electronics

MDPI

Review

A Systematic Literature Review on the Use of Federated Learning and Bioinspired Computing

Rafael Marin Machado de Souza [1,2], Andrew Holm [3], Márcio Biczyk [2] and Leandro Nunes de Castro [1,2,3,*]

[1] School of Technology, State University of Campinas (Unicamp), R. Paschoal Marmo, 1888-Jd. Nova Itália, Limeira 13484-332, SP, Brazil; rafael.marin@hc.fm.usp.br
[2] In.lab-InovaHC, Clinics Hospital of Medicine Faculty of University of Sao Paulo (USP), R. Dr. Ovídio Pires de Campos, 75-Cerqueira César, São Paulo 05401-000, SP, Brazil; marcio.biczyk@hc.fm.usp.br
[3] Department of Computing and Software Engineering, Florida Gulf Coast University (FGCU), 10501 Fgcu Blvd. S, Fort Myers, FL 33965, USA; adholm3723@eagle.fgcu.edu
* Correspondence: ldecastrosilva@fgcu.edu

Abstract: Federated learning (FL) and bioinspired computing (BIC), two distinct, yet complementary fields, have gained significant attention in the machine learning community due to their unique characteristics. FL enables decentralized machine learning by allowing models to be trained on data residing across multiple devices or servers without exchanging raw data, thus enhancing privacy and reducing communication overhead. Conversely, BIC draws inspiration from nature to develop robust and adaptive computational solutions for complex problems. This paper explores the state of the art in the integration of FL and BIC, introducing BIC techniques and discussing the motivations for their integration with FL. The convergence of these fields can lead to improved model accuracy, enhanced privacy, energy efficiency, and reduced communication overhead. This synergy addresses inherent challenges in FL, such as data heterogeneity and limited computational resources, and opens up new avenues for developing more efficient and autonomous learning systems. The integration of FL and BIC holds promise for various application domains, including healthcare, finance, and smart cities, where privacy-preserving and efficient computation is paramount. This survey provides a systematic review of the current research landscape, identifies key challenges and opportunities, and suggests future directions for the successful integration of FL and BIC.

Keywords: federated learning; bioinspired computing; natural computing; evolutionary algorithm; particle swarm optimization

check for updates

Citation: Marin Machado de Souza, R.; Holm, A.; Biczyk, M.; de Castro, L.N. A Systematic Literature Review on the Use of Federated Learning and Bioinspired Computing. *Electronics* **2024**, *13*, 3157. https://doi.org/10.3390/electronics13163157

Academic Editor: Heung-Il Suk

Received: 28 June 2024
Revised: 1 August 2024
Accepted: 3 August 2024
Published: 10 August 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Federated learning (FL) is an emerging paradigm in machine learning that enables multiple decentralized devices, such as mobile devices or servers, to collaboratively train a shared model while keeping the data locally. Instead of centralizing the data, the parameters of the model are shared and updated across devices. This approach is particularly useful for applications involving sensitive data, such as healthcare and finance, where data privacy and security are paramount [1–3]. FL has become an important paradigm due to its role in protecting privacy, its efficiency, its customization potential, and its capability of dealing with heterogeneous data. Despite all that, FL faces several challenges that need to be addressed for its effective implementation. These challenges include high communication costs due to frequent model updates, heterogeneity in systems and data, ensuring robust privacy and security, managing limited resources on edge devices, and achieving efficient model training and performance [4–6].

Bioinspired or nature-inspired computing is a subfield of computer science that leverages principles and mechanisms observed in biological systems to develop algorithms for solving complex problems. These algorithms cover various subareas, including evolutionary computing (e.g., Genetic Algorithms and Differential Evolution), swarm intelligence

(e.g., Particle Swarm Optimization and Ant Colony Optimization), and others like Artificial Immune Systems [7–11]. Such algorithms are collectively called bioinspired algorithms or bioinspired computing (BIC) and are characterized by their adaptability, robustness, and ability to find near-optimal solutions in complex search spaces. When used in conjunction with FL, bioinspired algorithms can significantly enhance performance by optimizing communication processes, managing heterogeneity, improving privacy and security, and ensuring efficient resource utilization.

The present review is the first one to focus on the use of BIC in conjunction with FL. To survey the literature, we used a systematic review approach following the PRISMA guidelines [12]. Three research questions were proposed aimed at finding which bioinspired algorithm is being used and what is the research context; what type of architecture, data partitioning, privacy mechanism, aggregation method, and system heterogeneity are observed in the works combining FL and BIC; and what FL core challenge is being addressed by the BIC algorithm. The search queries were applied over the ACM Digital Library, SCOPUS, IEEE Xplore, and Web of Science. From a total of 165 papers retrieved, 21 remained after the application of the inclusion and exclusion criteria.

The review is divided into three main topics, each one related to one research question. The survey results reveal that Particle Swarm Optimization (PSO) is the most widely used bioinspired algorithm, employed in 85.71% of the reviewed papers. This is mainly due to its simplicity, effectiveness, and adaptability in distributed optimization tasks, which is one of the core features of FL. The majority of the reviewed papers (approximately 71.4%) utilize a centralized federated learning architecture; the majority also use horizontal partitioning, and federated averaging is a common privacy mechanism. In terms of the aggregation method, federated averaging and optimization-based aggregation are the most common ones. Finally, the bioinspired algorithms are usually employed to address the core FL challenges of communication costs, system heterogeneity, privacy and security issues, and model training and performance.

Although federated learning is considered to have emerged only in the year 2016 with a work by Google [13], the literature is filled with reviews and surveys about FL, such as [1–6,14–16], to name but a few. Some of these reviews are broad in scope or focus on one specific topic and are different from our review in many respects. The main contributions of this paper are as follows:

- The first comprehensive survey of studies that integrate bioinspired algorithms with the FL challenges, analyzing their characteristics, methods, and effectiveness.
- The identification of the main BIC algorithms used in conjunction with FL and their context.
- The identification and discussion of the core challenges addressed by these hybrid approaches.
- The proposal of future research directions to further enhance the applicability and performance of BIC in FL environments.

The paper is organized as follows. Section 1 introduces the context and motivation for the survey. Section 2 provides an overview of federated learning, highlighting its core challenges. Section 3 introduces bioinspired computing, detailing its main subareas and characteristics. A focus is given to the Genetic Algorithm (GA) and the Particle Swarm Optimization (PSO) algorithm, the two most commonly used algorithms within the FL+BIC hybrid approaches. Section 4 describes the survey methodology used to select and analyze the relevant studies. Section 5 presents the survey results, divided into the BIC algorithms used and the work context, the FL characteristics, and the FL core challenges. Section 6 discusses open challenges and future directions for the hybridization of FL and BIC. Finally, Section 7 concludes the paper with a summary of key findings and implications for future research.

2. Federated Learning

Federated learning is a machine learning approach that allows for training models across multiple decentralized devices or servers while keeping the data localized. Instead of sending data from various devices to a central server for training, the training process happens locally on each device, and only model updates (such as weights) are sent to the central server or aggregator [1–3,14].

Since the data remain on the devices and are not centralized, federated learning can help preserve user privacy. Personal data do not need to leave the device, reducing privacy concerns associated with centralized data storage [17,18]. It can be more efficient in terms of bandwidth and computation, especially in scenarios where data are abundant, but sending them to a central server is impractical due to limitations in network bandwidth or data privacy concerns [19].

By allowing training on distributed data sources, federated learning supports decentralized ecosystems, enabling collaboration and learning across different organizations or individuals without requiring them to share sensitive data. Since models are trained across diverse devices, federated learning can capture a wider range of data distributions and characteristics, potentially leading to more robust and generalizable models [4,15].

Federated learning has applications in various domains, such as healthcare (where patient data privacy is critical), mobile devices (to improve personalized user experiences without compromising data privacy), and edge computing (where resources are limited) [15,16].

The literature on FL is broad and has been growing quickly and significantly over the past few years. There are a number of reviews and survey works on federated learning, and each one of them describes the area and focuses the review on one or more specific topics. Good and recent examples of review works on FL include [1–6,20]. This section provides a brief overview of some specific aspects about federated learning based on their relevance for the field and usefulness for the present systematic review. More specifically, the following subsections will discuss the main FL architecture types, the main characterizations, and the main challenges of the field.

2.1. Architecture

Federated learning architectures can vary based on the specific requirements of the application and the underlying infrastructure. Several common architectures and variations have been proposed and utilized in practice. In the original approach, centralized federated learning (CFL), a central server coordinates everything. Devices train models on their own data and send updates to the server, which combines them to improve the overall model. This is efficient, but requires a central point of control, which might not always be available [13].

Decentralized federated learning (DFL) ditches the central server altogether, and devices talk directly to each other, exchanging model updates. This is more robust in case of failures, but requires more complex coordination, potentially using blockchain technology [21]. Hierarchical federated learning (HFL) tackles large-scale deployments by using a tiered structure. Lower tiers handle local updates on individual devices, while higher tiers manage the bigger picture. This is ideal for complex networks like edge computing environments [22]. Finally, cross-silo federated learning (CSFL) lets organizations with regulations preventing data sharing collaborate. Each organization keeps control of its data, but contributes to a shared model through secure methods like differential privacy. This is useful in areas like healthcare, finance, and data analysis across multiple institutions [23,24].

2.2. Characterization of Federated Learning

Characterizing FL based on various aspects such as data partitioning, privacy mechanisms, aggregation method, and methods for addressing heterogeneity provides a comprehensive view of the different dimensions along which federated learning can be understood and implemented.

This section deals with the design principles of data partitioning, privacy mechanisms, aggregation methods, and system heterogeneity, as summarized in Figure 1. This diagram takes each of these four characteristics and unfolds them into categories, as will be described in the following sections.

