 2, 4, 8-9, 11, 16-21, 23-26, 32, 33, 38)
BC-Based FL Type of Network Infrastructure / Mechanism
Public 8 % (P: 1, 3, 33)
Private 43 % (P: 4, 6-7, 13, 16-19, 21-22, 24, 29-30, 32-33, 37, 39)
Consortium 45 % (P: 2-3, 6-11, 13, 18, 21-24, 27, 29-30, 32, 39)
Hybrid 30 % (P: 8-11, 15-17, 23-24, 27, 33, 39)
System Components / Roles
Producers 88 % (P: 1-4, 6-8, 10-12, 14-18, 20-22, 24-40)
Service Pro... 80 % (P: 3-5, 7-8, 10-13, 16-26, 28-29, 31-40)
Consumer 60 % (P: 1-3, 5-7, 9, 11, 14-16, 18, 20-23, 29, 32, 34-39)
Infrastructure 63 % (P: 1-2, 4-8, 10-14, 17-19, 21-23, 28-29, 32, 35-37, 40)
Broker 8 % (P: 19, 21-22)
Identity P... 8 % (P: 16, 20, 39)
Federator 85 % (P: 2-4, 6-11, 13-26, 28, 30-36, 38-40)
Operation Mode
Centralized FL 55 % (P: 6, 8, 11, 14-15, 19, 30-31, 38)
Collaborative FL 58 % (P: 1-5, 7-13, 16-29, 32-37, 39-40)
EC3. Technological solutions / frameworks / providers for applications
Fog/IoT/Edge Frameworks
Amazon AWS 3 % (P: 39)
Azure IoT Edge 3 % (P: 39)
Formal Model 18 % (P: 1, 3, 6, 19-20, 23, 35)
Other(Fr/DSL) 3 % (P: 32)
BC Frameworks
Amazon BC 3 % (P: 39)
Azure BC 3 % (P: 39)
Ethereum 3 % (P: 24)
Hyperledger 8 % (P: 11, 16, 40)
Formal Model 38 % (P: 1, 6-8, 20, 28-30, 33-38, 40)
Other(Fr/DSL) 15 % (P: 4, 9-10, 16, 20, 24)
FL Frameworks
FATE 3 % (P: 16)
Google TFF 8 % (P: 6, 16, 30)
Formal Model 48 % (P: 1, 6-8, 18, 20, 26-31, 33-36, 38-40)
Other(Fr/DSL) 5 % (P: 9, 32)
BC-FL Frameworks
Blade-FL 3 % (P: 32)
Formal Model 85 % (P: 1, 3, 7-8, 11-40)
Other(Fr/DSL) 5 % (P: 9-11)
EC4. Application
Field
Business Man... 8 % (P: 15, 34, 35)
Energy 10 % (P: 3, 19, 21, 38)
Governance 8 % (P: 2, 20, 27)
Healthcare 35 % (P: 6-7, 9, 15, 17, 19, 21, 24, 30, 32, 34-35, 37, 39)
Industrial M... 30 % (P: 5, 9, 11, 18-19, 21, 24, 26, 28, 38-40)
Smart Cities 30 % (P: 9, 16, 19, 21, 23-24, 26, 31, 34-37)
Transport 28 % (P: 4, 8-10, 19, 23, 25-26, 33, 35, 38)
Others 43 % (P: 1, 2, 9, 12-15, 20, 22, 24-25, 27, 29, 31, 35, 39-40)
Scenario
Autonomous V. 8 % (P: 21, 23, 35)
Smart Home S. 13 % (P: 21, 34-36, 39)
Others 100 % (P: 1-40)
Use case
Energy Trading 13 % (P: 9, 19, 21, 25, 38)
Fraud Detect... 18 % (P: 7, 9, 15, 20-21, 25, 35)
Pers. Health... 30 % (P: 6, 9, 15, 17, 21, 30, 32, 34-35, 37, 39, 40)
Others 88 % (P: 1-5, 8-16, 18, 20-40)
EC5. Cloud Features
Type of Cloud
Private 40 % (P: 1, 3-4, 9, 11-13, 17-18, 22, 24, 29, 30, 32, 40)
Pulic 5 % (P: 16, 26)
Community 20 % (P: 14, 19, 30-31, 34, 37-39)
Hybrid 48 % (P: 2, 5-8, 15-18, 21, 23-24, 28, 33-39)
Cloud Server Design
Containers B... 43 % (P: 2, 3, 5-6, 9, 16-19, 25, 31, 34-37, 39-40)
VM Based 60 % (P: 2, 11, 15, 17-18, 21-26, 28-40)
EC6. Fog (Edge) Nodes Features
Node Design
Hardware 70 % (P: 1, 4-10, 12-13, 15-16, 18-21, 26, 29-30, 32-40)
Software 78 % (P: 1-5, 7, 9-11, 14-17, 19, 21-37, 39)
Node Type
Base Stations 25 % (P: 1, 2, 4, 21, 23, 29, 31-32, 38, 40)
Cloudlet 5 % (P: 17, 24)
Gateway 40 % (P: 4, 9, 11-13, 14-19, 23-25, 33-34, 36-39)
Micro Data C... 13 % (P: 5, 19, 28, 31, 34)
MEC Nodes 28 % (P: 9, 12, 14, 25-27, 31, 35-38)
Router 3 % (P: 11)
Server 30 % (P: 2, 9, 15, 18, 21, 28-30, 33-34, 37-38)
Switch 3 % (P: 22)
Virtual Machine 15 % (P: 24-26, 30, 37-39)
Vehicles 5 % (P: 1, 8)
Other 38 % (P: 3, 6-7, 10-11, 13-16, 18, 20, 22-23, 32, 35)
Node Tasks (BC-FC-FL)
Mining 28 % (P: 1, 3, 7, 10, 12, 22, 24, 29, 35, 37, 39)
Model Agg 100 % (P: 1-40)
Processing 65 % (P: 1-2, 4, 7, 11, 13, 15, 18-24, 27-30, 32-39)
Node Functionality (Fog)
FCN 83 % (P: 1-7, 10, 12, 15-20, 22-24, 26-40)
FGN 33 % (P: 4, 8-10, 21, 23-25, 29-30, 33, 36-37, 39)
FOC 35 % (P: 3, 5, 8-11, 13-14, 16, 21, 27, 29, 35, 39)
FSN 18 % (P: 2, 10, 24, 27, 29, 32, 40)
Node Collaboration Type for FL
Horizontal 53 % (P: 2, 5, 6-7, 9-10, 12, 15, 19, 21-23, 27, 29-32, 34, 36, 38, 40)
Vertical 8 % (P: 11, 20, 25)
Hybrid 40 % (P: 1, 3-4, 8, 13, 14, 16-18, 24, 26, 28, 33, 35, 37, 39)
Service Models
IaaS 25 % (P: 4, 5-7, 10, 13, 23-24, 29, 32)
PaaS 38 % (P: 5, 7, 13, 19, 22-25, 30, 32-34, 37-39)
SaaS 80 % (P: 1-5, 7-10, 14-18, 20, 23-26, 28,-32, 34-40)
Deployment Models
Public FN 3 % (P: 16)
Private FN 45 % (P: 1, 4-6, 9, 11-13, 17-22, 29-30, 32, 40)
Community FN 23 % (P: 14, 25-26, 30-31, 36-38, 40)
Hybrid FN 45 % (P: 2, 3, 7, 8, 10, 15-16, 19, 23-25, 27-28, 33-35, 37, 39)
Collaboration Between Nodes
Cluster 45 % (P: 1, 3-5, 7, 11, 16, 18-20, 23, 25, 27, 29-30, 34-35, 38)
Master-Slave 18 % (P: 6-7, 16-17, 20, 25, 36)
P2P 85 % (P: 1-7, 9-15, 17-18, 21-26, 28-35, 37-40)
Other 5 % (P: 8, 35)
Networking System
IoT 80 % (P: 1-3, 5-7, 9, 11, 13-15, 17-24, 26-34, 36-40)
CDN 10 % (P: 3, 6, 16, 18)
LRPON/PLC 5 % (P: 3-4)
Mobile/RAN 33 % (P: 3-6, 20-21, 23, 27-28, 31, 36-37, 40)
Vehicular Net... 25 % (P: 3-4, 8, 10, 12, 19, 21, 23, 25, 38)
EC7. Clients (End-Devices) Features
Node Design
Hardware 95 % (P: 1-8, 10-26, 27-40)
Software 73 % (P: 1-5, 7-9, 13-18, 22-28, 30-34, 37, 40)
Type of Clients (End Devices)
Actuators 10 % (P: 5, 11, 18, 23)
Sensors 48 % (P: 5, 7-8, 10-11, 14, 17-19, 22-24, 28, 30, 32, 34, 36, 39-40)
Controllers 13 % (P: 17-19, 21, 30)
Data usr/center 13 % (P: 9, 15, 19, 22, 34)
IoT Devices 68 % (P: 1, 3-7, 9-11, 13-16, 18-19, 21-22, 24, 26-27, 32-34, 36-37, 39-40)
Mobile Devices 70 % (P: 1-3, 5, 7, 9, 13-22, 25, 27, 29, 30, 32, 34-40)
Smart Devices 58 % (P: 1, 3, 7, 9, 13, 15-16, 18-23, 26-28, 30, 32, 34, 36-37, 39-40)
Vehicles 28 % (P: 1, 4, 8, 10, 18-19, 21-23, 25, 38)
Others 30 % (P: 3, 10-13, 22, 28-31, 33, 38)
Type of Clients (BC)
Virtual Mach... 65 % (P: 1-3, 7, 14-15, 17-19, 21-31, 34, 36-40)
Containers 13 % (P: 3, 7, 16-17, 19)
Services 30 % (P: 9-10, 12-13, 22-23, 26, 28, 30, 32, 38, 40)
Messaging 3 % (P: 35)
Others 25 % (P: 4-6, 8, 11, 16, 25, 31, 33, 35)
Device Connections
Wireless 100 % (P: 1-40)
Wired 43 % (P: 3, 5, 7, 9, 11, 14-22, 24, 26, 33)
Device Tasks (FC-BC-FL)
Mining 50 % (P: 1, 3-4, 9, 12-15, 17, 21-23, 26-27, 31, 33-35, 37, 40)
Local Mod. Trai. 95 % (P: 1-25, 27-28, 30-40)
Local Mod. Agg. 73 % (P: 2-17, 21-22, 24-28, 30, 33-34, 36-37, 40)
Procesing 55 % (P: 1, 3-5, 7, 9-11, 17-20, 22-24, 32-36, 38, 40)
Client - Local ML Models
CNN 23 % (P: 5, 11, 18-19, 21, 28, 30, 36-37)
FNN 3 % (P: 30)
K-Means 3 % (P: 18)
LSTM 5 % (P: 28, 34)
Naive Bayes 5 % (P: 31, 35)
SGD 18 % (P: 8, 15, 28, 31, 38)
SVM 13 % (P: 25, 29, 35, 37, 39)
Others 55 % (P: 1, 3, 5-7, 9-10, 12-14, 16-17, 20, 22, 24-27, 30, 35, 37, 40)
Client used resources
Computational 100 % (P: 1-40)
Communicat... 78 % (P: 1, 4, 6, 9-16, 18-22, 25-31, 33-40)
Data distribution type
IID 15 % (P: 9, 21, 26-28, 30)
Non-IID 33 % (P: 1, 3, 6, 9, 18, 20-21, 27-28, 30, 33, 34, 36, 39)

(a)

FIGURE 14. Results per extraction criterion, SRQ1 (EC1-EC7)