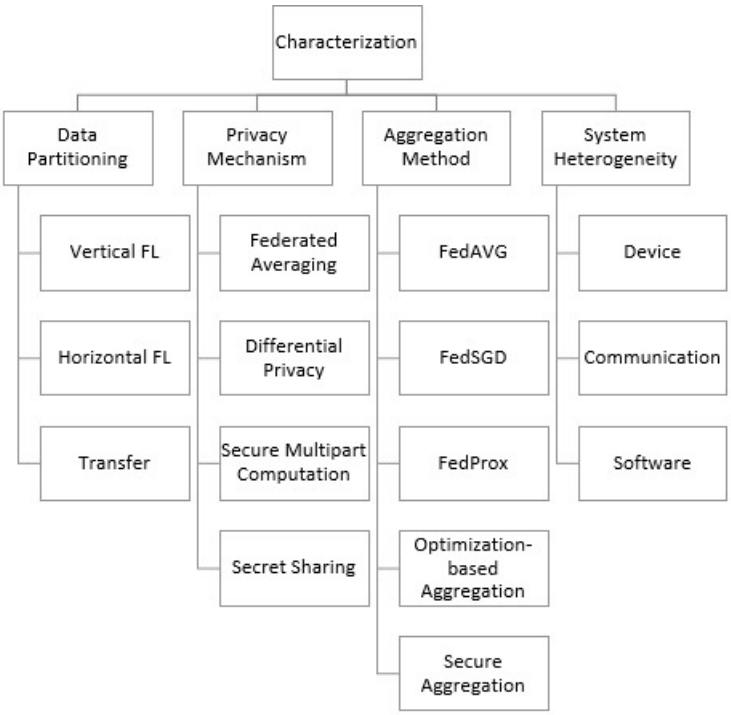


Figure 1. FL characterization and its branches.

2.2.1. Data Partitioning

Data partitioning is an aspect of federated learning that refers to the way the datasets are divided and distributed among the participating devices, as represented in Figure 2.

Within the domain of federated learning, various data partitioning strategies exist to facilitate collaborative model training while preserving data privacy. Horizontal federated learning (HFL) adopts a featurewise partitioning approach, in which individual devices or servers hold data about distinct samples (e.g., different users on a social media platform), but share the same set of features (e.g., posts, likes, etc.). This allows the training of local models on these independent datasets with subsequent aggregation to build a robust global model [25].

Vertical federated learning (VFL), by contrast, employs a samplewise partitioning strategy. In this scenario, devices or servers store data for identical sets of samples (e.g., the same users), but possess distinct feature sets. For instance, one device might hold demographic information, while another contains behavioral data (purchases, website visits) for the same user base. VFL enables secure collaboration on a model without revealing complete user profiles on any single device [26,27].

Finally, Federated transfer learning (FTL) leverages the established concept of transfer learning within a federated framework. This technique is particularly useful when data distributions across different domains (source and target) exhibit related, yet non-identical, characteristics. In this case, a pre-trained model is developed on a publicly available dataset (source domain), and then undergoes fine-tuning on individual devices (target domain) utilizing their local data. FTL empowers devices to benefit from pre-trained knowledge while maintaining data privacy [28].

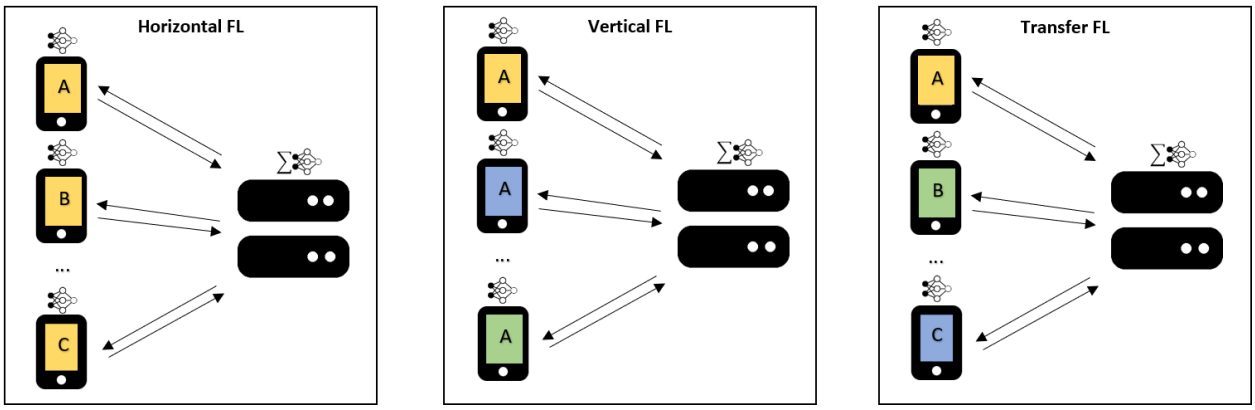


Figure 2. A representation of FL data partitioning. In horizontal FL, distinct samples share the same set of features. In vertical FL, the same sample shares distinct features. In federated transfer learning, there are data with different samples and features.

2.2.2. Privacy Mechanism

Federated learning offers a way to train machine learning models collaboratively without directly sharing the data themselves. This is particularly useful when dealing with sensitive information where privacy is a concern. *Federated averaging* (FedAVG), although it is the principal aggregation method, is also considered a privacy mechanism, ensuring only the model updates are transmitted from devices, keeping the raw data on the device. The server aggregates these updates to improve the global model without ever seeing the individual data points [13,17].

Another way to enhance the data privacy is through *differential privacy* (DP) techniques, which add noise to the updates during training. This noise is introduced by the inclusion or exclusion of a single data point, with minimal impact on the final model. This ensures that individual data points cannot be distinguished in the aggregated data [29].

Secure multiparty computation (SMC) techniques allow multiple devices in federated learning to jointly train a model over their inputs while keeping those inputs private [30]. One specific SMC technique is homomorphic encryption (HE) [31]. HE enables computations directly over encrypted data without decrypting them first. Additionally, *Secret Sharing* is a technique that uses cryptography to split a secret message into multiple parts. These parts are then given to different devices. No individual piece by itself reveals anything about the original secret. Only by combining a certain number of these pieces, as defined when the secret is split, can the original secret be recovered [32]. These techniques can be combined with each other in different machine learning models.

2.2.3. Aggregation Method

Federated learning (FL) enables collaborative machine learning training without directly sharing private data between participating devices. A crucial step in FL is the aggregation of local model updates received from devices. The aggregation method significantly affects the convergence, communication efficiency, and privacy guarantees of the FL training process [33].

Federated learning enables collaborative machine learning training without directly sharing private data across participating devices. A crucial step in FL is the aggregation of local model updates received from devices. The aggregation method significantly impacts the convergence, communication efficiency, and privacy guarantees of the FL training process [33].

The most widely used and foundational aggregation method in FL is *federated averaging* (FedAvg) [13]. It computes the average of local model updates received from a subset of participating devices. This method is simple to implement and offers good convergence properties. However, FedAvg can be susceptible to issues like non-independent and

identically distributed (non-IID) data distributions across devices, leading to performance degradation [34].

Federated stochastic gradient descent (FedSGD) is a variant of FedAvg that utilizes stochastic gradient descent (SGD) for local model updates on participating devices. Similar to FedAvg, it then averages the local updates to form the global model. This approach can potentially achieve faster convergence compared to FedAvg, but may also be more sensitive to non-IID data [35].

Another extension of FedAvg, FedProx introduces a proximity term to the objective function during aggregation. This term encourages similar updates from devices with similar data distributions, mitigating the impact of non-IID data heterogeneity across devices [36].

Optimization-based aggregation (OBA) is a category of methods that leverages optimization techniques to find the best global model considering the contributions from local models. These methods can be more complex to implement, but may offer better convergence and robustness compared to simpler averaging approaches [37].

Privacy-preserving aggregation techniques are crucial for FL, especially when dealing with sensitive data. *Secure aggregation* (SA) methods leverage cryptographic techniques like homomorphic encryption or *secure multiparty computation* (SMC) to perform aggregation, while keeping individual updates private [17,33]. However, these methods often introduce additional computational overhead.

The choice of an appropriate aggregation method depends on various factors like the application domain, the data characteristics, the communication constraints, and the privacy requirements. Recent research explores novel aggregation methods that combine these strategies or utilize techniques like federated meta-learning to adapt the aggregation process dynamically [25].

2.2.4. System Heterogeneity

Practical FL systems face significant challenges due to system heterogeneity across participating devices. This section explores the various aspects of system heterogeneity and their impact on FL performance.

System heterogeneity in FL encompasses three key aspects [4,13]. The first is device heterogeneity, where devices participating in FL can vary significantly in terms of computational resources (CPU, memory), storage capacity, and battery power. This heterogeneity can lead to imbalances in the training process, where devices with limited resources struggle to keep pace with powerful devices. Communication heterogeneity, where network connectivity can vary considerably across devices. Limited bandwidth or high latency can significantly slow down the communication between devices and the central server, hindering the training process. Software heterogeneity is the third, where devices may run on different operating systems and have varying software versions. This inconsistency can create compatibility issues and hinder the seamless operation of the FL protocol.

These factors combined can lead to several challenges in FL, including [4,5,13] slow convergence, where the heterogeneity can slow down the convergence of the global model, as slower devices contribute less frequently or with updates of lower quality, Accuracy Degradation, where the overall accuracy of the model can be degraded if updates from resource-constrained devices significantly deviate from the optimal direction, and unfairness: in extreme cases, heterogeneity can lead to unfairness, where devices with limited resources contribute minimally to the training process and receive less benefit from the resulting model.