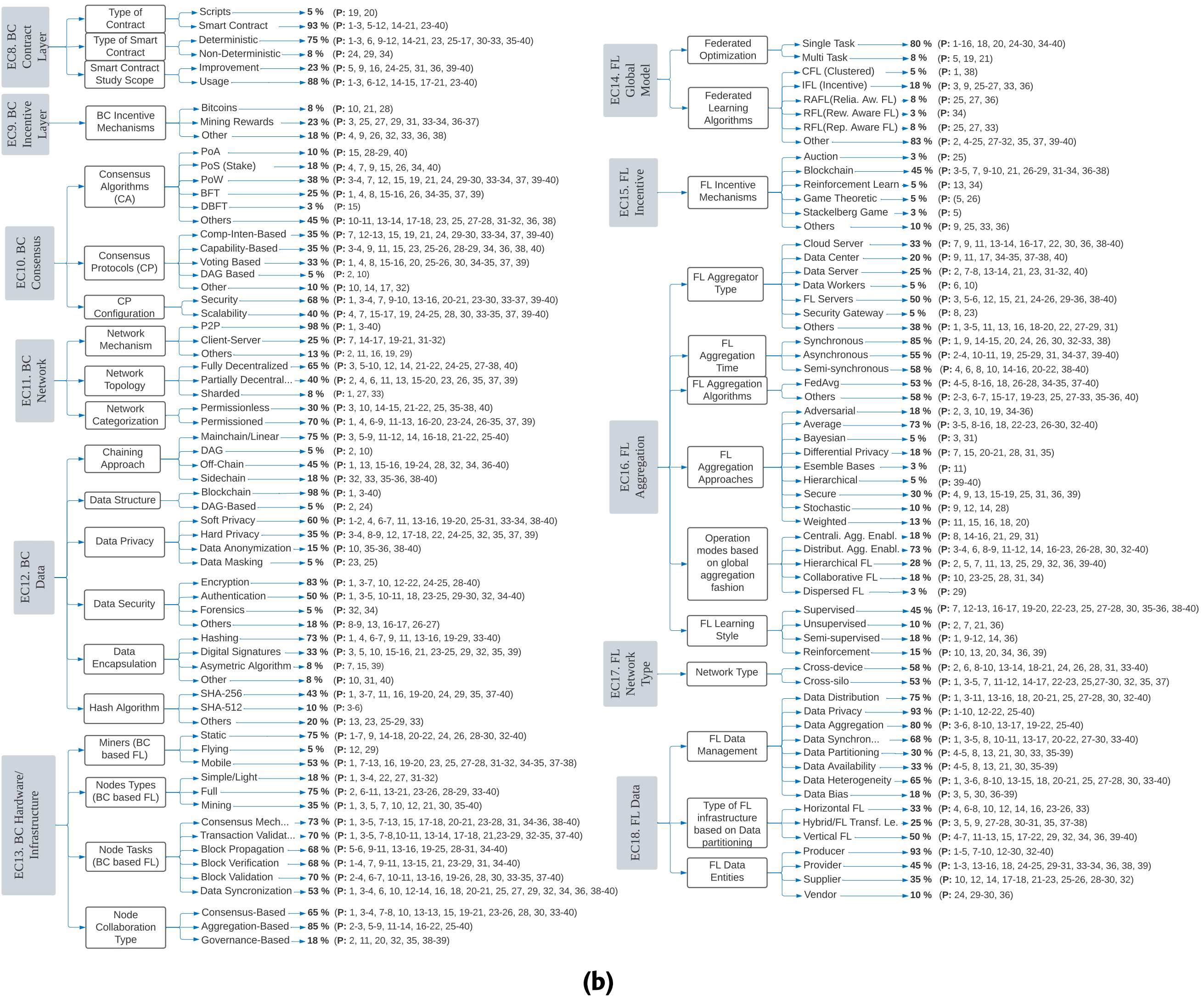

EC8. BC Contract Layer
Type of Contract
Scripts 5 % (P: 19, 20)
Smart Contract 93 % (P: 1-3, 5-12, 14-21, 23-40)
Type of Smart Contract
Deterministic 75 % (P: 1-3, 6, 9-12, 14-21, 23, 25-17, 30-33, 35-40)
Non-Deterministic 8 % (P: 24, 29, 34)
Smart Contract Study Scope
Improvement 23 % (P: 5, 9, 16, 24-25, 31, 36, 39-40)
Usage 88 % (P: 1-3, 6-12, 14-15, 17-21, 23-40)
EC9. BC Incentive Layer
BC Incentive Mechanisms
Bitcoins 8 % (P: 10, 21, 28)
Mining Rewards 23 % (P: 3, 25, 27, 29, 31, 33-34, 36-37)
Other 18 % (P: 4, 9, 26, 32, 33, 36, 38)
EC10. BC Consensus
Consensus Algorithms (CA)
PoA 10 % (P: 15, 28-29, 40)
PoS (Stake) 18 % (P: 4, 7, 9, 15, 26, 34, 40)
PoW 38 % (P: 3-4, 7, 12, 15, 19, 21, 24, 29-30, 33-34, 37, 39-40)
BFT 25 % (P: 1, 4, 8, 15-16, 26, 34-35, 37, 39)
DBFT 3 % (P: 15)
Others 45 % (P: 10-11, 13-14, 17-18, 23, 25, 27-28, 31-32, 36, 38)
Consensus Protocols (CP)
Comp-Inten-Based 35 % (P: 7, 12-13, 15, 19, 21, 24, 29-30, 33-34, 37, 39-40)
Capability-Based 35 % (P: 3-4, 9, 11, 15, 23, 25-26, 28-29, 34, 36, 38, 40)
Voting Based 33 % (P: 1, 4, 8, 15-16, 20, 25-26, 30, 34-35, 37, 39)
DAG Based 5 % (P: 2, 10)
Other 10 % (P: 10, 14, 17, 32)
CP Configuration
Security 68 % (P: 1, 3-4, 7, 9-10, 13-16, 20-21, 23-30, 33-37, 39-40)
Scalability 40 % (P: 4, 7, 15-17, 19, 24-25, 28, 30, 33-35, 37, 39-40)
EC11. BC Network
Network Mechanism
P2P 98 % (P: 1, 3-40)
Client-Server 25 % (P: 7, 14-17, 19-21, 31-32)
Others 13 % (P: 2, 11, 16, 19, 29)
Network Topology
Fully Decentralized 65 % (P: 3, 5-10, 12, 14, 21-22, 24-25, 27-38, 40)
Partially Decentral... 40 % (P: 2, 4, 6, 11, 13, 15-20, 23, 26, 35, 37, 39)
Sharded 8 % (P: 1, 27, 33)
Network Categorization
Permissionless 30 % (P: 3, 10, 14-15, 21-22, 25, 35-38, 40)
Permissioned 70 % (P: 1, 4, 6-9, 11-13, 16-20, 23-24, 26-35, 37, 39)
EC12. BC Data
Chaining Approach
Mainchain/Linear 75 % (P: 3, 5-9, 11-12, 14, 16-18, 21-22, 25-40)
DAG 5 % (P: 2, 10)
Off-Chain 45 % (P: 1, 13, 15-16, 19-24, 28, 32, 34, 36-40)
Sidechain 18 % (P: 32, 33, 35-36, 38-40)
Data Structure
Blockchain 98 % (P: 1, 3-40)
DAG-Based 5 % (P: 2, 24)
Data Privacy
Soft Privacy 60 % (P: 1-2, 4, 6-7, 11, 13-16, 19-20, 25-31, 33-34, 38-40)
Hard Privacy 35 % (P: 3-4, 8-9, 12, 17-18, 22, 24-25, 32, 35, 37, 39)
Data Anonymization 15 % (P: 10, 35-36, 38-40)
Data Masking 5 % (P: 23, 25)
Data Security
Encryption 83 % (P: 1, 3-7, 10, 12-22, 24-25, 28-40)
Authentication 50 % (P: 1, 3-5, 10-11, 18, 23-25, 29-30, 32, 34-40)
Forensics 5 % (P: 32, 34)
Others 18 % (P: 8-9, 13, 16-17, 26-27)
Data Encapsulation
Hashing 73 % (P: 1, 4, 6-7, 9, 11, 13-16, 19-29, 33-40)
Digital Signatures 33 % (P: 3, 5, 10, 15-16, 21, 23-25, 29, 32, 35, 39)
Asymetric Algorithm 8 % (P: 7, 15, 39)
Other 8 % (P: 10, 31, 40)
Hash Algorithm
SHA-256 43 % (P: 1, 3-7, 11, 16, 19-20, 24, 29, 35, 37-40)
SHA-512 10 % (P: 3-6)
Others 20 % (P: 13, 23, 25-29, 33)
EC13. BC Hardware/ Infrastructure
Miners (BC based FL)
Static 75 % (P: 1-7, 9, 14-18, 20-22, 24, 26, 28-30, 32-40)
Flying 5 % (P: 12, 29)
Mobile 53 % (P: 1, 7-13, 16, 19-20, 23, 25, 27-28, 31-32, 34-35, 37-38)
Nodes Types (BC based FL)
Simple/Light 18 % (P: 1, 3-4, 22, 27, 31-32)
Full 75 % (P: 2, 6-11, 13-21, 23-26, 28-29, 33-40)
Mining 35 % (P: 1, 3, 5, 7, 10, 12, 21, 30, 35-40)
Node Tasks (BC based FL)
Consensus Mech... 73 % (P: 1, 3-5, 7-13, 15, 17-18, 20-21, 23-28, 31, 34-36, 38-40)
Transaction Validat... 70 % (P: 1, 3-5, 7-8,10-11, 13-14, 17-18, 21,23-29, 32-35, 37-40)
Block Propagation 68 % (P: 5-6, 9-11, 13-16, 19-25, 28-31, 34-40)
Block Verification 68 % (P: 1-4, 7, 9-11, 13-15, 21, 23-29, 31, 34-40)
Block Validation 70 % (P: 2-4, 6-7, 10-11, 13-16, 19-26, 28, 30, 33-35, 37-40)
Data Syncronization 53 % (P: 1, 3-4, 6, 10, 12-14, 16, 18, 20-21, 25, 27, 29, 32, 34, 36, 38-40)
Node Collaboration Type
Consensus-Based 65 % (P: 1, 3-4, 7-8, 10, 13-13, 15, 19-21, 23-26, 28, 30, 33-40)
Aggregation-Based 85 % (P: 2-3, 5-9, 11-14, 16-22, 25-40)
Governance-Based 18 % (P: 2, 11, 20, 32, 35, 38-39)
EC14. FL Global Model
Federated Optimization
Single Task 80 % (P: 1-16, 18, 20, 24-30, 34-40)
Multi Task 8 % (P: 5, 19, 21)
Federated Learning Algorithms
CFL (Clustered) 5 % (P: 1, 38)
IFL (Incentive) 18 % (P: 3, 9, 25-27, 33, 36)
RAFL(Relia. Aw. FL) 8 % (P: 25, 27, 36)
RFL(Rew. Aware FL) 3 % (P: 34)
RFL(Rep. Aware FL) 8 % (P: 25, 27, 33)
Other 83 % (P: 2, 4-25, 27-32, 35, 37, 39-40)
EC15. FL Incentive
FL Incentive Mechanisms
Auction 3 % (P: 25)
Blockchain 45 % (P: 3-5, 7, 9-10, 21, 26-29, 31-34, 36-38)
Reinforcement Learn 5 % (P: 13, 34)
Game Theoretic 5 % (P: (5, 26)
Stackelberg Game 3 % (P: 5)
Others 10 % (P: 9, 25, 33, 36)
EC16. FL Aggregation
FL Aggregator Type
Cloud Server 33 % (P: 7, 9, 11, 13-14, 16-17, 22, 30, 36, 38-40)
Data Center 20 % (P: 9, 11, 17, 34-35, 37-38, 40)
Data Server 25 % (P: 2, 7-8, 13-14, 21, 23, 31-32, 40)
Data Workers 5 % (P: 6, 10)
FL Servers 50 % (P: 3, 5-6, 12, 15, 21, 24-26, 29-36, 38-40)
Security Gateway 5 % (P: 8, 23)
Others 38 % (P: 1, 3-5, 11, 13, 16, 18-20, 22, 27-29, 31)
FL Aggregation Time
Synchronous 85 % (P: 1, 9, 14-15, 20, 24, 26, 30, 32-33, 38)
Asynchronous 55 % (P: 2-4, 10-11, 19, 25-29, 31, 34-37, 39-40)
Semi-synchronous 58 % (P: 4, 6, 8, 10, 14-16, 20-22, 38-40)
FL Aggregation Algorithms
FedAvg 53 % (P: 4-5, 8-16, 18, 26-28, 34-35, 37-40)
Others 58 % (P: 2-3, 6-7, 15-17, 19-23, 25, 27-33, 35-36, 40)
FL Aggregation Approaches
Adversarial 18 % (P: 2, 3, 10, 19, 34-36)
Average 73 % (P: 3-5, 8-16, 18, 22-23, 26-30, 32-40)
Bayesian 5 % (P: 3, 31)
Differential Privacy 18 % (P: 7, 15, 20-21, 28, 31, 35)
Esemble Bases 3 % (P: 11)
Hierarchical 5 % (P: 39-40)
Secure 30 % (P: 4, 9, 13, 15-19, 25, 31, 36, 39)
Stochastic 10 % (P: 9, 12, 14, 28)
Weighted 13 % (P: 11, 15, 16, 18, 20)
Operation modes based on global aggregation fashion
Centrali. Agg. Enabl. 18 % (P: 8, 14-16, 21, 29, 31)
Distribut. Agg. Enabl. 73 % (P: 3-4, 6, 8-9, 11-12, 14, 16-23, 26-28, 30, 32-40)
Hierarchical FL 28 % (P: 2, 5, 7, 11, 13, 25, 29, 32, 36, 39-40)
Collaborative FL 18 % (P: 10, 23-25, 28, 31, 34)
Dispersed FL 3 % (P: 29)
FL Learning Style
Supervised 45 % (P: 7, 12-13, 16-17, 19-20, 22-23, 25, 27-28, 30, 35-36, 38-40)
Unsupervised 10 % (P: 2, 7, 21, 36)
Semi-supervised 18 % (P: 1, 9-12, 14, 36)
Reinforcement 15 % (P: 10, 13, 20, 34, 36, 39)
EC17. FL Network Type
Network Type
Cross-device 58 % (P: 2, 6, 8-10, 13-14, 18-21, 24, 26, 28, 31, 33-40)
Cross-silo 53 % (P: 1, 3-5, 7, 11-12, 14-17, 22-23, 25,27-30, 32, 35, 37)
EC18. FL Data
FL Data Management
Data Distribution 75 % (P: 1, 3-11, 13-16, 18, 20-21, 25, 27-28, 30, 32-40)
Data Privacy 93 % (P: 1-10, 12-22, 25-40)
Data Aggregation 80 % (P: 3-6, 8-10, 13-17, 19-22, 25-40)
Data Synchron... 68 % (P: 1, 3-5, 8, 10-11, 13-17, 20-22, 27-30, 33-40)
Data Partitioning 30 % (P: 4-5, 8, 13, 21, 30, 33, 35-39)
Data Availability 33 % (P: 4-5, 8, 13, 21, 30, 35-39)
Data Heterogeneity 65 % (P: 1, 3-6, 8-10, 13-15, 18, 20-21, 25, 27-28, 30, 33-40)
Data Bias 18 % (P: 3, 5, 30, 36-39)
Type of FL infrastructure based on Data partitioning
Horizontal FL 33 % (P: 4, 6-8, 10, 12, 14, 16, 23-26, 33)
Hybrid/FL Transf. Le. 25 % (P: 3, 5, 9, 27-28, 30-31, 35, 37-38)
Vertical FL 50 % (P: 4-7, 11-13, 15, 17-22, 29, 32, 34, 36, 39-40)
FL Data Entities
Producer 93 % (P: 1-5, 7-10, 12-30, 32-40)
Provider 45 % (P: 1-3, 13-16, 18, 24-25, 29-31, 33-34, 36, 38, 39)
Supplier 35 % (P: 10, 12, 14, 17-18, 21-23, 25-26, 28-30, 32)
Vendor 10 % (P: 24, 29-30, 36)
(b)

FIGURE 14. Results per extraction criterion, SRQ1 (EC8-EC18)

and improved data synchronization (68%). FL, identified in all studies, plays a pivotal role in model aggregation (100%) within a decentralized architecture (70%) distributed across multiple layers such as Device (88%), Data (93%), and Network (90%). Notably, the architecture design predominantly leans towards layered (68%) and microservices (53%) patterns. The technical dimensions echo a trend favoring hybrid integration schemes (58%) and a consortium-based BC-FL network infrastructure (45%), showcasing a balance between different approaches for system design. EC3 shows insights into frameworks supporting FC-BC-FL convergence, and it reveals a general presence of Formal Models in major IoT players like AWS, Azure, TFF, and others (e.g., IHS [106]). In BC Ethereum and Hyperledger reign supreme, while others (e.g., Algorand [78], [83], [94], AFFIRM [84], Solidity [98]) also hold a presence. In FL, Google TFF is the most popular. Integrated BC-FL frameworks present overall formal models, besides the use of others (i.e., Fedtrust [83], AFFIRM [84]), frameworks like Blade-FL and VBFL are underrepresented, suggesting the need for further exploration. Applications span diverse sectors, prominently healthcare (35%), industrial management (30%), smart cities (30%), and others (i.e., smart grid [83], disaster response [86], BC reliability prediction [88], Internet of Battle Things (IoBT) [96], Internet of Drone Things (IoDT) [103], Agriculture [98], MEC [101], Society [105], Gamming [109], Industry 5.0 [114]) (43%) indicating the versatility and potential impact of this integration. Within Cloud Features (EC5), a hybrid cloud deployment model (48%) and container-based cloud server design (43%) stand out, emphasizing flexibility and scalability. Fog (Edge) Node Features (EC6) spotlight hardware-centric designs (70%) and diverse node types, with gateways (40%) and Fog Computing Nodes (FCN) (83%) dominating. The Client (End-Devices) Features (EC7) underscore the prevalent usage of IoT devices/gateways (68%) and the significance of wireless connections (100%) in this ecosystem. Blockchain-related dimensions elucidate a strong inclination towards smart contracts (93%), permissioned networks (70%), and consensus algorithms such as PoW (38%). Federated Learning's global model (EC14) primarily revolves around single-task federated optimization schemes (80%) and an array of federated learning algorithms (83%). FL Aggregation (EC16) strategies lean towards average-based approaches (73%) in a distributed aggregation-enabled FL (73%) environment, showcasing collaborative learning paradigms. Furthermore, FL Data Management (EC18) accentuates the importance of data privacy (93%) and distribution (75%) within federated learning setups. The synthesis of these findings delineates a landscape where the integration of FL, BC, and FC for IoT applications thrives on decentralized, collaborative, and privacy-centric approaches, spanning multifaceted domains and demanding versatile technical infrastructures to materialize their potential. This detailed exploration reveals the interconnected nature of FL, BC, and FC in driving IoT applications toward enhanced security, privacy, and decentralized operation.