2.3. Core Challenges

Federated learning (FL) faces several core challenges that need to be addressed for its effective implementation. *Communication cost* is a significant concern, as FL relies on exchanging model updates between devices and a central server, which can be bandwidth-intensive, especially for geographically dispersed or large models. There is also a trade-off

between communication cost and accuracy in FL. Setting a communication budget as a restriction criterion in communication-focused FL benchmarks is worth considering [4]. In cross-device FL, only a few devices are often active during an iteration, necessitating an in-depth analysis of the asynchronous communication scheme, where devices become active based on certain events [38,39].

System heterogeneity is another critical issue, as FL environments may encompass a wide range of devices and algorithms with varying computational, storage, and network capabilities. Differences in operating systems, hardware architectures, software libraries, data distribution, communication, and energy constraints across devices may lead to imbalanced participation, compromised performance, and slow training times [34].

Privacy and security are paramount in FL. Although data privacy is a remarkable characteristic of FL, data leakage can occur during model updates, compromising user privacy, especially for sensitive data [17]. Malicious actors in the FL network may inject poisoned updates to manipulate the global model or steal sensitive information [40]. Additionally, a compromised central server can expose model updates, revealing information about local data [41]. While device-specific local or global privacy has been well studied, finer privacy requirements at the sample level are a promising, ongoing research topic [38].

Resource management in federated learning (FL) encompasses several critical challenges, including computational resource allocation, energy consumption, memory and storage constraints, bandwidth and communication overheads, and scalability issues. Efficiently allocating computational tasks and managing energy consumption are vital due to the varying capabilities of participating devices, many of which are resource-constrained. Memory and storage limitations can hinder device participation, while bandwidth and communication overheads can lead to network congestion and increased latency. Scalability and load balancing are essential to ensure that the system can handle a growing number of devices without performance degradation. Additionally, managing network variability, task scheduling, and resource heterogeneity is crucial for maintaining consistent performance and compatibility across diverse devices. Fault tolerance and robustness are also necessary to handle device failures and interruptions effectively. Resource-aware aggregation methods further help balance contributions from devices with different resource capabilities, optimizing the overall training process in FL environments [42,43].

Data issues, such as data quality, distribution, heterogeneity, and privacy, pose significant challenges in FL. The quality of local data greatly affects the performance of the global model. Additionally, the independent and identically distributed (non-IID) nature of local data in FL can lead to statistical heterogeneity, which degrades the performance of FL algorithms [44,45].

Finally, *model training and performance* in a federated setting are more complex than in a centralized setting due to the distributed nature of the data and the need to ensure data privacy. This complexity can lead to difficulties in model convergence and affect the performance of the trained model. Moreover, evaluating model performance in FL is challenging due to the inability to access raw data directly [5,45]. Model training and performance in FL face challenges such as convergence issues, scalability, communication overhead, resource constraints, and handling non-IID data. Additionally, synchronization and asynchronous training, model accuracy and generalization, hyperparameter tuning, security and robustness, fault tolerance, and fairness and bias are significant concerns.

Figure 3 summarizes the six core challenges in federated learning and their main problems. Which of these problems and how the bioinspired algorithms are being used to tackle them are the main focuses of this paper and will be discussed later in the paper.

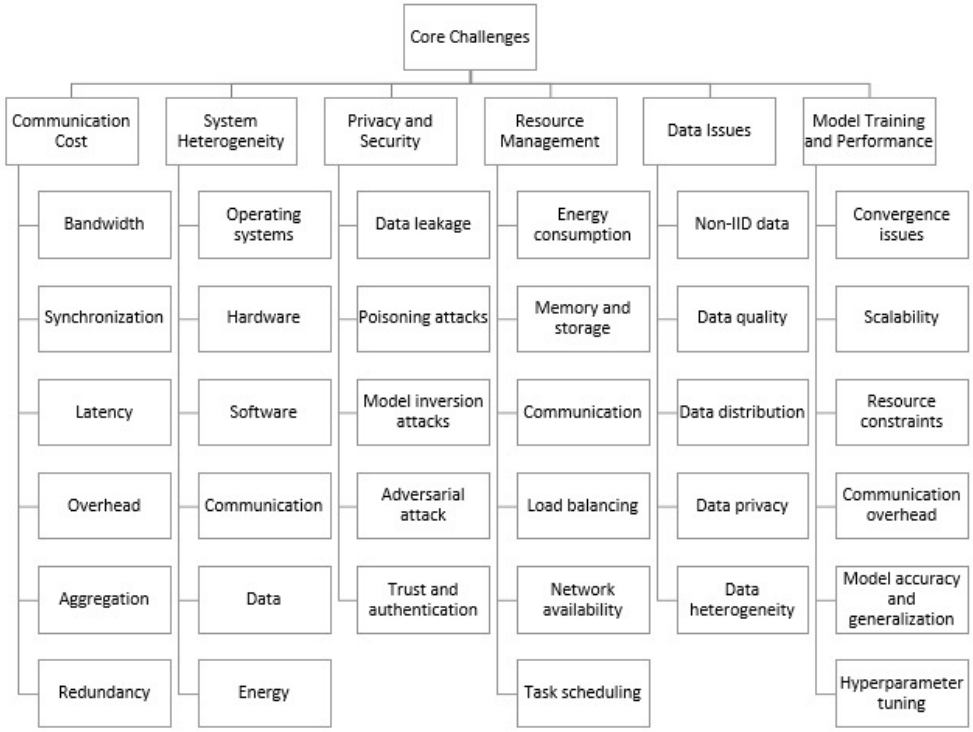


Figure 3. FL core challenges.

3. Bioinspired Computing

Bioinspired computing is an interdisciplinary field of research that draws inspiration from natural systems and processes to design novel computational algorithms for solving complex problems [7,10]. The idea is to leverage the remarkable problem-solving capabilities observed in nature, leading to the development of robust and efficient techniques for tackling complex challenges. This field has seen tremendous growth over the past few years due to the increased problem complexity, the rise of big data, the advancements of computational power and connectivity, improved algorithm analysis and design, interdisciplinary collaboration, and broader applicability. Bioinspired computing encompasses many subfields, the main ones being *evolutionary computing* (EC) and *swarm intelligence* (SI), as summarized in Figure 4, and will be briefly reviewed here.

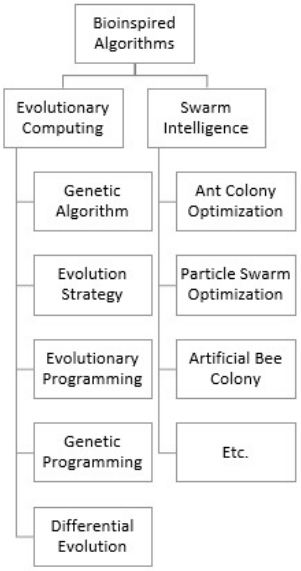


Figure 4. Main bioinspired algorithms covered in this review.

3.1. Evolutionary Computing

Evolutionary computing, also known as *evolutionary computation* or *evolutionary algorithms*, is a field of research that uses concepts from evolutionary biology to develop search and optimization techniques for solving complex problems. The foundational principles of evolutionary algorithms (EAs) are rooted in Darwinian evolution, which proposes that a population of individuals, subjected to genetic variation and natural selection, evolves over generations, resulting in individuals increasingly well-adapted to their environment. This simple, yet profound theory has given rise to the powerful class of computational methods known as evolutionary algorithms [46–50].

EAs are iterative, population-based methods that mimic the process of natural selection. They involve the creation, mutation, recombination, and selection of candidate solutions to optimize a given problem. The most well-known types of EAs include the following:

- *Genetic Algorithms* (GAs): GAs are inspired by the process of natural selection and, in their standard, form operate on a fixed-size population of potential solutions encoded as binary chromosomes. These solutions undergo genetic operations such as selection, crossover, and mutation to evolve over generations, optimizing the target problem.
- *Evolution Strategies* (ESs): ESs primarily focus on the optimization of real-valued parameters instead of binary chromosomes. They employ mechanisms like mutation and recombination, but emphasize the self-adaptation of the strategy parameters to dynamically adjust the search process.
- *Evolutionary Programming* (EP): EP emphasizes the evolution of finite-state machines or similar structures, often applied to optimization and prediction problems. The primary operation in EP is mutation, with a focus on adapting behavioral models.
- *Genetic Programming* (GP): GP extends the principles of GAs to the evolution of computer programs. Solutions are represented as tree structures, and genetic operations manipulate these trees to evolve programs that perform a specific task or solve a given problem.

A standard evolutionary algorithm follows a generic, iterative process, which can be summarized as follows Algorithm 1:

Algorithm 1. Evolutionary Algorithm (EA)

1. *Initialization*: Generate an initial population of individuals randomly representing potential solutions.
2. *Evaluation*: Assess the fitness of each individual in the population based on an objective function.
3. *Selection*: Select individuals based on their fitness to form a mating pool.
4. *Reproduction*: Generate new individuals through genetic recombination (crossover) and mutation.
5. *Variation*: Introduce genetic variation in the population through mutation and crossover.
6. *Replacement*: Form a new population by replacing part or all of the old population with the new individuals.
7. *Termination*: The process iterates until a stopping criterion is met, such as a maximum number of generations or a satisfactory fitness level.

In Algorithm 1 individuals are represented using data structures suitable for the problem domain, such as binary strings, real-valued vectors, or tree structures. The objective is defined by a fitness function, which measures the quality or adaptability of each individual. The representation and fitness function are crucial as they influence the search dynamics and the algorithm's ability to find optimal solutions.

3.2. Swarm Intelligence

Swarm intelligence is another important subarea of bioinspired computing. It is based on the collective behavior of decentralized and self-organized systems. The natural

inspirations for swarm intelligence are swarms of insects, flocks of birds, schools of fish, colonies of ants, and virtually any type of animal or insect society. The idea is that simple agents interact to produce complex, adaptive, and intelligent behaviors. This concept is particularly evident in social animals like insects, birds, and fish, where group dynamics lead to remarkable efficiencies and capabilities in survival, foraging, and navigation. The algorithms developed in this field are usually applied to solve search and optimization problems [9,11].

This field was first coined in the late 1980s by Gerardo Beni and Jing Wang in the context of cellular robotic systems [51]. Swarm intelligence systems exhibit several key properties [7]: proximity (the ability of individuals to form social links through interaction); quality (the capacity to evaluate interactions with the environment and each other); diversity (a fundamental aspect for handling unexpected situations and maintaining robustness); stability (avoiding drastic behavioral shifts in response to environmental changes); and adaptability (adjusting to environmental and population changes effectively).

These principles enable swarm systems to show emergent intelligence, where the collective behavior of the system exceeds the capabilities of any individual agent. There are a vast number of SI algorithms, and Table 1 references some of them. This list includes those works that either are commonly found in the literature, such as the ACO, PSO, and ABC algorithms, or were used in one or more of the works reviewed in this survey.

Table 1. Main swarm intelligence algorithms used with FL.

Algorithm	Description	Citation
Ant Colony Optimization (ACO)	Inspired by foraging behavior of ants.	[52]
Particle Swarm Optimization (PSO)	Mimics movement of bird flocks.	[53]
Artificial Bee Colony (ABC) Optimization	Simulates foraging behavior of honey bees.	[54]
Firefly Algorithm (FFA)	Inspired by flashing behavior of fireflies.	[55]
Bacterial Foraging Optimization (BFO)	Based on foraging strategies of E. coli bacteria.	[8]
Black Widow Algorithm (BWA)	Inspired by hunting behavior of black widow spiders.	[56]
Red Fox Optimization (RFO)	Mimics hunting strategies of red foxes.	[57]
Nomadic People Optimizer (NPO)	Inspired by the behavior of nomadic people searching for resources.	[58]
Polar Bear Optimization Algorithm (PBA)	Inspired by the hunting strategies of polar bears.	[59]
Dragonfly Optimization Algorithm (DFOA)	Mimics hunting behavior of dragonflies.	[60]
Whale Optimization Algorithm (WOA)	Inspired by social foraging and bubble-net hunting of humpback whales.	[61]
Cuckoo Search Algorithm (CSA)	Inspired by egg-laying behavior of some cuckoo bird species.	[62]
Dingo Optimization Algorithm (DOA)	Inspired by hunting and social behavior of dingoes.	[63]
Salp Swarm Algorithm (SSA)	Mimics swarming behavior of salps in the ocean.	[64]

Among all of these, one deserves particular attention in this review due to its common use in conjunction with FL environments, the Particle Swarm Optimization (PSO) algorithm, which appeared in 85% of the reviewed works.

Particle Swarm Optimization, introduced by James Kennedy and Russell Eberhart in 1995 [53], is a computational method for optimizing a wide range of functions. Inspired by the social behavior of birds flocking or fish schooling, PSO operates by having a swarm of particles (representing candidate solutions) move through the solution space to find optimal solutions.

In PSO, each particle represents a candidate solution to the optimization problem. The particles move through the solution space, influenced by their own best-known position and the best-known positions of their neighbors. The movement of particles is governed by the following equations [7]:

$$\mathbf{v_i}(t+1) = \mathbf{v_i}(t) + \phi_1 \cdot \mathbf{r_1} \cdot (\mathbf{p_i} - \mathbf{x_i}(t)) + \phi_2 \cdot \mathbf{r_2} \cdot (\mathbf{p_g} - \mathbf{x_i}(t)) \quad (1)$$

$$\mathbf{x_i}(t+1) = \mathbf{x_i}(t) + \mathbf{v_i}(t+1) \quad (2)$$

where $\mathbf{v_i}(t)$ is the velocity of particle i at time t; $\mathbf{x_i}(t)$ is the position of particle i at time t; $\mathbf{p_i}$ is the best position found by particle i (personal best); $\mathbf{p_g}$ is the best position found by any particle in the swarm (global best); ϕ_1 and ϕ_2 are acceleration coefficients; and $\mathbf{r_1}$ and $\mathbf{r_2}$ are random vectors with components uniformly distributed in [0, 1].

The position-update equation ensures that each particle moves towards a weighted combination of its personal best and the global best position. The velocity equation incorporates both cognitive (individual learning) and social (group learning) components, facilitating the particles' exploration and exploitation capabilities.

The Particle Swarm Optimization (PSO) algorithm can be implemented using different neighborhood topologies. In the global best (gbest) topology, each particle is influenced by the best performing particle in the entire swarm, allowing for rapid convergence, but potentially leading to premature convergence to local optima. On the other hand, the local best (lbest) topology influences each particle by the best performing particle within its local neighborhood, promoting diversity and aiding in the avoidance of local optima, although it may require more iterations to converge.

PSO offers several advantages over other optimization algorithms. The algorithm is easy to implement and requires only a few parameters to be adjusted, which is its simplicity. PSO can be applied to a wide range of optimization problems, including continuous, discrete, and multi-objective optimization, demonstrating its flexibility. PSO often converges faster than other evolutionary algorithms, such as Genetic Algorithms (GAs), particularly for continuous optimization problems, indicating its efficiency.

PSO has been successfully applied to various real-world problems. These include Engineering design, where it is used for optimizing the design parameters of mechanical and electrical systems [65,66], neural network training, where it is used for adjusting the weights and biases of neural networks for better performance [67], financial modeling, where it is used for forecasting and optimizing investment portfolios [68], and image and signal processing, where it is used for enhancing image quality and reducing noise in signals [69].

A typical PSO algorithm can be implemented as follows Algorithm 2:

Algorithm 2. Particle Swarm Optimization (PSO)

1. *Initialization*: Generate initial particles' positions and velocities randomly.
2. *Evaluation*: Assess the fitness of each particle based on an objective function.
3. *Replacement*: Update particles' personal bests and the global bests.
4. *Termination*: The process iterates from step 2 until a stopping criterion is met, such as a maximum number of generations or a satisfactory fitness level.

Algorithm 2 outlines the basic steps involved in the PSO algorithm, emphasizing the iterative nature of the process where particles continually update their positions and velocities based on their own experiences and the experiences of their neighbors.

4. Survey Methodology

This paper provides a comprehensive systematic review of the state of the art in the integration of federated learning (FL) and bioinspired algorithms. By exploring the synergies between these two fields, this review aims to elucidate the benefits, challenges, and research gaps in their combined use. The study follows the Preferred Reporting Items for Systematic Reviews and Meta-Analysis (PRISMA) guidelines from 2020 [12], encompassing critical processes such as formulating research questions and search terms (search strings), selecting relevant research sources and management software, and establishing inclusion and exclusion criteria. These individual processes will be elaborated in detail in the following sections.

4.1. Research Questions and Search Protocol

This systematic review addresses the following questions:

- Question 1: Which bioinspired algorithm is being used with FL, and what is the research context?
- Question 2: What type of architecture, data partitioning, privacy mechanism, aggregation method, and system heterogeneity are observed in the FL+BIC approaches?
- Question 3: What FL core challenge is being addressed by the bioinspired computing algorithm?

To address these questions, this study uses the four main computer science databases in the article search (ACM Digital Library, SCOPUS, IEEE Xplore, and Web of Science) to search for studies of any period that match with the strings defined by the PICOC method [70,71], as presented in Table 2. PICOC is an acronym for Population, Intervention, Comparison, Outcome, and Context, a framework used in computer science to find and describe the research questions of a systematic literature review [71], and each of these items is defined in Table 2.

Table 2. PICOC definitions.

Index	Description	Value	Terms
Population	Focus area or domain	Any	Any
Intervention	The main method or technique	Federated learning (FL)	Federated learning
Comparison	The method or technique that is compared with intervention	Bioinspired Computing	evolutionary computing, swarm intelligence
Outcome	The results expected by the intervention and comparison application	Machine learning (ML) tasks	Classification, prediction, Clustering, Regression, etc.
Context	The context of the applications	Any	Any

With intervention and comparison, it was possible to define the search string ("federated learning" AND ("evolutionary computing" OR "swarm intelligence") that was applied to the advanced search of each database engines defined previously. The outcomes were used as filters in the selection process to find applications of ML tasks, and population is used in the analysis section to summarize the findings.

4.2. Selection Process

Applying the search string in the scientific databases resulted in 165 papers, filtered in two stages, first removing the repeated and the conference reviews, then applying the criteria using titles, keywords, abstracts, and conclusions, as defined in Table 3.

Table 3. Selection criteria.

Criteria Type	Criteria
Language	Inclusion: Studies in English
Source	Inclusion: Conferences, or journal articles, or book chapters
Accessibility	Exclusion: Pre-Print, not peer-reviewed, conferences abstracts Exclusion: Not accessible or full-text not available
Relevance	Inclusion: Applications of FL and bioinspired computing in ML tasks Inclusion: About FL and bioinspired computing algorithms Exclusion: About Deep Learning (DL)
Frequency	Exclusion: Repeated.

Considering that federated learning (FL) was originally created by Google [13] as a solution for Deep Learning (DL) issues, the inclusion of DL would broaden the search focus, making it less specific and targeted. Therefore, papers involving Deep Learning without the use of other bioinspired algorithms will be excluded.

After applying the selection criteria of Table 3, 48 conference articles (proceedings) and 18 repeated articles were removed at the first stage of filtering. At the second stage, 5 were not accessible and 74 were not about federated learning and bioinspired algorithms,

not applied to an ML task, or they were exclusively about DL without using any bioinspired approach. This resulted in 20 papers, as can be seen in Figure 5.