The FC-BC-FL convergence serves as a catalyst for innovative solutions across various domains, prominently healthcare, industrial management, and smart cities. The robust architecture primarily follows a layered and microservices pattern, accommodating the complexities of this integration. Notably, the emphasis on hybrid integration schemes and consortium-based network infrastructures reflects the pursuit of balanced system design strategies. Cloud deployment and Fog (Edge) Node designs exhibit flexibility and scalability, essential for accommodating diverse IoT environments. The prevalence of wireless connections and the utilization of IoT devices/gateways underscore the omnipresence of edge devices in this ecosystem. Within the BC realm, smart contracts, permissioned networks, and consensus algorithms like PoW emerge as pivotal components. FL, on the other hand, emphasizes collaborative learning paradigms through diverse optimization schemes and algorithms. Moreover, FL's focus on data privacy and distribution within its management echoes the criticality of secure and synchronized data handling. Overall, this synthesis delineates a landscape wherein the FC-BC-FL integration for IoT thrives on decentralized, collaborative, and privacy-centric approaches, demanding multifaceted technical infrastructures to harness their full potential.

2) SRQ2 results

Figure 15 showcases the results concerning SRQ2 (EC19-EC33). Notably, within the realm of security, the findings underscored the significance of encryption techniques (93%) in BC, along with encryption-based privacy-preserving FL (70%) and privacy-preserving schemes (FL) (70%). Additionally, the analysis emphasized the prevalence of security concerns related to availability (53%), confidentiality (88%), and integrity (90%). Efficiency, another pivotal facet, demonstrated computational efficiency (95%) as a predominant factor, albeit with lesser emphasis on cost efficiency (10%). Performance indicators showed promising trends in terms of accuracy (90%) but also flagged concerns regarding energy (40%) and latency (58%). Moreover, interoperability and data management surfaced as critical domains, with storage (93%), data sharing (93%), and validation (98%) ranking significantly. Noteworthy within resource allocation/service provisioning metrics were aspects like training time (78%) and data flow (80%), signifying their importance in optimizing FC. The SLR further highlighted the prominence of trust support (85%), authorization (98%), and key management (85%) within the trust and access control domains, crucial for securing IoT applications. Overall, the findings converge on the potential of BC and FL amalgamation in FC to substantially fortify IoT applications, particularly by addressing security concerns, enhancing privacy, optimizing efficiency, ensuring interoperability, managing data effectively, and fortifying trust and access control mechanisms. Besides, from the results can be mentioned that the strengths in privacy preservation, efficiency optimization, and security fortification. Notably, the research emphasizes robust privacy measures, efficient resource utilization, and a strong security focus, showcasing the potential of this integration to fortify IoT ecosystems against data breaches and vulnerabilities. Insights into attack types and defense mechanisms further contribute to foundational strategies for proactive security measures. However, the research landscape reveals notable weaknesses, including limited real-world implementations, challenges in standardization and interoperability, incomplete exploration of evolving attack vectors, and insufficient scrutiny of scalability and cost implications. Bridging these gaps requires concerted efforts toward practical deployments, standardized protocols, comprehensive security considerations, and a deeper understanding of scalability and cost-effectiveness to fully harness the transformative potential of FC-BC-FL integration within IoT domains.

3) SRQ3 results

Figure 16 illustrates the outcomes related to SRQ3 (EC34-EC38). The analysis reveals a predominant emphasis on the stages of Analysis (93%), Design (95%), and Implementation (98%), indicating a mature developmental process compared to Testing (85%). Primary validation methods include experiments (63%) and proof of concepts (28%), indicating a preference for empirical verification. Additionally, the evaluation of Experiment Features shows a tendency towards simulated scenarios (43%) over real-world instances. Python (PyTorch) (38%) and various unspecified tools (80%) are the dominant choices for Blockchain systems and simulation tools, respectively. Diverse datasets such as FEMNIST (8%) and MNIST (35%) have been predominantly used, focusing primarily on images (43%). Notably, most experiments involve a substantial number of clients (20-*), indicating scalability considerations. Overall, the review portrays a landscape where robust design and implementation are evident, yet further exploration and validation in diverse real-world scenarios are necessary for comprehensive advancements. The integration of these technologies displays a maturing progression through developmental phases but with noticeable disparities