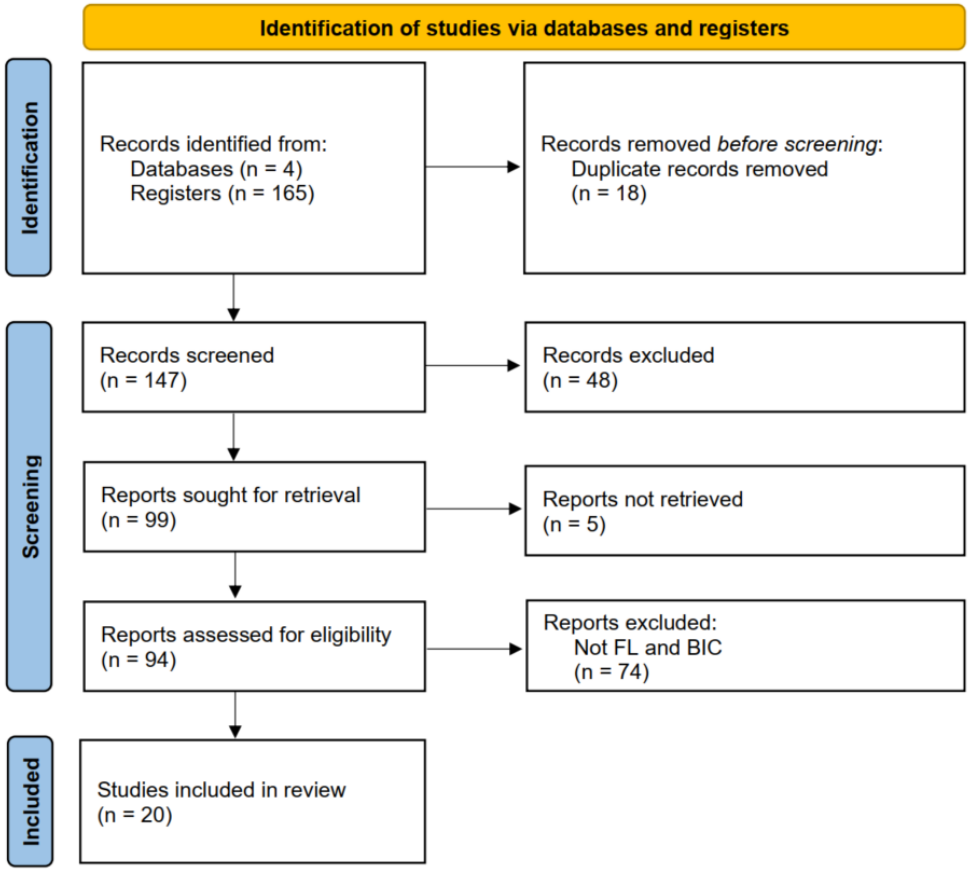


Figure 5. PRISMA flow diagram. Generated with [72].

5. Survey Results

This section presents the results of the systematic survey on the hybridization of federated learning (FL) and bioinspired algorithms. The survey is divided into three main subsections: FL characteristics and core challenges. The Section 5.1 examines various FL characteristics, including the architecture, data partitioning methods, privacy mechanisms, aggregation methods, and system heterogeneity. The Sections 5.2 and 5.3 addresses the core challenges tackled by bioinspired algorithms in FL, focusing on communication cost, system heterogeneity, privacy and security, resource management, data issues, and model training and performance.

By statistically analyzing the distribution of papers by country, it is possible to see that China has the largest number of institutions (8) publishing on federated learning and bioinspired algorithms, followed by India (6). In third place there are four countries: Lebanon (4); Egypt (4); Saudi Arabia (4); and the United States (4). Then, there is Japan (3), France (2), Portugal (2), South Korea (2), Sweden (2), Bangladesh (1), Canada (1), Iraq (1), Ireland (1), Malaysia (1), Pakistan (1), Poland (1), Qatar (1), and Spain (1), as can be observed in Figure 6.

The analysis of the distribution of papers by year of publication shows a positive trend line, with a projected growth in interest in research on the topic, as can be seen in Figure 7.

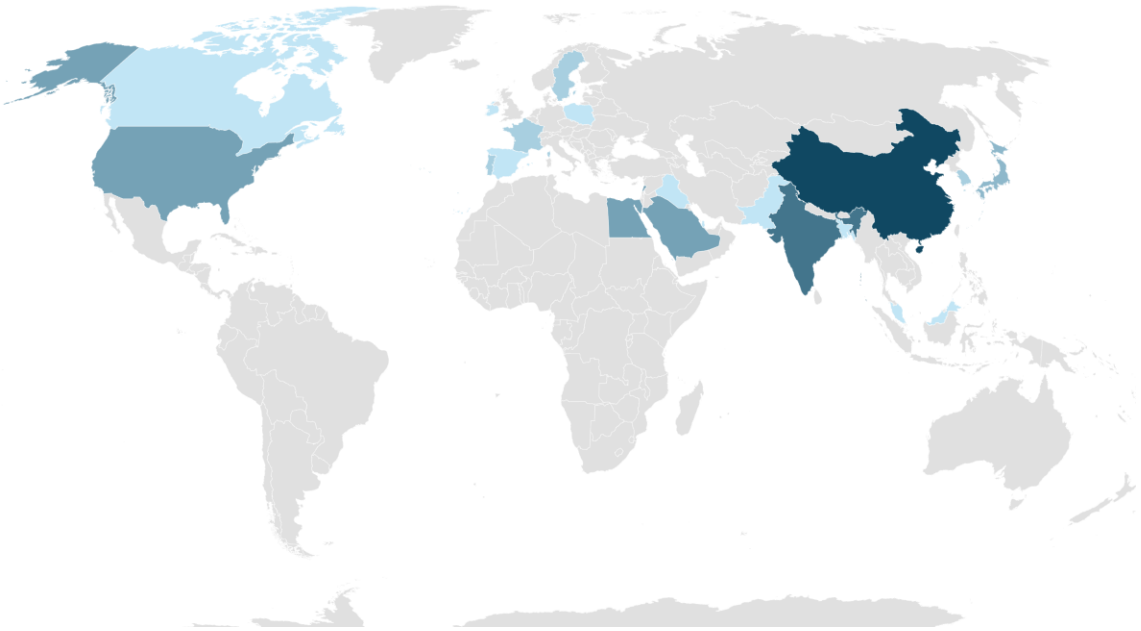


Figure 6. Distribution of institutes investigating FL and BIC by country (the colors represent the concentration of institutes by country) .

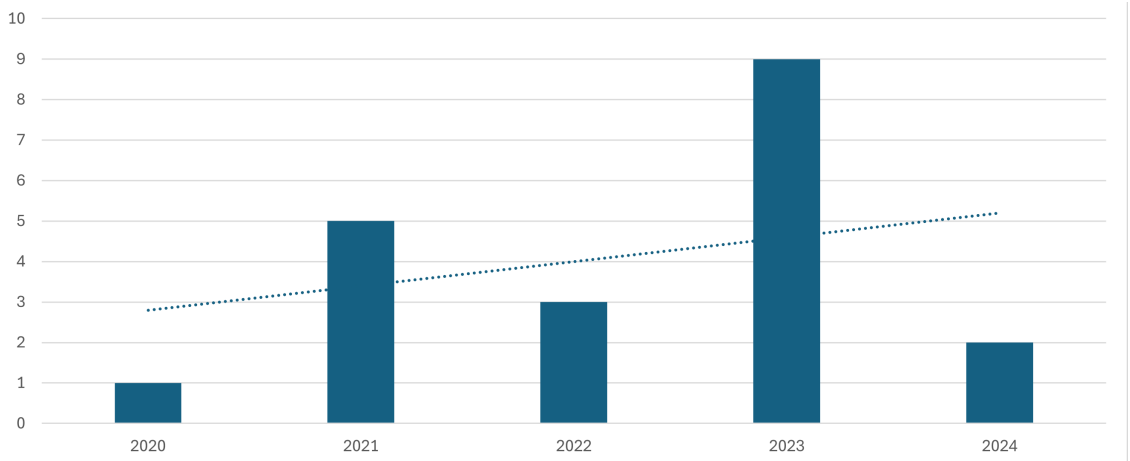


Figure 7. Year of publication and trend line.

5.1. Q1: Overview of Bioinspired Algorithms in Federated Learning

Bioinspired algorithms have been increasingly integrated into federated learning (FL) to address various challenges and enhance the overall performance of FL systems. Table 4 provides an overview of how these algorithms are utilized across the 20 selected papers.

As can be observed, 85% of the works use PSO as the main optimization algorithm. This is mainly due to its simplicity, ease of implementation, and proven effectiveness in various optimization tasks. Most importantly, PSO is particularly well suited for federated learning environments as it efficiently handles the optimization of model parameters, convergence speed, and resource management. Its ability to adaptively search for optimal solutions in a distributed manner makes it ideal for the decentralized and collaborative nature of federated learning systems.

In terms of focus, the papers cover a diverse range of applications and improvements within FL. The primary areas of application include optimizing energy consumption in smart buildings and cities, enhancing IoT security for industry and healthcare, preventing fraud, and predicting diseases like brain strokes and schizophrenia.

Table 4. Overview of bioinspired algorithms and the papers' scope. SHO: Spotted Hyenna Optimization. RFO: Red Fox Optimization. NPO: Nomadic People Optimizer. PBA: Polar Bear Algorithm. DFOA: Dragonfly Optimization Algorithm. WOA: Whale Optimization Algorithm. CSA: Cuckoo Search Algorithm. FFA: Firefly Algorithm. Jaya: Jaya Algorithm. DOA: Dingo Optimization Algorithm. SSA: Salp Swarm Algorithm.