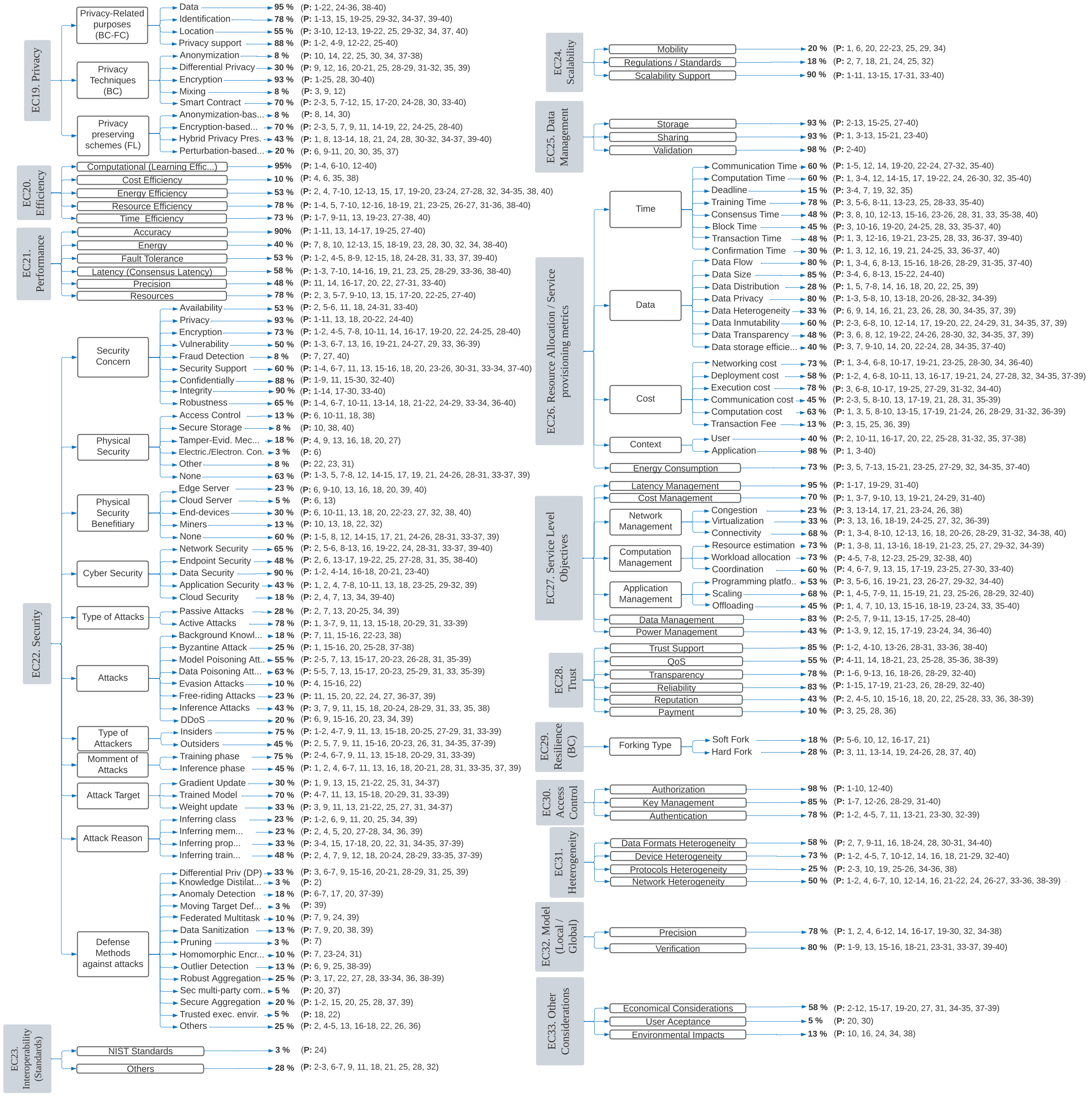

EC19. Privacy
Privacy-Related purposes (BC-FC)
Data 95 % (P: 1-22, 24-36, 38-40)
Identification 78 % (P: 1-13, 15, 19-25, 29-32, 34-37, 39-40)
Location 55 % (P: 3-10, 12-13, 19-22, 25, 29-32, 34, 37, 40)
Privacy support 88 % (P: 1-2, 4-9, 12-22, 25-40)
Privacy Techniques (BC)
Anonymization 8 % (P: 10, 14, 22, 25, 30, 34, 37-38)
Differential Privacy 30 % (P: 9, 12, 16, 20-21, 25, 28-29, 31-32, 35, 39)
Encryption 93 % (P: 1-25, 28, 30-40)
Mixing 8 % (P: 3, 9, 12)
Smart Contract 70 % (P: 2-3, 5, 7-12, 15, 17-20, 24-28, 30, 33-40)
Privacy preserving schemes (FL)
Anonymization-bas... 8 % (P: 8, 14, 30)
Encryption-based... 70 % (P: 2-3, 5, 7, 9, 11, 14-19, 22, 24-25, 28-40)
Hybrid Privacy Pres. 43 % (P: 1, 8, 13-14, 18, 21, 24, 28, 30-32, 34-37, 39-40)
Perturbation-based... 20 % (P: 6, 9-11, 20, 30, 35, 37)
EC20. Efficiency
Computational (Learning Effic...) 95% (P: 1-4, 6-10, 12-40)
Cost Efficiency 10 % (P: 4, 6, 35, 38)
Energy Efficiency 53 % (P: 2, 4, 7-10, 12-13, 15, 17, 19-20, 23-24, 27-28, 32, 34-35, 38, 40)
Resource Efficiency 78 % (P: 1-4, 5, 7-10, 12-16, 18-19, 21, 23-25, 26-27, 31-36, 38-40)
Time Efficiency 73 % (P: 1-7, 9-11, 13, 19-23, 27-38, 40)
EC21. Performance
Accuracy 90% (P: 1-11, 13, 14-17, 19-25, 27-40)
Energy 40 % (P: 7, 8, 10, 12-13, 15, 18-19, 23, 28, 30, 32, 34, 38-40)
Fault Tolerance 53 % (P: 1-2, 4-5, 8-9, 12-15, 18, 24-28, 31, 33, 37, 39-40)
Latency (Consensus Latency) 58 % (P: 1-3, 7-10, 14-16, 19, 21, 23, 25, 28-29, 33-36, 38-40)
Precision 48 % (P: 11, 14, 16-17, 20, 22, 27-31, 33-40)
Resources 78 % (P: 2, 3, 5-7, 9-10, 13, 15, 17-20, 22-25, 27-40)
EC22. Security
Security Concern
Availability 53 % (P: 2, 5-6, 11, 18, 24-31, 33-40)
Privacy 93 % (P: 1-11, 13, 18, 20-22, 24-40)
Encryption 73 % (P: 1-2, 4-5, 7-8, 10-11, 14, 16-17, 19-20, 22, 24-25, 28-40)
Vulnerability 50 % (P: 1-3, 6-7, 13, 16, 19-21, 24-27, 29, 33, 36-39)
Fraud Detection 8 % (P: 7, 27, 40)
Security Support 60 % (P: 1-4, 6-7, 11, 13, 15-16, 18, 20, 23-26, 30-31, 33-34, 37-40)
Confidentially 88 % (P: 1-9, 11, 15-30, 32-40)
Integrity 90 % (P: 1-14, 17-30, 33-40)
Robustness 65 % (P: 1-4, 6-7, 10-11, 13-14, 18, 21-22, 24-29, 33-34, 36-40)
Physical Security
Access Control 13 % (P: 6, 10-11, 18, 38)
Secure Storage 8 % (P: 10, 38, 40)
Tamper-Evid. Mec... 18 % (P: 4, 9, 13, 16, 18, 20, 27)
Electric./Electron. Con. 3 % (P: 6)
Other 8 % (P: 22, 23, 31)
None 63 % (P: 1-3, 5, 7-8, 12, 14-15, 17, 19, 21, 24-26, 28-31, 33-37, 39)
Physical Security Benefitiary
Edge Server 23 % (P: 6, 9-10, 13, 16, 18, 20, 39, 40)
Cloud Server 5 % (P: 6, 13)
End-devices 30 % (P: 6, 10-11, 13, 18, 20, 22-23, 27, 32, 38, 40)
Miners 13 % (P: 10, 13, 18, 22, 32)
None 60 % (P: 1-5, 8, 12, 14-15, 17, 21, 24-26, 28-31, 33-37, 39)
Cyber Security
Network Security 65 % (P: 2, 5-6, 8-13, 16, 19-22, 24, 28-31, 33-37, 39-40)
Endpoint Security 48 % (P: 2, 6, 13-17, 19-22, 25, 27-28, 31, 35, 38-40)
Data Security 90 % (P: 1-2, 4-14, 16-18, 20-21, 23-40)
Application Security 43 % (P: 1, 2, 4, 7-8, 10-11, 13, 18, 23-25, 29-32, 39)
Cloud Security 18 % (P: 2, 4, 7, 13, 34, 39-40)
Type of Attacks
Passive Attacks 28 % (P: 2, 7, 13, 20-25, 34, 39)
Active Attacks 78 % (P: 1, 3-7, 9, 11, 13, 15-18, 20-29, 31, 33-39)
Attacks
Background Knowl... 18 % (P: 7, 11, 15-16, 22-23, 38)
Byzantine Attack 25 % (P: 1, 15-16, 20, 25-28, 37-38)
Model Poisoning Att.. 55 % (P: 2-5, 7, 13, 15-17, 20-23, 26-28, 31, 35-39)
Data Poisoning Att... 63 % (P: 5-5, 7, 13, 15-17, 20-23, 25-29, 31, 33, 35-39)
Evasion Attacks 10 % (P: 4, 15-16, 22)
Free-riding Attacks 23 % (P: 11, 15, 20, 22, 24, 27, 36-37, 39)
Inference Attacks 43 % (P: 3, 7, 9, 11, 15, 18, 20-24, 28-29, 31, 33, 35, 38)
DDoS 20 % (P: 6, 9, 15-16, 20, 23, 34, 39)
Type of Attackers
Insiders 75 % (P: 1-2, 4-7, 9, 11, 13, 15-18, 20-25, 27-29, 31, 33-39)
Outsiders 45 % (P: 2, 5, 7, 9, 11, 15-16, 20-23, 26, 31, 34-35, 37-39)
Momment of Attacks
Training phase 75 % (P: 2-4, 6-7, 9, 11, 13, 15-18, 20-29, 31, 33-39)
Inference phase 45 % (P: 1, 2, 4, 6-7, 11, 13, 16, 18, 20-21, 28, 31, 33-35, 37, 39)
Attack Target
Gradient Update 30 % (P: 1, 9, 13, 15, 21-22, 25, 31, 34-37)
Trained Model 70 % (P: 4-7, 11, 13, 15-18, 20-29, 31, 33-39)
Weight update 33 % (P: 3, 9, 11, 13, 21-22, 25, 27, 31, 34-37)
Attack Reason
Inferring class 23 % (P: 1-2, 6, 9, 11, 20, 25, 34, 39)
Inferring mem... 23 % (P: 2, 4, 5, 20, 27-28, 34, 36, 39)
Inferring prop... 33 % (P: 3-4, 15, 17-18, 20, 22, 31, 34-35, 37-39)
Inferring train... 48 % (P: 2, 4, 7, 9, 12, 18, 20-24, 28-29, 33-35, 37-39)
Defense Methods against attacks
Differential Priv (DP) 33 % (P: 3, 6-7, 9, 15-16, 20-21, 28-29, 31, 25, 39)
Knowledge Distilat... 3 % (P: 2)
Anomaly Detection 18 % (P: 6-7, 17, 20, 37-39)
Moving Target Def... 3 % (P: 39)
Federated Multitask 10 % (P: 7, 9, 24, 39)
Data Sanitization 13 % (P: 7, 9, 20, 38, 39)
Pruning 3 % (P: 7)
Homomorphic Encr... 10 % (P: 7, 23-24, 31)
Outlier Detection 13 % (P: 6, 9, 25, 38-39)
Robust Aggregation 25 % (P: 3, 17, 22, 27, 28, 33-34, 36, 38-39)
Sec multi-party com.. 5 % (P: 20, 37)
Secure Aggregation 20 % (P: 1-2, 15, 20, 25, 28, 37, 39)
Trusted exec. envir. 5 % (P: 18, 22)
Others 25 % (P: 2, 4-5, 13, 16-18, 22, 26, 36)
EC23. Interoperability (Standards)
NIST Standards 3 % (P: 24)
Others 28 % (P: 2-3, 6-7, 9, 11, 18, 21, 25, 28, 32)
EC24. Scalability
Mobility 20 % (P: 1, 6, 20, 22-23, 25, 29, 34)
Regulations / Standards 18 % (P: 2, 7, 18, 21, 24, 25, 32)
Scalability Support 90 % (P: 1-11, 13-15, 17-31, 33-40)
EC25. Data Management
Storage 93 % (P: 2-13, 15-25, 27-40)
Sharing 93 % (P: 1, 3-13, 15-21, 23-40)
Validation 98 % (P: 2-40)
EC26. Resource Allocation / Service provisioning metrics
Time
Communication Time 60 % (P: 1-5, 12, 14, 19-20, 22-24, 27-32, 35-40)
Computation Time 60 % (P: 1, 3-4, 12, 14-15, 17, 19-22, 24, 26-30, 32, 35-40)
Deadline 15 % (P: 3-4, 7, 19, 32, 35)
Training Time 78 % (P: 3, 5-6, 8-11, 13-23, 25, 28-33, 35-40)
Consensus Time 48 % (P: 3, 8, 10, 12-13, 15-16, 23-26, 28, 31, 33, 35-38, 40)
Block Time 45 % (P: 3, 10-16, 19-20, 24-25, 28, 33, 35-37, 40)
Transaction Time 48 % (P: 1, 3, 12-16, 19-21, 23-25, 28, 33, 36-37, 39-40)
Confirmation Time 30 % (P: 1, 3, 12, 16, 19, 21, 24-25, 33, 36-37, 40)
Data
Data Flow 80 % (P: 1, 3-4, 6, 8-13, 15-16, 18-26, 28-29, 31-35, 37-40)
Data Size 85 % (P: 3-4, 6, 8-13, 15-22, 24-40)
Data Distribution 28 % (P: 1, 5, 7-8, 14, 16, 18, 20, 22, 25, 39)
Data Privacy 80 % (P: 1-3, 5-8, 10, 13-18, 20-26, 28-32, 34-39)
Data Heterogeneity 33 % (P: 6, 9, 14, 16, 21, 23, 26, 28, 30, 34-35, 37, 39)
Data Inmutability 60 % (P: 2-3, 6-8, 10, 12-14, 17, 19-20, 22, 24-29, 31, 34-35, 37, 39)
Data Transparency 48 % (P: 3, 6, 8, 12, 19-22, 24-26, 28-30, 32, 34-35, 37, 39)
Data storage efficie... 40 % (P: 3, 7, 9-10, 14, 20, 22-24, 28, 34-35, 37-40)
Cost
Networking cost 73 % (P: 1, 3-4, 6-8, 10-17, 19-21, 23-25, 28-30, 34, 36-40)
Deployment cost 58 % (P: 1-2, 4, 6-8, 10-11, 13, 16-17, 19-21, 24, 27-28, 32, 34-35, 37-39)
Execution cost 78 % (P: 3, 6-8, 10-17, 19-25, 27-29, 31-32, 34-40)
Communication cost 45 % (P: 2-3, 5, 8-10, 13, 17-19, 21, 28, 31, 35-39)
Computation cost 63 % (P: 1, 3, 5, 8-10, 13-15, 17-19, 21-24, 26, 28-29, 31-32, 36-39)
Transaction Fee 13 % (P: 3, 15, 25, 36, 39)
Context
User 40 % (P: 2, 10-11, 16-17, 20, 22, 25-28, 31-32, 35, 37-38)
Application 98 % (P: 1, 3-40)
Energy Consumption 73 % (P: 3, 5, 7-13, 15-21, 23-25, 27-29, 32, 34-35, 37-40)
EC27. Service Level Objectives
Latency Management 95 % (P: 1-17, 19-29, 31-40)
Cost Management 70 % (P: 1, 3-7, 9-10, 13, 19-21, 24-29, 31-40)
Network Management
Congestion 23 % (P: 3, 13-14, 17, 21, 23-24, 26, 38)
Virtualization 33 % (P: 3, 13, 16, 18-19, 24-25, 27, 32, 36-39)
Connectivity 68 % (P: 1, 3-4, 8-10, 12-13, 16, 18, 20-26, 28-29, 31-32, 34-38, 40)
Computation Management
Resource estimation 73 % (P: 1, 3-8, 11, 13-16, 18-19, 21-23, 25, 27, 29-32, 34-39)
Workload allocation 73 % (P: 4-5, 7-8, 12-23, 25-29, 32-38, 40)
Coordination 60 % (P: 4, 6-7, 9, 13, 15, 17-19, 23-25, 27-30, 33-40)
Application Management
Programming platfo.. 53 % (P: 3, 5-6, 16, 19-21, 23, 26-27, 29-32, 34-40)
Scaling 68 % (P: 1, 4-5, 7-9, 11, 15-19, 21, 23, 25-26, 28-29, 32-40)
Offloading 45 % (P: 1, 4, 7, 10, 13, 15-16, 18-19, 23-24, 33, 35-40)
Data Management 83 % (P: 2-5, 7, 9-11, 13-15, 17-25, 28-40)
Power Management 43 % (P: 1-3, 9, 12, 15, 17-19, 23-24, 34, 36-40)
EC28. Trust
Trust Support 85 % (P: 1-2, 4-10, 13-26, 28-31, 33-36, 38-40)
QoS 55 % (P: 4-11, 14, 18-21, 23, 25-28, 35-36, 38-39)
Transparency 78 % (P: 1-6, 9-13, 16, 18-26, 28-29, 32-40)
Reliability 83 % (P: 1-15, 17-19, 21-23, 26, 28-29, 32-40)
Reputation 43 % (P: 2, 4-5, 10, 15-16, 18, 20, 22, 25-28, 33, 36, 38-39)
Payment 10 % (P: 3, 25, 28, 36)
EC29. Resilience (BC)
Forking Type
Soft Fork 18 % (P: 5-6, 10, 12, 16-17, 21)
Hard Fork 28 % (P: 3, 11, 13-14, 19, 24-26, 28, 37, 40)
EC30. Access Control
Authorization 98 % (P: 1-10, 12-40)
Key Management 85 % (P: 1-7, 12-26, 28-29, 31-40)
Authentication 78 % (P: 1-2, 4-5, 7, 11, 13-21, 23-30, 32-39)
EC31. Heterogeneity
Data Formats Heterogeneity 58 % (P: 2, 7, 9-11, 16, 18-24, 28, 30-31, 34-40)
Device Heterogeneity 73 % (P: 1-2, 4-5, 7, 10-12, 14, 16, 18, 21-29, 32-40)
Protocols Heterogeneity 25 % (P: 2-3, 10, 19, 25-26, 34-36, 38)
Network Heterogeneity 50 % (P: 1-2, 4, 6-7, 10, 12-14, 16, 21-22, 24, 26-27, 33-36, 38-39)
EC32. Model (Local / Global)
Precision 78 % (P: 1, 2, 4, 6-12, 14, 16-17, 19-30, 32, 34-38)
Verification 80 % (P: 1-9, 13, 15-16, 18-21, 23-31, 33-37, 39-40)
EC33. Other Considerations
Economical Considerations 58 % (P: 2-12, 15-17, 19-20, 27, 31, 34-35, 37-39)
User Aceptance 5 % (P: 20, 30)
Environmental Impacts 13 % (P: 10, 16, 24, 34, 38)

FIGURE 15. Results per extraction criterion, SRQ2 (EC19 - EC33)

in testing and validation methodologies. While significant progress is evident in the design and implementation stages, the relatively lower emphasis on testing (85%) suggests a potential gap that requires further scrutiny to ensure the robustness and reliability of integrated systems. The prevalent use of experimental validation methods highlights an empirical inclination within the research community, to seek tangible evidence to validate theoretical frameworks. The prevalence of simulated scenarios (43%) suggests a practical approach towards initial validation, but also indicates the necessity for deeper exploration in real-world settings to ensure practical applicability. The dominance of Python (i.e. PyTorch) in Blockchain systems and the use of unspecified tools reflect both flexibility and ambiguity within the domain. These findings underscore the need for a balanced approach, combining theoretical robustness with a nuanced understanding of practical applicability to strengthen the evolution of FC, BC, and FL integration for IoT applications.

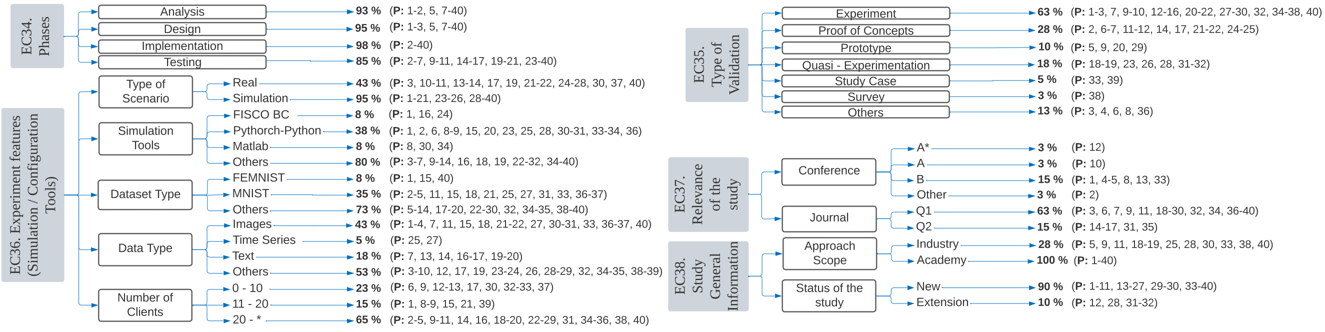

EC34. Phases
Analysis
Design
Implementation
Testing
93 % (P: 1-2, 5, 7-40)
95 % (P: 1-3, 5, 7-40)
98 % (P: 2-40)
85 % (P: 2-7, 9-11, 14-17, 19-21, 23-40)
EC36. Experiment features (Simulation / Configuration Tools)
Type of Scenario
Real
43 % (P: 3, 10-11, 13-14, 17, 19, 21-22, 24-28, 30, 37, 40)
Simulation
95 % (P: 1-21, 23-26, 28-40)
Simulation Tools
FISCO BC
8 % (P: 1, 16, 24)
Pythorch-Python
38 % (P: 1, 2, 6, 8-9, 15, 20, 23, 25, 28, 30-31, 33-34, 36)
Matlab
8 % (P: 8, 30, 34)
Others
80 % (P: 3-7, 9-14, 16, 18, 19, 22-32, 34-40)
Dataset Type
FEMNIST
8 % (P: 1, 15, 40)
MNIST
35 % (P: 2-5, 11, 15, 18, 21, 25, 27, 31, 33, 36-37)
Others
73 % (P: 5-14, 17-20, 22-30, 32, 34-35, 38-40)
Data Type
Images
43 % (P: 1-4, 7, 11, 15, 18, 21-22, 27, 30-31, 33, 36-37, 40)
Time Series
5 % (P: 25, 27)
Text
18 % (P: 7, 13, 14, 16-17, 19-20)
Others
53 % (P: 3-10, 12, 17, 19, 23-24, 26, 28-29, 32, 34-35, 38-39)
Number of Clients
0 - 10
23 % (P: 6, 9, 12-13, 17, 30, 32-33, 37)
11 - 20
15 % (P: 1, 8-9, 15, 21, 39)
20 - *
65 % (P: 2-5, 9-11, 14, 16, 18-20, 22-29, 31, 34-36, 38, 40)
EC35. Type of Validation
Experiment
63 % (P: 1-3, 7, 9-10, 12-16, 20-22, 27-30, 32, 34-38, 40)
Proof of Concepts
28 % (P: 2, 6-7, 11-12, 14, 17, 21-22, 24-25)
Prototype
10 % (P: 5, 9, 20, 29)
Quasi - Experimentation
18 % (P: 18-19, 23, 26, 28, 31-32)
Study Case
5 % (P: 33, 39)
Survey
3 % (P: 38)
Others
13 % (P: 3, 4, 6, 8, 36)
EC37. Relevance of the study
Conference
A*
3 % (P: 12)
A
3 % (P: 10)
B
15 % (P: 1, 4-5, 8, 13, 33)
Other
3 % (P: 2)
Journal
Q1
63 % (P: 3, 6, 7, 9, 11, 18-30, 32, 34, 36-40)
Q2
15 % (P: 14-17, 31, 35)
EC38. Study General Information
Approach Scope
Industry
28 % (P: 5, 9, 11, 18-19, 25, 28, 30, 33, 38, 40)
Academy
100 % (P: 1-40)
Status of the study
New
90 % (P: 1-11, 13-27, 29-30, 33-40)
Extension
10 % (P: 12, 28, 31-32)

FIGURE 16. Results per extraction criterion, SRQ3 (EC34 - EC38)

B. RESULTS PER YEAR AND COUNTRY

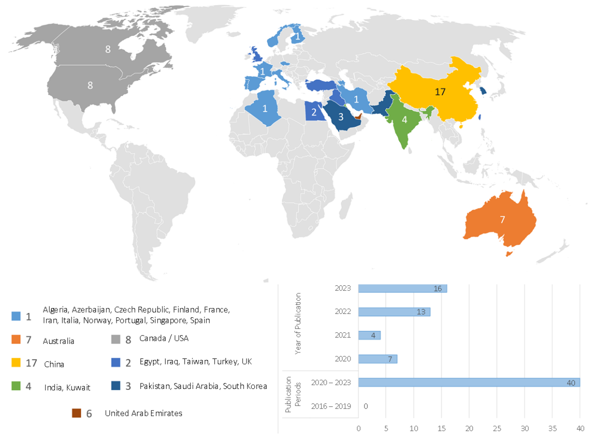


FIGURE 17. Results per country and year

Figure 17 shows the distribution of results across different years and countries among the 40 selected studies. Based on the figure, we can state that all these studies were conducted from 2020 onwards, indicating the contemporary relevance of this evolving research field. As of 2023 (with the analysis considered until August), 40% of the studies were conducted, underscoring the current prevalence of this topic. Among the countries showing a substantial interest in integrating these technologies, China leads with 17 studies, followed by the United States and Canada with 8 studies each, and Australia with 7 studies. Additionally, parts of Europe, Africa, and Arab countries have shown notable interest in this field. While the FL concept emerged in 2016, it took time to integrate these technologies. Nevertheless, recent trends unmistakably demonstrate the growing significance of this combination.

V. DISCUSSION

This section synthesizes the relevant aspects derived from the review, covering various approaches and techniques for integrating FC-BC-FL technologies. This synthesis is based on carefully selected extraction criteria and is illustrated through bubble graphs, analyzing the technologies integration, alongside their associated challenges and opportunities.

A. CHALLENGES AND OPPORTUNITIES. THE SYNTHESIS OF THE MOST RELEVANT FINDINGS

This subsection synthesizes key findings concerning existing frameworks, application fields, interoperability, networking systems, on-chain/off-chain structures, and BC-FL features.

1) Frameworks use per application field and standardization

Figure 18 illustrates the practical application of BC and FL frameworks in specific fields and the standardization when integrating FC-BC-FL, providing insights into the number of studies addressing each framework within particular application fields. Several studies feature innovative formal models crafted to introduce solutions for integrating FC-BC-FL. Notably, Healthcare (six studies in BC, six in FL, 12 in BC-FL), Industry (three studies in BC, six in FL, 10 in BC-FL), Smart Cities (four studies in BC, five in FL, 11 in BC-FL), Transport (four studies in BC, five in FL, 8 in BC-FL), and Others (five studies in BC, 8 in FL, 15 in BC-FL) emerge as a main area for BC and FL framework implementation.

However, a significant research gap exists in utilizing certain existing frameworks and exploiting their capabilities. For example, there is limited exploration of frameworks like Flower, IBM FL, or Azure Flute within FL. Similarly, there is a lack of exploration in utilizing IBM BC, Cisco BC, or wider adoption of Ethereum or Hyperledger for BC applications. Moreover, existing BC-FL frameworks have not been used within fields integrating all three technologies. Moreover, most analyzed studies introduce formal models for new frameworks, posing a challenge in testing their application in different fields to demonstrate their versatility. Additionally, the Energy and Business sectors under-utilize these frameworks, resulting in research gaps. Addressing these areas presents significant opportunities to strengthen existing frameworks and explore untapped areas.

In this early phase of developing specialized frameworks for integrating FC-BC-FL, notable efforts are underway to bridge this gap. These efforts involve creating new formal models that integrate distributed intelligence and security into applications. Additionally, existing platforms are compelled to adapt to these requirements, aiming to provide the combined capabilities of these three technologies to IoT applications. Both emerging and established frameworks are emphasizing this integration, promising more robust IoT solutions.

Regarding standardization, there is a notable absence of using existing sources like IEEE, ISO, and NIST to standardize BC-FL frameworks for instance to wider adoption and convergence of these technologies. Furthermore, the specification of elements within solutions using tools like DSLs is lacking investigation (see Figure 18).

Emphasizing the importance of specification and standardization across FC/EC-BC-FL integration is crucial for seamless interoperability, compatibility mitigation, security enhancement, and innovation acceleration. Standards provide a unified framework for communication and collaboration, facilitating cohesive system development and enabling efficient utilization of combined technology benefits. Adherence to standardized protocols fosters trust and confidence in data exchange integrity, vital for privacy-sensitive and reliable applications. Ultimately, specification and standardization propel widespread adoption and advancement of these technologies across diverse industries.

An important challenge lies in ensuring that new solutions prioritize adherence to standards and specifications, fostering scalability and interoperability. By aligning with established norms from reputable organizations, developers can pave the way for seamless integration and accelerated adoption, ultimately driving advancements in the field.

2) BC and FL features per networking systems

Figure 19 illustrates the distribution of studies examining networking systems that integrate FC-BC-FL, encompassing various BC features like consensus protocols, chaining approaches, and data security, alongside FL elements such as aggregator types, aggregation algorithms, and learning styles.

The adoption of specific BC-FL features is primarily observed in IoT networks, with notable application domains including vehicular and mobile/RAN networks. The rapid emergence of 5G and forthcoming challenges posed by 6G networks create a pressing need to enhance distributed systems within these mobile scenarios. Mobile environments encapsulate essential components of FC/EC, offering an optimal platform for implementing BC technology and FL methodologies. However, despite potential advantages, there remains a significant gap in understanding how to effectively leverage these technologies within dynamic and resource-constrained Mobile/RAN network environments.

An analysis of the consensus protocols utilized across diverse networking systems reveals prevailing trends. Notably, capability-based protocols feature prominently, with 11 studies in IoT, one in CDN, two in LRPON/PLC, six in mobile/RAN, and five in vehicular networks. Additionally, compute-intensive protocols are notable, with 13 studies in IoT and three each in vehicular and mobile/RAN networks, while voting-based protocols exhibit presence with 8 in IoT,

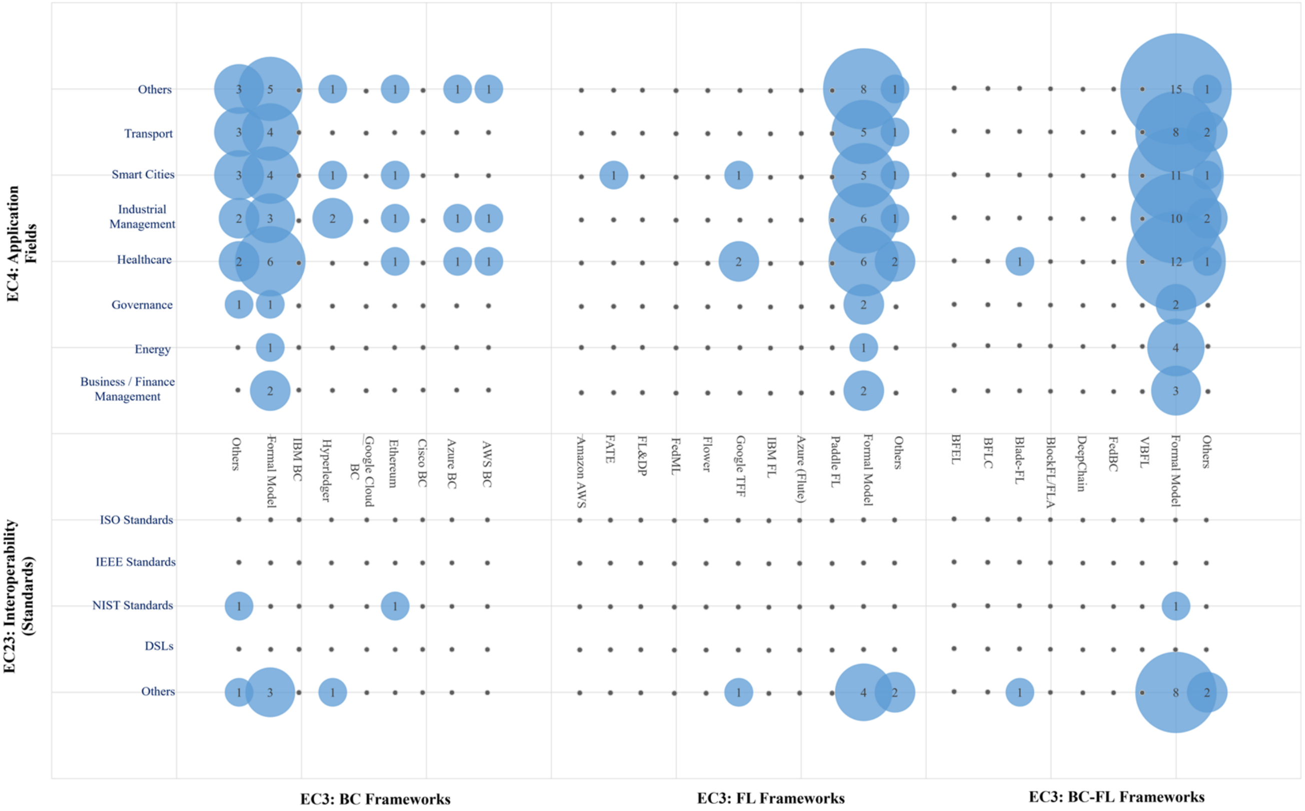


FIGURE 18. BC, FL, and BC-FL frameworks used by application fields in FC-BC-FL integration

one each in CDN and LRPON/PLC, and three each in mobile/RAN and vehicular networks. In contrast, DAG-based protocols appear less utilized, represented by only one study in IoT and one in vehicular networks, despite the increasing FC-BC-FL integration. Furthermore, alternative forms of consensus like Secure Multiparty Computation-based (P10), are presented in IoT and vehicular networks (see Figure 19).

Regarding the chaining approach, the main chain predominates in IoT applications, drawing significant attention with 24 dedicated studies. Specifically, 10 studies focused on mobile/RAN, six on vehicular networks, four on CDN, and one on LRPON/PLC within the main chain paradigm. Additionally, off-chain chaining approaches play a role in these network types, with 16 studies in IoT, 7 in Mobile/RAN, four in vehicular networks, and one in CDN. It's also noteworthy that sidechain approaches, with five studies in IoT and two in Mobile/RAN and vehicular networks, and DAG approaches with no studies, appear to be underutilized within these contexts. Conversely, the types of aggregators used by network systems in integrating FC-BC-FL show that FL servers are most commonly used, with 16 studies in IoT, two in CDN, one in LRPON, 7 in Mobile/RAN, and five in vehicular networks. Moreover, cloud servers (11 studies in IoT, one each in CDN and vehicular, and two in Mobile/RAN networks) and other aggregators (e.g., fog/edge nodes, fog/edge servers, edge devices, CDN servers, Fog Cloud Agent - FCA) are commonly employed, with 13 studies in IoT, 7 in mobile/RAN, three in vehicular and CDN, and two in LRPON/PLC (see Figure 19).

The most utilized aggregation algorithms in networking systems are FedAvg algorithms (14 studies in IoT, two in CDN, one in LRPON/PLC, six in Mobile, and five in vehicular networks). However, there is a lack of utilization of other aggregation algorithms such as FedProx or FedMA. Other types of aggregations are more prevalent, as observed in 20 studies for IoT, 10 in Mobile/RAN, and five in vehicular networks (e.g., Stochastic Gradient Descend (SGD), Distributed Approximate Newton (DAN), DLP-LDP, Differential Privacy SGD, FedSGD). Nonetheless, the learning styles employed across networking systems indicate a prevalence of supervised models, with 13 studies in IoT, six in Mobile/RAN, five in vehicular networks, and one in CDN. Additionally, unsupervised, semi-supervised, and reinforcement learning methods are utilized, but to a lesser extent (see Figure 19).

VI. ADVANTAGES AND LIMITATIONS OF THE FC-BC-FL INTEGRATION

The synergy among FC, BC, and FL offers several advantages but also presents limitations depending on their utilized approaches and techniques. Following the SLR development, this section will address these crucial aspects.