Paper Citation	Bioinspired Algorithm(s)	Focus of the Work
[73]	PSO	Optimize aggregation to improve energy consumption in smart buildings
[74]	GA, PSO	Detect insurance fraud by optimizing model parameters
[75]	PSO	Efficient energy harvesting in IIoT environments
[76]	PSO	Enhance accuracy and convergence speed in healthcare IoT devices
[77]	PSO	Maintain communication efficiency and Byzantine fault tolerance
[78]	PSO	Handle malicious and heterogeneous data in hierarchical FL
[79]	PSO	Optimize energy consumption in smart city energy communities
[80]	PSO	Reduce communication costs in federated learning
[81]	PSO	Predict brain strokes by optimizing model performance
[82]	PSO	Hyperparameter tuning to enhance model convergence
[83]	SHO,PSO	Optimize a trust evaluation scheme
[84]	BWA, RFO, NPO, PBA, DFOA, WOA, CSA, PSO	Manage image processing tasks and optimize model performance
[85]	PSO	Optimize models for Industrial IoT and smart city services
[86]	PSO	Secure IoT devices and enhance data privacy
[87]	PSO	Secure IoT devices, focusing on privacy and security
[88]	PSO, GA	Predict energy consumption in smart buildings
[89]	PSO	Reduce communication costs in federated contrastive learning
[90]	FFA, Jaya	Enhance schizophrenia diagnosis through optimized models
[91]	PSO	Securely distributed computation with divided data
[92]	DOA, SSA	Task scheduling in edge–cloud computing environments

5.2. Q2: Federated Learning Characteristics

FL architectures can be classified into four main types: centralized, decentralized, hierarchical, and cross-silo. Most of the reviewed papers [73–76,80–88,90] utilize a centralized architecture, where a central server coordinates the training process by aggregating updates from clients. This architecture is favored for its simplicity and effectiveness in various applications such as energy consumption optimization in smart buildings and healthcare IoT devices. Papers such as [77] propose a decentralized architecture, where there is no central server and clients communicate directly with each other (peer-to-peer) to aggregate model updates. This approach enhances robustness against single points of failure and is suitable for scenarios requiring high resilience. The papers [78,92] adopt a hierarchical architecture, which involves multiple layers of aggregation. This structure can efficiently handle large-scale federated systems by reducing the communication burden on the central server. The papers [79,89,91] employ a cross-silo architecture, commonly used among organizations or institutions where data are partitioned across silos, ensuring that each organization contributes to the global model without exposing its data.

Data partitioning methods in FL include horizontal, vertical, and transfer learning. The majority of the reviewed papers [73–84,86–88,90–92] use horizontal partitioning, where each client's dataset contains a subset of the overall data samples. This method is widely applicable in scenarios where data samples are distributed across different devices or locations. The paper [89] utilizes vertical partitioning, where different clients hold different features of the same data records. This method is suitable for applications requiring feature-based collaboration among clients. The paper [85] incorporates transfer learning, where knowledge from one task is transferred to improve the learning process of another task. This method is particularly useful in adapting models to new tasks or domains.

Ensuring data privacy in FL is crucial, and various mechanisms are employed to achieve this. A common privacy mechanism used in many papers [73–86,88–90,92] is federated averaging (FedAvg), which involves averaging the locally trained model updates

before sending them to the central server. This method helps protect the raw data from being exposed. The paper [87] employs differential privacy (DP), which adds noise to the data or model updates to ensure that individual data points cannot be inferred from the aggregated results. This enhances privacy while maintaining model performance. The paper [91] uses secure multiparty computation (SMC) to perform distributed computations on encrypted data, ensuring that the data remain confidential even during processing. This method is highly effective in scenarios requiring strong privacy guarantees.

Aggregation methods in FL determine how model updates from clients are combined to form the global model. The papers [79,82,83,85–87,90–92] use federated averaging (FedAvg), which involves the simple averaging of model updates. This method is straightforward and effective in various applications. The papers [73–78,80,81,84,88,89] propose optimization-based aggregation (OBA), in the sense that they involve optimizing the aggregation process to improve model accuracy and convergence.

System heterogeneity in FL refers to the variations in device capabilities, communication, software environments, and other factors affecting model training. The papers [73,75–78,80,81,83–86,88] address device heterogeneity (DH), focusing on optimizing FL algorithms to handle variations in device performance and capabilities. A significant number of papers [73–87,89–92] tackle communication heterogeneity (CH) by proposing methods to reduce communication costs and handle varying network conditions. The papers [74,83,85] discuss software heterogeneity (SH), ensuring that FL algorithms are compatible with different software environments and platforms. While not explicitly mentioned in the reviewed papers, slow convergence (SC) is an inherent challenge in FL, which can be mitigated through techniques like optimization-based aggregation and advanced hyperparameter tuning.

Table 5 summarizes the main characteristics of the FL architectures, data partitioning methods, privacy mechanisms, aggregation methods, and types of system heterogeneity discussed in the reviewed papers.

Table 5. Federated learning characteristics. FedAvg: federated averaging. DP: differential privacy. OBA: optimization-based aggregation. SMC: secure multiparty computation. DH: device heterogeneity. CH: communication heterogeneity. SH: software heterogeneity.

Paper	Architecture	Data Partitioning	Privacy Mechanism	Aggregation Method	System Heterogeneity
[73]	Centralized	Horizontal	FedAvg	OBA	DH, CH
[74]	Centralized	Horizontal	FedAvg	OBA	CH, SH
[75]	Centralized	Horizontal	FedAvg	OBA	DH, CH
[76]	Centralized	Horizontal	FedAvg	OBA	DH, CH
[77]	Decentralized	Horizontal	FedAvg	OBA	DH, CH
[78]	Hierarchical	Horizontal	FedAvg	OBA	DH, CH
[79]	Cross-silo	Horizontal	FedAvg	FedAvg	CH
[80]	Centralized	Horizontal	FedAvg	OBA	DH, CH
[81]	Centralized	Horizontal	FedAvg	OBA	DH, CH
[82]	Centralized	Horizontal	FedAvg	FedAvg	CH
[83]	Centralized	Horizontal	FedAvg	FedAvg	CH
[84]	Centralized	Horizontal	FedAvg	FedAvg	DH, CH
[85]	Centralized	Transfer	FedAvg	FedAvg	DH, CH, SH
[86]	Centralized	Horizontal	FedAvg	FedAvg	DH, CH
[87]	Centralized	Horizontal	DP	FedAvg	CH
[88]	Centralized	Horizontal	FedAvg	OBA	DH
[89]	Cross-silo	Vertical	FedAvg	OBA	CH
[90]	Centralized	Horizontal	FedAvg	FedAvg	CH
[91]	Cross-silo	Horizontal	SMC	FedAvg	CH
[92]	Hierarchical	Horizontal	FedAvg	FedAvg	CH

5.3. Q3: Core Challenges

The core challenges identified include communication cost, system heterogeneity, privacy and security, resource management, data issues, and model training and performance. Table 6 provides a summary of these challenges and how each paper addresses them.

Communication cost is a significant concern in FL due to the frequent exchange of model updates between clients and the central server. The papers [76,77,80,89] address this challenge by implementing Particle Swarm Optimization (PSO) to optimize the communication process, thereby reducing the number of updates required and conserving bandwidth.

For instance, [77] employs a communication-efficient distributed swarm learning approach that minimizes the overhead associated with data transmission.

All papers address some kind of system heterogeneity, which includes device heterogeneity (DH), communication heterogeneity (CH), and software heterogeneity (SH), which are other critical challenges, as represented in Table 5.

Privacy and security are paramount in FL, given the decentralized nature of data storage and processing. The paper [77] presents a solution against Byzantine attacks; the paper [83] creates a method for data trust assessment; the papers [87,91] explore the use of differential privacy (DP) and secure multiparty computation (SMC) to enhance data confidentiality. The bioinspired algorithms in these papers, such as PSO, help in tuning the parameters to balance privacy and model accuracy. For instance, [87] integrates DP with PSO to protect individual data points while optimizing the model's performance.

Resource management is crucial for the efficient functioning of FL, especially in scenarios involving limited computational resources. The papers [73,75,79,88,92] propose optimization techniques to manage resources effectively. The hybrid aggregation approach in [73], for example, improves energy consumption in smart buildings by optimizing the federated learning process using the PSO and Levenberg–Marquardt algorithms.

Data issues such as non-independent and identically distributed (non-IID) data and data imbalance are addressed in the papers [77,78]. These papers propose bioinspired algorithms that adjust the learning process to account for data diversity and ensure fair model training across all clients. The evolutionary multi-model federated learning approach in [78] is particularly effective in handling malicious and heterogeneous data, ensuring robust performance.

Model training and performance are enhanced through bioinspired algorithms that optimize the hyperparameters and features, improving convergence rates. The papers [74,76,81,82,85,86,90] demonstrate the effectiveness of such algorithms. For example, Ref. [76] uses an improved PSO to enhance the training of healthcare IoT devices, achieving better model accuracy and faster convergence compared to traditional methods. Similarly, Ref. [82] utilizes the PSO algorithm for hyperparameter tuning to improve model training efficiency.

In summary, bioinspired algorithms play an important role in addressing key challenges in federated learning. They optimize communication costs, manage system heterogeneity, enhance privacy and security, improve resource management, tackle data issues, and boost model training and performance. Table 6 summarizes the core challenges tackled by each paper, providing a comprehensive overview of their contributions to FL.

Table 6. Core challenges tackled by the bioinspired algorithms per paper (marked as check). CC: communication cost. PS: privacy and security. RM: resource management. DI: data issues. MTP: model training and performance.

Paper	CC	PS	RM	DI	MTP
[73]			✓		
[74]					✓
[75]			✓		
[76]	✓				✓
[77]	✓	✓		✓	
[78]				✓	
[79]			✓		
[80]	✓				
[81]					✓
[82]					✓
[83]		✓			
[84]		✓			
[85]					✓
[86]					✓
[87]		✓			
[88]			✓		
[89]	✓				
[90]					✓
[91]		✓			
[92]			✓		

6. Open Challenges and Future Directions

Despite the significant advancements in hybridizing FL with bioinspired algorithms, several challenges remain. Ensuring the scalability of bioinspired algorithms in large-scale FL systems is a critical issue. As the number of clients increases, the computational and communication overhead can become prohibitive [4]. Addressing the diverse nature of devices and data in FL environments is also a complex task. Variations in hardware capabilities, network conditions, and data distributions require adaptive and robust bioinspired algorithms that can manage this heterogeneity.