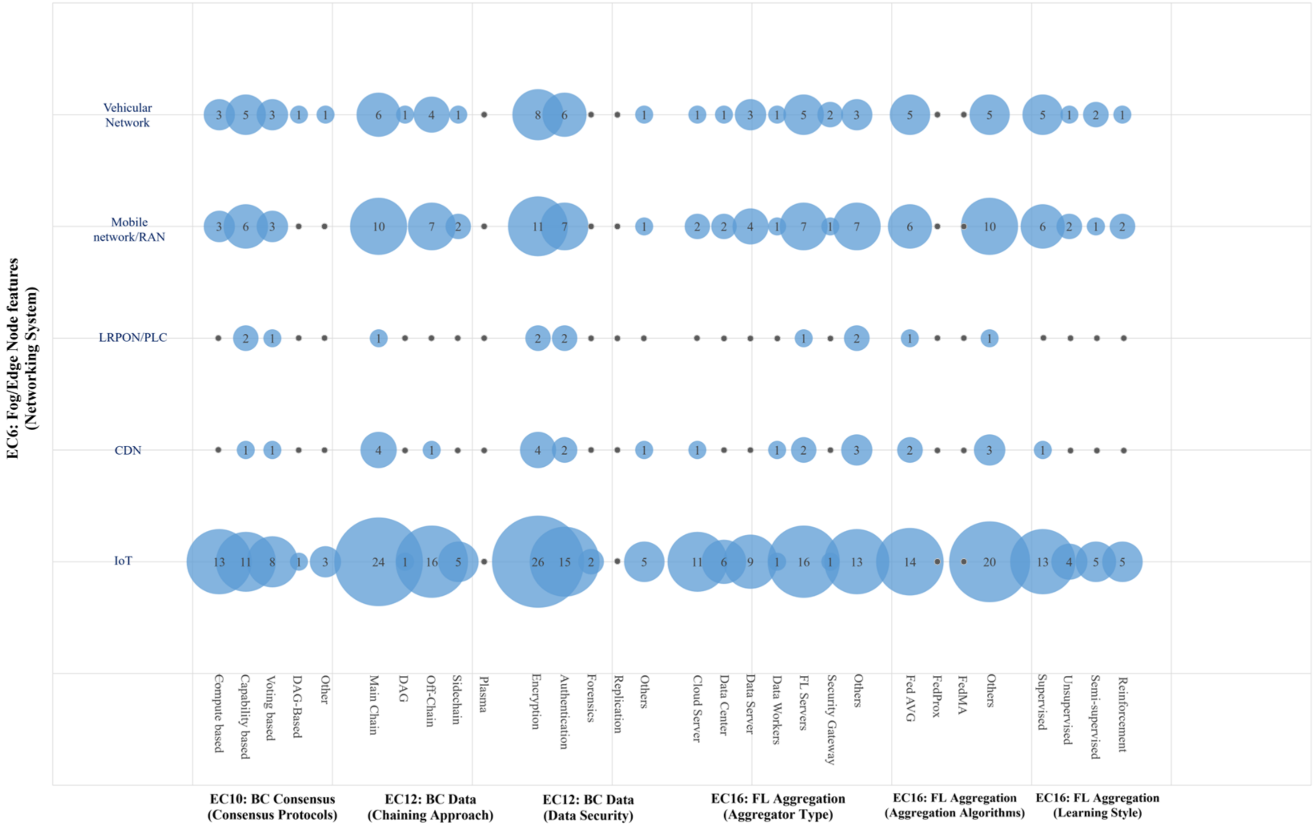


FIGURE 19. Fog/Edge node networking system used by BC: consensus protocol, chaining approach, data security, FL: aggregator type, aggregation algorithms, learning style

A. ADVANTAGES

1) *Data Privacy.* The integration allows sensitive data to be processed locally on edge devices or within the fog network, reducing the need to transmit raw data to centralized servers. Through BC's immutable and transparent ledger, data access and transactions are securely recorded, ensuring that only authorized parties can access specific data. FL further reinforces privacy by enabling model training on decentralized edge devices without transferring raw data, thereby minimizing the risk of data exposure and preserving user privacy.
2) *Data Security.* The integration leverages distributed ledger technology provided by BC, which ensures tamper-resistant and transparent data transactions. The FC/EC processing closer to the source reduces the attack surface and vulnerabilities associated with transmitting data over long distances. FL allows model training directly on fog/edge devices, avoiding the need to transfer sensitive data to centralized servers, thus mitigating the risk of interception or unauthorized access.
3) *Decentralization.* It embodies the core of integrating these technologies. BC decentralizes the consensus process, ensuring no single entity controls the network. This distributed consensus model enhances trust and eliminates intermediaries, lowering the risk of data manipulation or censorship. Fog/Edge further decentralizes data processing by enabling computations at the network's edge, closer to data sources, reducing reliance on centralized cloud infrastructure. FL decentralizes model training by allowing edge devices to collaboratively train models without sending raw data to a central server.
4) *Scalability.* This integration optimizes the utilization of computing resources and scalability by distributing computational tasks across Fog/Edge nodes, enabling data processing and model training closer to the data source, thereby reducing latency and bandwidth requirements. BC ensures scalability by providing a platform for recording and validating transactions, accommodating growing data volumes without performance issues. FL enhances scalability by enabling collaborative model training across data sources, leveraging collective computational power and bandwidth while minimizing reliance on centralized infrastructure.
5) *Synchronization.* This integration facilitates synchronization through efficient coordination of data processing and model updates across distributed Fog/Edge nodes. It ensures timely synchronization of data without relying heavily on centralized servers. BC synchronizes data transactions across all nodes in the network, ensuring consistency and transparency. FL synchronizes model updates across sources, allowing models to learn from diverse data sources while maintaining synchronization with the central model.
6) *Other.* Additional benefits include collaborative model training, enabling distributed devices to enhance a global model while safeguarding data privacy, thereby improving accuracy and compliance. Another advantage is optimized data processing, ensuring efficient resource utilization by processing data closer to its source, leading to enhanced scalability, resilience, and security. Together, these features offer a comprehensive solution for maximizing utility and security in distributed data applications.

B. LIMITATIONS

1) *Complexity.* Integrating FC-BC-FL requires intricate system design and coordination, which may increase implementation complexity and development costs.
2) *Security Concerns.* While BC enhances data security, it's not immune to vulnerabilities such as attacks or smart contract bugs. FL relies on secure communication protocols to protect privacy, but edge devices may still be susceptible to physical attacks or malware.
3) *Data Privacy and Regulation.* FL addresses privacy concerns by keeping data local, but regulatory compliance and data governance become more challenging in distributed environments. Ensuring compliance with privacy regulations such as GDPR requires careful management of data access and consent mechanisms.
4) *Performance Overhead.* The additional computational and communication overhead introduced by BC and FL may impact system performance, particularly in resource-constrained IoT environments. Balancing performance requirements with security and privacy concerns is essential for successful integration.
5) *Ethical Issues.* Integrating FC-BC-FL presents ethical considerations that demand careful examination. Foremost among these are concerns surrounding data privacy, as the vast amounts of sensitive information generated by IoT devices require robust safeguards to prevent unauthorized access or misuse. Security is another critical aspect, as the decentralized nature of these technologies introduces new vulnerabilities that must be addressed to mitigate potential cyber threats. Furthermore, the deployment of such advanced systems may have profound social implications, including issues of digital divide, algorithmic bias, and the exacerbation of existing inequalities. Therefore, ethical frameworks must be established to guide the responsible development and implementation of these technologies, ensuring that they not only deliver technical advancements but also uphold fundamental principles of fairness, transparency, and societal well-being.

As presented, the integration of these paradigms showcases promising opportunities to enhance IoT solutions. Here, addressing their complexities and limitations is crucial to realizing their full potential in real-world applications.

VII. CONCLUSIONS AND FUTURE WORK

Integrating Fog/Edge Computing, Blockchain, and Federated Learning within the Internet of Things ecosystem presents a compelling avenue for enhancing various network systems and applications. This SLR study presented an analysis and categorization of BC and FL technologies within FC environments, addressing a research gap. While adhering to the guidelines outlined by Kitchenham for conducting an SLR, it is crucial to acknowledge inherent limitations in the process. Despite efforts to encompass a broad spectrum of sources, inadvertent exclusion of relevant studies due to stringent inclusion criteria remains a possibility. Furthermore, resource constraints such as time limitations and database accessibility may impact the review's comprehensiveness. In addition to limitations in the review process, constraints regarding the findings obtained must be acknowledged, such as potential biases among selected studies. Moreover, the dynamic nature of the involved technologies may render some findings outdated or incomplete. Thus, while the SLR provides valuable insights, it is imperative to interpret results with awareness of these limitations, ensuring a nuanced understanding of the research landscape.

The success attained in this study, compared to existing ones, can be attributed to several key factors that enhance its effectiveness in addressing research objectives. Firstly, the meticulous analysis involved comparing 16 similar literature review studies against our SLR proposal, which allowed us to identify critical gaps in existing research concerning the analysis of FC/EC-BC-FL integration. Consequently, areas requiring further exploration were pinpointed, and provided a detailed assessment of them. This approach enriches the understanding of the subject matter and brings a holistic perspective by examining the integration of all three paradigms comprehensively, rather than focusing solely on partial combinations as presented in existing studies. Secondly, the meticulous examination of various criteria, including architectural features, security concerns, and application domains, provides a comprehensive overview of integration potential.

The SLR of 40 papers uncovers insights into this emerging field, using 38 criteria covering FC-BC-FL integration's architectural and technical features. Since 2020, there has been a notable rise in interest from enterprises and academia in developed nations contributing to integrated solutions with these technologies. The survey data was analyzed criteria by criteria to showcase their influence and consideration within the domain. Additionally, a combined analysis unveils relationships and research prospects. Interpretations of strengths, weaknesses, and research directions for FC-BC-FL integration from the surveyed literature are also provided.

The FC-BC-FL integration presents multifaceted advantages poised to revolutionize IoT landscapes, offering enhanced data privacy, security, and decentralized consensus mechanisms through BC, while FL facilitates collaborative model training without compromising data privacy. Simultaneously, FC/EC optimizes data processing, reducing latency by distributing computational tasks closer to the data source for amplified efficiency and responsiveness within IoT frameworks, offering future directions for improving integration and leveraging its advantages. These can include:

Frameworks Utilization and Implementation: Future efforts could focus on developing guidelines or tools to facilitate widespread and practical implementation of existing frameworks. The emergence of novel formal models and frameworks within examined studies signifies a proactive response to specialized framework deficiencies, significantly contributing to the maturation of integration methodologies. However, additional efforts are needed.

Specification and Standardization: Establishing technical and industry standards and protocols would ensure interoperability and streamline integration practices, fostering more robust and scalable implementations.

Exploration in Diverse Network Contexts: Examining how this integration can extend its benefits to diverse network environments beyond IoT, including Mobile/RAN Networks (e.g., 5G and 6G), would broaden its applicability and potential impact. For instance, addressing computation offloading or resource allocation needs. Further exploration in these networks would unveil insights and opportunities for leveraging integration to tackle specific challenges or enhance performance across various applications.

Comprehensive Approach: Emphasizing interdisciplinary collaborations and holistic research efforts to address identified gaps, enhance understanding, and advance the integration's applicability.

In conclusion, the synthesis of these technologies within the IoT domain represents an avant-garde approach poised to augment network systems profoundly. Addressing the identified gaps in existing frameworks, standardization efforts, and expanding the scope to encompass diverse network scenarios can further enrich the understanding and applicability of this integration. This comprehensive approach holds immense promise in propelling the evolution of IoT ecosystems towards heightened efficiency, security, and scalability.

REFERENCES

[1] R. Minerva, A. Biru, and D. Rotondi, "Towards a Definition of the Internet of Things (IoT)," IEEE Internet Iniviative, MsC. Thesis, Future Broadband Networks., Politecnico di Torino., Torino., Piedmont, 2015., Available: https://iot.ieee.org/definition.html.

[2] A. M. Rahmadi, S. Bayramov, and B. Kiani Kalejahi, "Internet of Things Applications: Opportunities and Threats," *Springer Link J. Wireless Personal Communications*, vol. 122, no. 1, pp. *451-476*, Jan, 2022, DOI: 10.1007/s11277-021-08907-0 [Online].

[3] A. Khanna, and S. Kaur, "Internet of Things (IoT), Applications and Challenges: A Comprehensive Review," *Springer Link J. Wireless Personal Communications*, vol. 114, no. 1, pp. *1687-1762*, Sep, 2020, DOI: 10.1007/s11277-020-07446-4 [Online].

[4] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," Proceedings of the first edition of the MCC workshop on Mobile cloud computing - MCC '12, 2012, DOI: https://doi.org/10.1145/2342509.2342513

[5] H. N. Atlam, R. J. Walters, and G. B. Wills, "Fog Computing and the Internet of Things: A Review," *MDPI J. Big Data and Cognitive Computing*, vol. 2, no. 2, pp. *10*, Sep, 2020. DOI: http://dx.doi.org/10.3390/bdcc2020010 [Online].

[6] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Decentralized business review Journal, 2008., Available: https://bitcoin.org/bitcoin.pdf.

[7] J. Konečný, B. McMahan , and D. Ramage. ''Federated Optimization: Distributed Optimization Beyond the Datacenter,'' Google. Seattle, WA, USA., Nov, 11 2015. [Online] Available: https://arxiv.org/abs/1511.03575.

[8] M. Iorga, L. Feldman, R. Barton, M. Martin, N. Goren. and C. Mahmoudi, (2015), Fog Computing Conceptual Model, Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD, [online], https://doi.org/10.6028/NIST.SP.500-32

[9] H. -N. Dai, Z. Zheng and Y. Zhang, "Blockchain for Internet of Things: A Survey," in IEEE Internet of Things Journal, vol. 6, no. 5, pp. 8076-8094, Oct. 2019, DOI: 10.1109/JIOT.2019.2920987 [Online].

[10] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li and H. Vincent Poor, "Federated Learning for Internet of Things: A Comprehensive Survey," in "IEEE Communications Surveys & Tutorials," vol. 23, no. 3, pp. 1622-1658, 2021, DOI: 10.1109/COMST.2021.3075439.

[11] Y. I. Alzoubi, A. Gill, and A. Mishra, "A systematic review of the purposes of Blockchain and fog computing integration: classification and open issues," Journal of Cloud Computing, vol. 11, no. 1, Nov. 2022, DOI: https://doi.org/10.1186/s13677-022-00353-y.

[12] F. Firouzi, S. Jiang, K. Chakrabarty, B. Farahani, M. Daneshmand, J. Song, and K. Mankodiya., "Fusion of IoT, AI, Edge–Fog–Cloud, and Blockchain: Challenges, Solutions, and a Case Study in Healthcare and Medicine," in IEEE Internet of Things Journal, vol. 10, no. 5, pp. 3686-3705, 1 March1, 2023, DOI: 10.1109/JIOT.2022.3191881.

[13] R. Yang, F. R. Yu, P. Si, Z. Yang, and Y. Zhang, "Integrated Blockchain and Edge Computing Systems: A Survey, Some Research Issues and Challenges," in IEEE Communications Surveys & Tutorials, vol. 21, no. 2, pp. 1508-1532, 2019, DOI: 10.1109/COMST.2019.2894727.

[14] H. Baniata and A. Kertesz, "A Survey on Blockchain-Fog Integration Approaches," in IEEE Access, vol. 8, pp. 102657-102668, 2020, DOI: 10.1109/ACCESS.2020.2999213.

[15] 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," in IEEE Communications Surveys & Tutorials, vol. 21, no. 2, pp. 1676-1717, May 2019, DOI: 10.1109/COMST.2018.2886932.

[16] D. C. Nguyen, M. Ding, Q.-V Pharm, P. N. Pathirana, L. B. Lee, A. Seneviratne, J. Li, D. Niyato, and H. V. Poor, "Federated Learning Meets Blockchain in Edge Computing: Opportunities and Challenges," IEEE Internet of Things Journal, pp. 1–1, 2021, DOI: https://doi.org/10.1109/jiot.2021.3072611.