Although bioinspired algorithms can enhance privacy and security, ensuring robust protection against sophisticated attacks remains a challenge. Techniques like differential privacy and secure multiparty computation need further refinement to maintain high model performance without compromising data security [93]. Many devices participating in FL have limited computational power and battery life. Developing bioinspired algorithms that are resource-efficient and can operate under these constraints is crucial for practical applications [14].

Understanding how machine learning models reach their conclusions is also increasingly important, especially in sensitive areas like healthcare. While traditional models could be highly accurate, their lack of transparency hinders trust and prevents experts from fully understanding their decisions. Bioinspired methods, such as evolutionary algorithms, offer a potential solution by creating models that are easier to interpret, like the grammatical evolution that could generate human-readable rules. Federated learning combines these interpretative models from multiple sources without sharing sensitive patient data. This approach protects privacy while producing transparent models that can be understood by both doctors and patients, which is crucial for applications, for example like diabetes management [94].

Achieving fast convergence in FL while maintaining accuracy is challenging. Bioinspired algorithms need to be optimized to reduce the time required for models to converge, especially in dynamic environments with frequently changing data [13]. The complexity of bioinspired algorithms can hinder their practical implementation in FL. Simplifying these algorithms without sacrificing their efficacy is an ongoing research challenge [95]. To address these challenges, future research in hybridizing FL with bioinspired algorithms could focus on the following areas:

- Scalability in large-scale FL systems: Increasing the scalability of bioinspired algorithms in large-scale FL systems presents a considerable challenge. As the number of clients increases, the computational and communication overhead may become impractical. Future research could focus on the following:
  - Investigating hierarchical and decentralized FL architectures to distribute processing and reduce load on the central server.
  - Developing resource-efficient bioinspired algorithms suitable for resource-limited devices.
  - Designing efficient communication mechanisms to minimize the amount of data exchanged between clients and the server.
- Managing device and data heterogeneity: The diverse nature of devices and data in FL environments demands adaptable and robust bioinspired algorithms. Future research could focus on the following:
  - Designing bioinspired algorithms capable of handling different hardware capabilities, network conditions, and data distributions.
  - Developing mechanisms for detecting and adapting to heterogeneity in real time, optimizing the learning process.
  - Exploring transfer learning techniques to enable collaboration between customers with heterogeneous datasets.

- Enhancing privacy and security: While bioinspired algorithms can increase privacy and security, ensuring robust protection against sophisticated attacks remains a challenge. Future research could focus on the following:
  - Refining privacy-preserving techniques such as differential privacy, homomorphic encryption, and secure multiparty computation. Integrating these techniques with bioinspired algorithms can provide robust security while maintaining model performance.
  - Investigating new methods for detecting and defending against attacks targeting FL systems by leveraging swarm intelligence capabilities.
  - Exploring the use of blockchain technologies to increase security and data integrity in FL systems.
- Efficient resource management: Many devices participating in FL have limited computing power and battery life. The development of resource-efficient bioinspired algorithms capable of operating under these constraints is crucial for practical applications. Future research could focus on the following:
  - Designing lightweight bioinspired algorithms that minimize energy consumption and computational resource utilization.
  - Optimizing computing and communication processes to extend battery life and reduce latency, especially on mobile and IoT devices.
  - Exploring edge computing techniques to offload computationally intensive tasks to the client side, reducing the load on resource-constrained devices.
- Accelerating model convergence: Achieving fast convergence in FL while maintaining accuracy is challenging. Bioinspired algorithms need to be optimized to reduce the time it takes for models to converge, especially in dynamic environments with constantly changing data. Future research could focus on the following:
  - Investigating new bioinspired or hybrid algorithms that demonstrate faster convergence rates in FL scenarios.
  - Optimizing algorithm and search engine parameters to accelerate convergence without compromising model accuracy.
  - Exploring the use of information from previous customers or global models to bootstrap the learning process and accelerate convergence.
- Simplifying the implementation of bioinspired algorithms: The complexity of bioinspired algorithms can make their practical implementation in FL difficult. Simplifying these algorithms without sacrificing their effectiveness is an ongoing research challenge. Future research could focus on the following:
  - Developing user-friendly software frameworks and libraries that simplify the integration of bioinspired algorithms into FL systems.
  - Creating guidelines and best practices for the design and implementation of bioinspired algorithms in FL environments, considering specific constraints.
  - Exploring the explainability of machine learning models.
  - Investigating automated machine learning techniques to optimize and tune the parameters of bioinspired algorithms, reducing the need for manual tuning.

Addressing these challenges and exploring the future directions outlined in this section will pave the way for more efficient, scalable, secure, and practical FL systems. Combining different bioinspired algorithms, such as PSO with Genetic Algorithms or Ant Colony Optimization, can leverage the strengths of each method. Hybrid approaches can enhance convergence speed, accuracy, and robustness. Conducting real-world experiments and deploying FL systems with bioinspired algorithms in various applications (e.g., healthcare, smart cities, the IoT) will provide valuable insights.

7. Conclusions

The integration of bioinspired algorithms with federated learning presents a promising avenue for addressing several critical challenges in decentralized machine learning. This systematic review highlighted the key areas where these algorithms have shown significant potential, including communication cost reduction, system heterogeneity management, privacy and security enhancement, resource management, and improvements in data handling and model performance.

Bioinspired algorithms such as Particle Swarm Optimization and Genetic Algorithms have been effectively employed to optimize various aspects of FL. For instance, PSO has been widely used to minimize communication overhead by optimizing the frequency and size of model updates, as seen in [76,80], for example. This optimization is crucial in scenarios where bandwidth is limited or where communication costs are high. Similarly, GAs have been utilized to optimize feature subset selection, as proposed in [74], and to perform local model optimization on the client side [82,88].

Privacy and security remain paramount in FL. Integrating bioinspired algorithms with techniques like differential privacy and secure multiparty computation helps in tuning parameters to achieve a balance between data protection and model accuracy. The papers [77,91] illustrated the effectiveness of these integrations in enhancing privacy while maintaining robust model performance.

Resource management is another critical area where bioinspired algorithms have made significant contributions. By optimizing the computational and communication processes, these algorithms extend battery life and reduce latency in resource-constrained devices. The hybrid aggregation approaches discussed in [73,75] are examples of how bioinspired algorithms can improve energy consumption in smart environments.

Handling non-IID data and ensuring fair model training across diverse clients is a persistent challenge in FL. Bioinspired algorithms such as evolutionary multi-model learning, as seen in [78], effectively address these data issues by adjusting the learning process to account for data diversity and distribution.

Finally, enhancing model training and performance through hyperparameter optimization and improved convergence rates is another area where bioinspired algorithms have shown promising results. Techniques such as improved PSO have been shown to significantly boost model accuracy and reduce training times, as presented in [74,81].

In conclusion, the hybridization of FL with bioinspired algorithms offers a robust framework for tackling some of the most pressing challenges in decentralized machine learning. By leveraging the strengths of these algorithms, researchers can develop more efficient, secure, and scalable FL systems. The continued exploration and refinement of these hybrid approaches will be crucial in advancing the field and broadening the applicability of FL across various domains.

Author Contributions: Conceptualization, L.N.d.C. and R.M.M.d.S.; methodology, L.N.d.C. and R.M.M.d.S.; validation, L.N.d.C. and R.M.M.d.S.; formal analysis, L.N.d.C. and R.M.M.d.S.; investigation, L.N.d.C., R.M.M.d.S. and A.H.; resources, M.B.; data curation, L.N.d.C. and R.M.M.d.S.; writing—original draft preparation, L.N.d.C. and R.M.M.d.S.; writing—review and editing, L.N.d.C. and R.M.M.d.S.; visualization, L.N.d.C. and R.M.M.d.S.; supervision, L.N.d.C.; project administration, M.B.; funding acquisition, M.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by FAPESP (grant number 2021/11.905-0 and process number 2023/13355-3).

Data Availability Statement: The original contributions presented in the study are included in the article; further inquiries can be directed to the corresponding author.