[17] L. U. Khan, W. Saad, Z. Han, E. Hossain and C. S. Hong, "Federated Learning for Internet of Things: Recent Advances, Taxonomy, and Open Challenges," in IEEE Communications Surveys & Tutorials, vol. 23, no. 3, pp. 1759-1799, Sept, 2021, DOI: 10.1109/COMST.2021.3090430.

[18] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y-C. Liang, Q. Yang, D. Niyato, and C. Miao., "Federated Learning in Mobile Edge Networks: A Comprehensive Survey," in IEEE Communications Surveys & Tutorials, vol. 22, no. 3, pp. 2031-2063, Oct 2020, DOI: 10.1109/COMST.2020.2986024.

[19] J. Zhu, J. Cao, D. Saxena, S. Jiang, and H. Ferradi, "Blockchain-empowered Federated Learning: Challenges, Solutions, and Future Directions," ACM Computing Surveys, Nov. 2022, DOI: https://doi.org/10.1145/3570953.

[20] M. Ali, H. Karimipour, and M. Tariq, "Integration of Blockchain and Federated Learning for Internet of Things: Recent Advances and Future Challenges," Computers and Security, p. 102355, Jun. 2021, DOI: https://doi.org/10.1016/j.cose.2021.102355.

[21] D. Hou, J. Zhang, K. L. Man, J. Ma and Z. Peng, "A Systematic Literature Review of Blockchain-based Federated Learning: Architectures, Applications and Issues," 2021 2nd Information Communication Technologies Conference (ICTC), Nanjing, China, 2021, pp. 302-307, DOI: 10.1109/ICTC51749.2021.9441499.

[22] D. Li et al., "Blockchain for federated learning toward secure distributed machine learning systems: a systemic survey," Soft Computing, Nov. 2021, DOI: https://doi.org/10.1007/s00500-021-06496-5.

[23] I. Wael, M. Nour, B. Turnbull, N. Sohrabi, and Z. Tari, "Blockchain-based Federated Learning for Securing Internet of Things: A Comprehensive Survey," ACM Computing Surveys, Sep. 2022, DOI: https://doi.org/10.1145/3560816.

[24] D. Li, Z. Luo, and B. Cao, "Blockchain-based federated learning methodologies in smart environments," Cluster Computing, Nov. 2021, DOI: https://doi.org/10.1007/s10586-021-03424-y.

[25] Y. Qu, M. P. Uddin, C. Gan, Y. Xiang, L. Gao, and J. Yearwood, "Blockchain-Enabled Federated Learning: A Survey," ACM Computing Surveys, Mar. 2022, DOI: https://doi.org/10.1145/3524104.

[26] S. H. Alsamhi et al., "Federated Learning Meets Blockchain in Decentralized Data-Sharing: Healthcare Use Case," in IEEE Internet of Things Journal, doi: 10.1109/JIOT.2024.3367249.

[27] B. Kitchenham, ''Procedures for performing systematic reviews,'' Keele University., Keele., UK, Rep. *1-26*, 2004.

[28] B. Kitchenham, ''Guidelines for performing systematic literature reviews in software engineering,'' Keele University., Keele, Staffs., UK, Rep. *1-65*, 2007.

[29] P. Mårtensson, U. Fors, S.-B. Wallin, U. Zander, and G. H. Nilsson, "Evaluating research: A multidisciplinary approach to assessing research practice and quality," Research Policy, vol. 45, no. 3, pp. 593–603, Apr. 2016, DOI: https://doi.org/10.1016/j.respol.2015.11.009.

[30] "IoT, from Cloud to Fog Computing," Cisco Blogs, Mar. 26, 2015. https://blogs.cisco.com/perspectives/iot-from-cloud-to-fog-computing

[31] P. Hu, S. Dhelim, H. Ning, and T. Qiu, "Survey on fog computing: architecture, key technologies, applications and open issues," Journal of Network and Computer Applications, vol. 98, pp. 27–42, Nov. 2017, DOI: https://doi.org/10.1016/j.jnca.2017.09.002.

[32] A. Yousefpour et al., "All one needs to know about fog computing and related edge computing paradigms: A complete survey," Journal of Systems Architecture, Feb. 2019, DOI: https://doi.org/10.1016/j.sysarc.2019.02.009

[33] B. Hamza, "Integrating Blockchain and Fog Computing Technologies for Efficient Privacy-preserving Systems," www.doktori.bibl.u-szeged.hu, Feb. 16, 2023. https://www.doktori.bibl.u-szeged.hu/id/eprint/11555/

[34] D. Yaga, P. Mell, N. Roby, and K. Scarfone, "Blockchain Technology Overview," National Institute of Standards and Technology, Oct. 2018, DOI: https://doi.org/10.6028/nist.ir.8202.

[35] M. Nofer, P. Gomber, O. Hinz, and D. Schiereck, "Blockchain", "Springer Link J. Business and Information Systems Engineering", vol. 59, no. 3, pp. 183-187, Jun, 2017, DOI: https://doi.org/10.1007/s12599-017-0467-3.

[36] M. Moshawrab, M. Adda, A. Bouzouane, H. Ibrahim, and A. Raad, "Reviewing Federated Learning Aggregation Algorithms; Strategies, Contributions, Limitations and Future Perspectives," Electronics, vol. 12, no. 10, p. 2287, Jan. 2023, DOI: https://doi.org/10.3390/electronics12102287

[37] H. Zhang, J. Bosch, and H. H. Olsson, "Federated Learning Systems: Architecture Alternatives," IEEE Xplore, Dec. 01, 2020. https://ieeexplore.ieee.org/abstract/document/9359305/.

[38] M. Richards, "Software Architecture Patterns," 146.201, 2015, DOI: http://hdl.handle.net/1/5665.

[39] X. Huang, C. Zhi, Q. Chen and J. Zhang, "Blockchain-Enabled Clustered Federated Learning in Fog Computing Networks," 2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall), Norman, OK, USA, 2021, pp. 1-5, DOI: 10.1109/VTC2021-Fall52928.2021.9625303.

[40] X. Xu et al., "A Taxonomy of Blockchain-Based Systems for Architecture Design," 2017 IEEE International Conference on Software Architecture (ICSA), Gothenburg, Sweden, 2017, pp. 243-252, DOI: 10.1109/ICSA.2017.33.

[41] M. S. Jere, T. Farnan and F. Koushanfar, "A Taxonomy of Attacks on Federated Learning," in IEEE Security & Privacy, vol. 19, no. 2, pp. 20-28, March-April 2021, DOI: 10.1109/MSEC.2020.3039941.

[42] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," Future Generation Computer Systems, vol. 115, pp. 619-640, 2021. [Online]. Available: https://doi.org/10.1016/j.future.2020.10.007.

[43] J. Kang et al., 'Scalable and Communication-Efficient Decentralized Federated Edge Learning with Multi-blockchain Framework', in Blockchain and Trustworthy Systems, 2020, pp. 152–165.

[44] Y. Li, C. Chen, N. Liu, H. Huang, Z. Zheng and Q. Yan, "A Blockchain-Based Decentralized Federated Learning Framework with Committee Consensus," in IEEE Network, vol. 35, no. 1, pp. 234-241, January/February 2021, DOI: 10.1109/MNET.011.2000263.

[45] J. Li et al., "Blockchain Assisted Decentralized Federated Learning (BLADE-FL): Performance Analysis and Resource Allocation," in IEEE Transactions on Parallel and Distributed Systems, vol. 33, no. 10, pp. 2401-2415, 1 Oct. 2022, DOI: 10.1109/TPDS.2021.3138848.

[46] T. A. K. Pham, "BlockFL: blockchain-enabled decentralized federated learning and model trading," dr.ntu.edu.sg, 2022, Accessed: Aug. 29, 2023. [Online]. Available: https://dr.ntu.edu.sg/handle/10356/156495

[47] H. B. Desai, M. S. Ozdayi, and M. Kantarcioglu, "BlockFLA: Accountable Federated Learning via Hybrid Blockchain Architecture," Proceedings of

the Eleventh ACM Conference on Data and Application Security and Privacy, Apr. 2021, DOI: https://doi.org/10.1145/3422337.3447837.

[48] X. Wu, Z. Wang, J. Zhao, Y. Zhang and Y. Wu, "FedBC: Blockchain-based Decentralized Federated Learning," 2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), Dalian, China, 2020, pp. 217-221, DOI: 10.1109/ICAICA50127.2020.9182705.

[49] H. Chen, S. A. Asif, J. Park, C. Shen, and M. Bennis, "Robust Blockchained Federated Learning with Model Validation and Proof-of-Stake Inspired Consensus," 2021. [Online]. Available: https://doi.org/10.48550/arXiv.2101.03300.

[50] P. Chinnasamy, A. Albakri, M. Khan, A. Ambeth Raja, A. Kiran, and J. Chinna Babu, "Smart Contract-Enabled Secure Sharing of Health Data for a Mobile Cloud-Based E-Health System," vol. 13, no. 6, pp. 3970–3970, Mar. 2023, doi: https://doi.org/10.3390/app13063970.

[51] M. D. Ansari, A. Sharma, M. Khan, and L. Ting, "A secure framework for IoT-based smart climate agriculture system: Toward blockchain and edge computing," philpapers.org, 2022. https://philpapers.org/rec/ANSASF (accessed Mar. 04, 2024).

[52] A. Padma and M. Ramaiah, "Blockchain Based an Efficient and Secure Privacy Preserved Framework for Smart Cities," in IEEE Access, vol. 12, pp. 21985-22002, 2024, doi: 10.1109/ACCESS.2024.3364078.

[53] A. Sokol and M. Hogan, 'NIST Cloud Computing Standards Roadmap'. Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD, 22-Jul-2013.

[54] R. Mahmud, R. Kotagiri, and R. Buyya, "Fog Computing: A Taxonomy, Survey and Future Directions," in "Internet of Everything: Algorithms, Methodologies, Technologies and Perspectives," B. Di Martino, K.-C. Li, L. T. Yang, and A. Esposito, Eds. Singapore: Springer, 2018, pp. 103-130. [Online]. Available: https://doi.org/10.1007/978-981-10-5861-5_5.

[55] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li and H. Vincent Poor, "Federated Learning for Internet of Things: A Comprehensive Survey," in IEEE Communications Surveys & Tutorials, vol. 23, no. 3, pp. 1622-1658, thirdquarter 2021, DOI: 10.1109/COMST.2021.3075439.

[56] T. Li, A. K. Sahu, A. Talwalkar and V. Smith, "Federated Learning: Challenges, Methods, and Future Directions," in IEEE Signal Processing Magazine, vol. 37, no. 3, pp. 50-60, May 2020, DOI: 10.1109/MSP.2020.2975749.

[57] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," Knowledge-Based Systems, vol. 216, p. 106775, Mar. 2021, DOI: https://doi.org/10.1016/j.knosys.2021.106775.

[58] S. N. Khan, F. Loukil, C. Ghedira-Guegan, E. Benkhelifa, and A. Bani-Hani, "Blockchain smart contracts: Applications, challenges, and future trends," Peer-to-Peer Networking and Applications, vol. 14, no. 1, Apr. 2021, DOI: https://doi.org/10.1007/s12083-021-01127-0.

[59] M. Alharby and A. van Moorsel, "Blockchain Based Smart Contracts: A Systematic Mapping Study," in Computer Science & Information Technology (CS & IT), Academy & Industry Research Collaboration Center (AIRCC), Aug. 2017. [Online]. Available: https://doi.org/10.5121%2Fcsit.2017.71011.

[60] Z. Wang and Q. Hu, "Blockchain-based federated learning: A comprehensive survey," arXiv preprint arXiv:2110.02182, 2021. [Online]. Available: https://doi.org/10.48550/arXiv.2110.02182.

[61] P. Tasca and C. J. Tessone, "A Taxonomy of Blockchain Technologies: Principles of Identification and Classification", ledger, vol. 4, Feb. 2019, DOI: https://doi.org/10.5195/ledger.2019.140.

[62] L. Ismail and H. Materwala, "A Review of Blockchain Architecture and Consensus Protocols: Use Cases, Challenges, and Solutions," Symmetry, vol. 11, no. 10, p. 1198, Sep. 2019, DOI: https://doi.org/10.3390/sym11101198.

[63] Y. Xiao, N. Zhang, W. Lou and Y. T. Hou, "A Survey of Distributed Consensus Protocols for Blockchain Networks," in IEEE Communications Surveys & Tutorials, vol. 22, no. 2, pp. 1432-1465, Secondquarter 2020, DOI: 10.1109/COMST.2020.2969706.

[64] A. Singh, G. Kumar, R. Saha, M. Conti, M. Alazab, and R. Thomas, "A survey and taxonomy of consensus protocols for blockchains," Journal of Systems Architecture, vol. 127, p. 102503, 2022. [Online]. Available: https://doi.org/10.1016/j.sysarc.2022.102503.

[65] M. R. Islam, M. M. Rahman, M. Mahmud, M. A. Rahman, M. H. S. Mohamad and A. H. Embong, "A Review on Blockchain Security Issues and Challenges," 2021 IEEE 12th Control and System Graduate Research Colloquium (ICSGRC), Shah Alam, Malaysia, 2021, pp. 227-232, DOI: 10.1109/ICSGRC53186.2021.9515276.

[66] A. Alghamdi et al., "Blockchain Empowered Federated Learning Ecosystem for Securing Consumer IoT Features Analysis," Sensors, vol. 22, no. 18, p. 6786, Sep. 2022, DOI: 10.3390/s22186786.

[67] I. Kholod et al., "Open-Source Federated Learning Frameworks for IoT: A Comparative Review and Analysis," Sensors, vol. 21, no. 1, p. 167, Dec. 2020, DOI: 10.3390/s21010167.

[68] Y. I. Alzoubi, A. Gill, and A. Mishra, "A systematic review of the purposes of Blockchain and fog computing integration: classification and open issues," Journal of Cloud Computing, vol. 11, no. 1, p. 80, 2022. [Online]. Available: https://doi.org/10.1186/s13677-022-00353-y.

[69] X. Yin, Y. Zhu, and J. Hu, "A Comprehensive Survey of Privacy-Preserving Federated Learning: A Taxonomy, Review, and Future Directions," ACM Comput. Surv., vol. 54, no. 6, art. no. 131, Jul. 2022, [Online]. Available: https://doi.org/10.1145/3460427.

[70] E. Lee, Y. -D. Seo, S. -R. Oh and Y. -G. Kim, "A Survey on Standards for Interoperability and Security in the Internet of Things," in IEEE Communications Surveys & Tutorials, vol. 23, no. 2, pp. 1020-1047, Second quarter 2021, DOI: 10.1109/COMST.2021.3067354.

[71] International Organization for Standardization, "Standards," ISO, 2019. https://www.iso.org/standards.html.

[72] "Standards | NIST," NIST, May 30, 2023. https://www.nist.gov/standards.

[73] M. Fowler, "Domain-Specific Languages," Addison-Wesley Signature Series (Fowler), Pearson Education, 2010. [Online]. Available: https://books.google.hu/books?id=ri1muolw_YwC.

[74] A. Reinhardt and K. Milzow, "Evaluation in research and research funding organizations: European practices," in ESF Member Organisation Forum on Evaluation of Publicly Funded Research, 2012.

[75] L. C. Briand, C. M. Differding, and H. D. Rombach, "Practical guidelines for measurement-based process improvement," Software Process: Improvement and Practice, vol. 2, no. 4, pp. 253-280, 1996.

[76] X. Huang, Y. Ren, Y. He and Q. Chen, "Malicious Models-based Federated Learning in Fog Computing Networks," 2022 14th International Conference on Wireless Communications and Signal Processing (WCSP), Nanjing, China, 2022, pp. 192-196, doi: 10.1109/WCSP55476.2022.10039266.

[77] Y. Qu et al., "Decentralized Privacy Using Blockchain-Enabled Federated Learning in Fog Computing," in IEEE Internet of Things Journal, vol. 7, no. 6, pp. 5171-5183, June 2020, doi: 10.1109/JIOT.2020.2977383.

[78] J. Alotaibi and L. Alazzawi, "PPIoV: A Privacy Preserving-Based Framework for IoV- Fog Environment Using Federated Learning and Blockchain," 2022 IEEE World AI IoT Congress (AIIoT), Seattle, WA, USA, 2022, pp. 597-603, doi: 10.1109/AIIoT54504.2022.9817205.

[79] M. Aloqaily, I. Al Ridhawi, F. Karray and M. Guizani, "Towards Blockchain-based Hierarchical Federated Learning for Cyber-Physical Systems," 2022 International Balkan Conference on Communications and Networking (BalkanCom), Sarajevo, Bosnia and Herzegovina, 2022, pp. 46-50, doi: 10.1109/BalkanCom55633.2022.9900546.

[80] M. J. Baucas, P. Spachos and K. N. Plataniotis, "Federated Learning and Blockchain-Enabled Fog-IoT Platform for Wearables in Predictive Healthcare," in IEEE Transactions on Computational Social Systems, vol. 10, no. 4, pp. 1732-1741, Aug. 2023, doi: 10.1109/TCSS.2023.3235950.

[81] A. Lakhan et al., "Federated-Learning Based Privacy Preservation and Fraud-Enabled Blockchain IoMT System for Healthcare," in IEEE Journal of Biomedical and Health Informatics, vol. 27, no. 2, pp. 664-672, Feb. 2023, doi: 10.1109/JBHI.2022.3165945.

[82] S. Otoum, I. Al Ridhawi and H. T. Mouftah, "Blockchain-Supported Federated Learning for Trustworthy Vehicular Networks," GLOBECOM 2020 - 2020 IEEE Global Communications Conference, Taipei, Taiwan, 2020, pp. 1-6, doi: 10.1109/GLOBECOM42002.2020.9322159.

[83] M. Abdel-Basset, N. Moustafa and H. Hawash, "Privacy-Preserved Cyberattack Detection in Industrial Edge of Things (IEoT): A Blockchain-Orchestrated Federated Learning Approach," in IEEE Transactions on Industrial Informatics, vol. 18, no. 11, pp. 7920-7934, Nov. 2022, doi: 10.1109/TII.2022.3167663.

[84] J. A. Khan and K. Ozbay, "AFFIRM: Privacy-by-Design Blockchain for Mobility Data in Web3 using Information Centric Fog Networks with Collaborative Learning," 2023 International Conference on Computing, Networking and Communications (ICNC), Honolulu, HI, USA, 2023, pp. 456-462, doi: 10.1109/ICNC57223.2023.10074160.

[85] I. A. Ridhawi, M. Aloqaily, A. Abbas and F. Karray, "An Intelligent Blockchain-Assisted Cooperative Framework for Industry 4.0 Service Management," in IEEE Transactions on Network and Service Management, vol. 19, no. 4, pp. 3858-3871, Dec. 2022, doi: 10.1109/TNSM.2022.3217395.

[86] S. R. Pokhrel, “Federated learning meets blockchain at 6G edge,” Proceedings of the 2nd ACM MobiCom Workshop on Drone Assisted Wireless Communications for 5G and Beyond, Sep. 2020, doi: 10.1145/3414045.3415949.

[87] G. Qu, H. Wu, and N. Cui, “Joint blockchain and federated learning-based offloading in harsh edge computing environments,” Proceedings of the International Workshop on Big Data in Emergent Distributed Environments, Jun. 2021, doi: 10.1145/3460866.3461765.

[88] J. Xu, J. Lin, W. Liang, and K.-C. Li, “Privacy preserving personalized blockchain reliability prediction via federated learning in IoT environments,” Cluster Computing, Sep. 2021, doi:10.1007/s10586-021-03399-w.

[89] I. Ullah, X. Deng, X. Pei, P. Jiang, and H. Mushtaq, “A verifiable and privacy-preserving blockchain-based federated learning approach,” Peer-to-Peer Networking and Applications, vol. 16, no. 5, pp. 2256–2270, Jul. 2023, doi: 10.1007/s12083-023-01531-8.

[90] Y. Wang, Y. Tian, X. Yin, and X. Hei, “A trusted recommendation scheme for privacy protection based on federated learning,” CCF Transactions on Networking, vol. 3, no. 3–4, pp. 218–228, Nov. 2020, doi: 10.1007/s42045-020-00045-8.

[91] S. K. Vishwakarma et al., “Secure federated learning architecture for fuzzy classifier in healthcare environment,” Soft Computing, Jul. 2023, doi: https://doi.org/10.1007/s00500-023-08629-4.

[92] W. Zheng, Y. Cao, and H. Tan, “Secure sharing of industrial IoT data based on distributed trust management and trusted execution environments: a federated learning approach,” Neural Computing and Applications, vol. 35, no. 29, pp. 21499–21509, Mar. 2023, doi: 10.1007/s00521-023-08375-6.

[93] Lakhan, Abdullah, et al. Federated learning-aware multi-objective modeling and blockchain-enable system for IIoT applications. Computers and Electrical Engineering, 2022, vol. 100, p. 107839.

[94] Wang, Weilong, et al. Privacy protection federated learning system based on blockchain and edge computing in mobile crowdsourcing. Computer Networks, 2022, vol. 215, p. 109206.

[95] Wan, Y., Qu, Y., Gao, L., & Xiang, Y. (2022). Privacy-preserving blockchain-enabled federated learning for B5G-Driven edge computing. Computer Networks, 204, 108671.

[96] Sharma, P. K., Park, J. H., & Cho, K. (2020). Blockchain and federated learning-based distributed computing defence framework for sustainable society. Sustainable Cities and Society, 59, 102220.

[97] Baker, Thar, et al. "A blockchain-based Fog-oriented lightweight framework for smart public vehicular transportation systems." Computer Networks 203 (2022): 108676.

[98] Sezer, Bora Bugra, Hasret Turkmen, and Urfat Nuriyev. "PPFchain: A novel framework privacy-preserving blockchain-based federated learning method for sensor networks." Internet of Things 22 (2023): 100781.

[99] A. Smahi et al., “BV-ICVs: A privacy-preserving and verifiable federated learning framework for V2X environments using blockchain and zkSNARKs,” Journal of King Saud University - Computer and Information Sciences, vol. 35, no. 6, p. 101542, Jun. 2023, doi: 10.1016/j.jksuci.2023.03.020.

[100] K. Wang et al., “A trusted consensus fusion scheme for decentralized collaborated learning in massive IoT domain,” Information Fusion, vol. 72, pp. 100–109, Aug. 2021, doi: 10.1016/j.inffus.2021.02.011.

[101] X. Huang, L. Han, D. Li, K. Xie, and Y. Zhang, “A reliable and fair federated learning mechanism for mobile edge computing,” Computer Networks, vol. 226, p. 109678, May 2023, doi: 10.1016/j.comnet.2023.109678.

[102] D. Hamouda, M. A. Ferrag, N. Benhamida, and H. Seridi, “PPSS: A privacy-preserving secure framework using blockchain-enabled federated deep learning for Industrial IoTs,” Pervasive and Mobile Computing, p. 101738, Dec. 2022, doi: 10.1016/j.pmcj.2022.101738.

[103] J. Akram, M. Umair, R. H. Jhaveri, M. N. Riaz, H. Chi, and S. Malebary, “Chained-Drones: Blockchain-based privacy-preserving framework for secure and intelligent service provisioning in Internet of Drone Things,” Computers and Electrical Engineering, vol. 110, p. 108772, Sep. 2023, doi: 10.1016/j.compeleceng.2023.108772.

[104] A. Heidari, D. Javaheri, S. Toumaj, N. J. Navimipour, M. Rezaei, and M. Unal, “A new lung cancer detection method based on the chest CT images using Federated Learning and blockchain systems,” Artificial Intelligence in Medicine, vol. 141, p. 102572, Jul. 2023, doi:10.1016/j.artmed.2023.102572.

[105] Z. Kuang and C. Chen, “Research on smart city data encryption and communication efficiency improvement under federated learning framework,” Egyptian Informatics Journal, vol. 24, no. 2, pp. 217–227, Jul. 2023, doi:10.1016/j.eij.2023.02.005.

[106] X. Su, L. An, Z. Cheng, and Y. Weng, “Cloud–edge collaboration-based bi-level optimal scheduling for intelligent healthcare systems,” Future Generation Computer Systems, vol. 141, pp. 28–39, Apr. 2023, doi:10.1016/j.future.2022.11.005.

[107] H. Moudoud, S. Cherkaoui and L. Khoukhi, "Towards a Secure and Reliable Federated Learning using Blockchain," 2021 IEEE Global Communications Conference (GLOBECOM), Madrid, Spain, 2021, pp. 01-06, doi: 10.1109/GLOBECOM46510.2021.9685388.

[108] S. Otoum, I. A. Ridhawi and H. Mouftah, "Securing Critical IoT Infrastructures With Blockchain-Supported Federated Learning," in IEEE Internet of Things Journal, vol. 9, no. 4, pp. 2592-2601, 15 Feb.15, 2022, doi: 10.1109/JIOT.2021.3088056.

[109] K. N. Mishra, V. Bhattacharjee, S. Saket, and S. P. Mishra, “Security provisions in smart edge computing devices using blockchain and machine learning algorithms: a novel approach,” Cluster Computing, Nov. 2022, doi:10.1007/s10586-022-03813-x.

[110] Y. Xu et al., "BESIFL: Blockchain-Empowered Secure and Incentive Federated Learning Paradigm in IoT," in IEEE Internet of Things Journal, vol. 10, no. 8, pp. 6561-6573, 15 April15, 2023, doi: 10.1109/JIOT.2021.3138693.

[111] Q. Wang, Y. Guo, X. Wang, T. Ji, L. Yu and P. Li, "AI at the Edge: Blockchain-Empowered Secure Multiparty Learning With Heterogeneous Models," in IEEE Internet of Things Journal, vol. 7, no. 10, pp. 9600-9610, Oct. 2020, doi: 10.1109/JIOT.2020.2987843.

[112] S. Otoum, I. A. Ridhawi and H. Mouftah, "A Federated Learning and Blockchain-Enabled Sustainable Energy Trade at the Edge: A Framework for Industry 4.0," in IEEE Internet of Things Journal, vol. 10, no. 4, pp. 3018-3026, 15 Feb.15, 2023, doi: 10.1109/JIOT.2022.3140430.

[113] M. Sarhan, W. W. Lo, S. Layeghy, and M. Portmann, “HBFL: A hierarchical blockchain-based federated learning framework for collaborative IoT intrusion detection,” Computers and Electrical Engineering, vol. 103, p. 108379, Oct. 2022, doi: 10.1016/j.compeleceng.2022.108379.

[114] S. K. Singh, L. T. Yang, and J. H. Park, “FusionFedBlock: Fusion of blockchain and federated learning to preserve privacy in industry 5.0,” Information Fusion, vol. 90, pp. 233–240, Feb. 2023, doi:10.1016/j.inffus.2022.09.027.

WILSON VALDEZ SOLIS is a Ph.D. student in Computer Science at the Department of Software Engineering, University of Szeged, Hungary. He was awarded in 2022 with the Stipendium Hungaricum Scholarship to perform doctoral studies in Hungary. Nowadays, he is a member of the IoT-Cloud Research Group at the Department. He received a B.S.Eng degree in Electronics and Telecommunications from the Universidad de Cuenca, Ecuador in 2017, and M.S. degree in Industry 4.0 from the Universidad Internacional La Rioja (UNIR) in 2020. From 2018 to 2022, he was a Research Assistant at the Group of Investigation on Information Technologies (GIIT) at the Universidad de Cuenca, participating in CEPRA projects in the period 2018-2022. He is the author of 20 articles and co-developer of Assistive Technology devices. His research interests include Cloud Computing, the Internet of Things, Edge Computing, Fog Computing, Blockchain, Machine Learning, and Federated Learning.

JUAN MARCELO PARRA-ULLAURI earned his bachelor's degree in Electronics and Telecommunications Engineering from the Universidad de Cuenca, Ecuador in 2017. In 2022, he is PhD in Computer Science from Aston University in the UK. He currently works as a Senior Research Associate at the Smart Internet Lab, focusing on researching AI/ML for Networked Systems. His research interests include the Internet of Things, Distributed Machine Learning, Cloud Computing, Explainability in Autonomous Systems, and Data Engineering. He is the author of more than 20 articles. He actively collaborates on academy-industry projects, contributes to open-source FL projects, and leads research activities within EU and UK-funded projects.

ATTILA KERTESZ is an Associate Professor with the Department of Software Engineering, University of Szeged, Hungary, leading the IoT-Cloud Research Group at the Department. His research interests include federative management of IoT, Blockchain, Fog and Cloud systems, and data management issues of distributed systems in general. He was the leader of the National Project OTKA FK 131793 financed by the Hungarian Scientific Research Fund, and he was the leader of the Fog-Block4Trust sub-grant project of the TruBlo EU H2020 project and a work package leader of the GINOP IoLT Project financed by the Hungarian Government and the European Regional Development Fund. He has also been a Management Committee Member of the CERCIRAS and INDAIR-POLLNET EU COST actions, while he has participated in several successful European projects, including ENTICE EU H2020, COST IC1304, COST IC0805, SHIWA, S-Cube EU FP7, and the CoreGRID EU FP6 Network of Excellence projects. He has been a member of numerous program committees for European conferences and workshops and published over 150 scientific papers having more than 2000 citations.

• • •