Conflicts of Interest: The author Marcio Byczic is employed by the In.Lab-InovaHC Research Institute. The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ABC	Artificial Bee Colony Optimization
ACO	Ant Colony Optimization
BFO	Bacterial Foraging Optimization
BIC	bioinspired computing
BWA	Black Widow Algorithm
CC	communication cost
CFL	centralized federated learning
CH	communication heterogeneity
CSA	Cuckoo Search Algorithm
CSFL	cross-silo federated learning
DE	Differential Evolution
DFL	decentralized federated learning
DH	device heterogeneity
DI	data issues
DL	Deep Learning
DFOA	Dragonfly Optimization Algorithm
DOA	Dingo Optimization Algorithm
DP	differential privacy
EA	evolutionary algorithm
EP	Evolutionary Programming
ESs	Evolution Strategies
FedAVG	federated averaging
FedProx	Federated Extension
FedSGD	federated stochastic gradient descent
FFA	Firefly Algorithm
FL	federated learning
FTL	federated transfer learning
GA	Genetic Algorithm
GP	Genetic Programming
HE	homomorphic encryption
HFL (architecture)	hierarchical federated learning
HFL (data partitioning)	horizontal federated learning
IID	independent and identically distributed
ML	machine learning
MTP	model training and performance
NC	Natural Computing
NPO	Nomadic People Optimizer
OBA	Optimization-based aggregation
PBA	Polar Bear Optimization Algorithm
PS	privacy and security
PSO	Particle Swarm Optimization
RFO	Red Fox Optimization
RM	resource management
SA	secure aggregation
SH	software heterogeneity
SMC	secure multiparty computation
SSA	Salp Swarm Algorithm
VFL	vertical federated learning
WOA	Whale Optimization Algorithm

References

1. Zhang, C.; Xie, Y.; Bai, H.; Yu, B.; Li, W.; Gao, Y. A survey on federated learning. *Knowl.-Based Syst.* **2021**, *216*, 106775. [CrossRef]
2. Aledhari, M.; Razzak, R.; Parizi, R.M.; Saeed, F. Federated learning: A survey on enabling technologies, protocols, and applications. *IEEE Access* **2020**, *8*, 140699–140725. [CrossRef]

3. AbdulRahman, S.; Tout, H.; Ould-Slimane, H.; Mourad, A.; Talhi, C.; Guizani, M. A survey on federated learning: The journey from centralized to distributed on-site learning and beyond. *IEEE Internet Things J.* **2020**, *8*, 5476–5497. [CrossRef]
4. Li, T.; Sahu, A.K.; Talwalkar, A.; Smith, V. Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Process. Mag.* **2020**, *37*, 50–60. [CrossRef]
5. Wen, J.; Zhang, Z.; Lan, Y.; Cui, Z.; Cai, J.; Zhang, W. A Survey on Federated Learning: Challenges and Applications. *Int. J. Mach. Learn. Cybern.* **2023**, *14*, 513–535. [CrossRef] [PubMed]
6. Shanmugarasa, Y.; Paik, H.Y.; Kanhere, S.S.; Zhu, L. A Systematic Review of Federated Learning from Clients' Perspective: Challenges and Solutions. *Artif. Intell. Rev.* **2023**, *56*, 1773–1827. [CrossRef]
7. de Castro, L.N. *Fundamentals of Natural Computing: Basic Concepts, Algorithms, and Applications*; CRC Press: Boca Raton, FL, USA, 2006.
8. Passino, K.M. *Biomimicry for Optimization, Control, and Automation*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2005.
9. Sharma, A.; Sharma, A.; Pandey, J.K.; Ram, M. *Swarm Intelligence: Foundation, Principles, and Engineering Applications*; CRC Press: Boca Raton, FL, USA, 2022.
10. Dasegowda, K.; Radhakrishnan, A.; Rambabu, M.; Peri, S.; Vasudevan, K.; Prabhavathi, H.; Kareem, M.A. Nature-Inspired Computing: Scope and Applications of Artificial Immune Systems Toward Analysis and Diagnosis of Complex Problems. In *Nature-Inspired Intelligent Computing Techniques in Bioinformatics*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 147–162.
11. Awad, M.; Khanna, R. Bioinspired Computing: Swarm Intelligence. In *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*; Apress: Berkeley, CA, USA, 2015; pp. 105–125. [CrossRef]
12. Page, M.J.; McKenzie, J.E.; Bossuyt, P.M.; Boutron, I.; Hoffmann, T.C.; Mulrow, C.D.; Moher, D. The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ* **2021**, *372*, n71. [CrossRef]
13. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; Arcas, B.A.y. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the Artificial Intelligence and Statistics, Lauderdale, FL, USA, 20–22 April 2017; pp. 1273–1282.
14. Li, Q.; Wen, Z.; Wu, Z.; Hu, S.; Wang, N.; Li, Y.; Liu, X.; He, B. A survey on federated learning systems: Vision, hype and reality for data privacy and protection. *IEEE Trans. Knowl. Data Eng.* **2021**, *35*, 3347–3366. [CrossRef]
15. Shome, D.; Waqar, O.; Khan, W.U. Federated learning and next generation wireless communications: A survey on bidirectional relationship. *Trans. Emerg. Telecommun. Technol.* **2022**, *33*, e4458. [CrossRef]
16. Lim, W.Y.B.; Luong, N.C.; Hoang, D.T.; Jiao, Y.; Liang, Y.C.; Yang, Q.; Niyato, D.; Miao, C. Federated Learning in Mobile Edge Networks: A Comprehensive Survey. *IEEE Commun. Surv. Tutorials* **2020**, *22*, 2031–2063. [CrossRef]
17. Bonawitz, K.; Ivanov, V.; Kreuter, B.; Marcedone, A.; McMahan, H.B.; Patel, S.; Ramage, D.; Segal, A.; Seth, K. Practical Secure Aggregation for Privacy-Preserving Machine Learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, New York, NY, USA, 30 October–3 November 2017; pp. 1175–1191. [CrossRef]
18. Konečný, J.; McMahan, H.B.; Yu, F.X.; Richtárik, P.; Suresh, A.T.; Bacon, D. Federated Learning: Strategies for Improving Communication Efficiency. *arXiv* **2016**, arXiv:1610.05492.
19. Beikmohammadi, A.; Khirirat, S.; Magnússon, S. On the Convergence of Federated Learning Algorithms Without Data Similarity. *IEEE Trans. Big Data* **2024**, 1–10, *Early Access*. [CrossRef]
20. Liu, B.; Lv, N.; Guo, Y.; Li, Y. Recent advances on federated learning: A systematic survey. *Neurocomputing* **2024**, *597*, 128019. [CrossRef]
21. Warnat-Herresthal, S.; Schultze, H.; Shastry, K.L.; Manamohan, S.; Mukherjee, S.; Garg, V.; Sarveswara, R.; Händler, K.; Pickkers, P.; Aziz, N.A.; et al. Swarm Learning for decentralized and confidential clinical machine learning. *Nature* **2021**, *594*, 265–270. [CrossRef]
22. Liu, L.; Zhang, J.; Song, S.; Letaief, K.B. Client-Edge-Cloud Hierarchical Federated Learning. In Proceedings of the ICC 2020—2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 7–11 June 2020; pp. 1–6. [CrossRef]
23. Durgapal, P.; Kataria, P.; Garg, G.; Anand, A.S. A Comprehensive Distributed Framework for Cross-silo Federated Learning using Blockchain. In Proceedings of the 2023 Fifth International Conference on Blockchain Computing and Applications (BCCA), Kuwait, Kuwait, 24–26 October 2023; pp. 538–545. [CrossRef]
24. Souza, R.M.M.; Holm, A.; Biczyk, M.; de Castro, L. Healthchain—Bidimensional Blockchain for Collaborative and Secure Deep Learning in Healthcare. In Proceedings of the 21st International Conference on Distributed Computing and Artificial Intelligence (DCAI 2024), Salamanca, Spain, 26–28 June 2024.
25. Qi, P.; Chiaro, D.; Guzzo, A.; Ianni, M.; Fortino, G.; Piccialli, F. Model aggregation techniques in federated learning: A comprehensive survey. *Future Gener. Comput. Syst.* **2023**, *150*, 272–293. [CrossRef]
26. Liu, Y.; Kang, Y.; Zou, T.; Pu, Y.; He, Y.; Ye, X.; Ouyang, Y.; Zhang, Y.Q.; Yang, Q. Vertical Federated Learning: Concepts, Advances, and Challenges. *IEEE Trans. Knowl. Data Eng.* **2024**, *36*, 3615–3634. [CrossRef]
27. Das, A.; Patterson, S. Multi-Tier Federated Learning for Vertically Partitioned Data. In Proceedings of the ICASSP 2021—2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; pp. 3100–3104. [CrossRef]
28. Guo, W.; Zhuang, F.; Zhang, X.; Tong, Y.; Dong, J. A Comprehensive Survey of Federated Transfer Learning: Challenges, Methods and Applications. *arXiv* **2024**, arXiv:2403.01387.

29. Ouadrhiri, A.E.; Abdelhadi, A. Differential Privacy for Deep and Federated Learning: A Survey. *IEEE Access* **2022**, *10*, 22359–22380. [CrossRef]
30. Zhou, I.; Tofigh, F.; Piccardi, M.; Abolhasan, M.; Franklin, D.; Lipman, J. Secure Multi-Party Computation for Machine Learning: A Survey. *IEEE Access* **2024**, *12*, 53881–53899. [CrossRef]
31. Park, J.; Lim, H. Privacy-Preserving Federated Learning Using Homomorphic Encryption. *Appl. Sci.* **2022**, *12*, 734. [CrossRef]
32. Yin, X.; Zhu, Y.; Hu, J. A Comprehensive Survey of Privacy-preserving Federated Learning: A Taxonomy, Review, and Future Directions. *ACM Comput. Surv.* **2021**, *54*, 1–36. [CrossRef]
33. Liu, Z.; Guo, J.; Yang, W.; Fan, J.; Lam, K.Y.; Zhao, J. Privacy-Preserving Aggregation in Federated Learning: A Survey. *IEEE Trans. Big Data* **2022**, 1–20. [CrossRef]
34. Lu, Z.; Pan, H.; Dai, Y.; Si, X.; Zhang, Y. Federated Learning With Non-IID Data: A Survey. *IEEE Internet Things J.* **2024**, *11*, 19188–19209. [CrossRef]
35. Yuan, H.; Ma, T. Federated accelerated stochastic gradient descent. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 5332–5344.
36. Li, T.; Sahu, A.K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; Smith, V. Federated optimization in heterogeneous networks. *Proc. Mach. Learn. Syst.* **2020**, *2*, 429–450.
37. Gao, Y.; Lu, G.; Gao, J.; Li, J. A High-Performance Federated Learning Aggregation Algorithm Based on Learning Rate Adjustment and Client Sampling. *Mathematics* **2023**, *11*, 4344. [CrossRef]
38. Kairouz, P.; McMahan, H.B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A.N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. Advances and open problems in federated learning. *Found. Trends® Mach. Learn.* **2021**, *14*, 1–210. [CrossRef]
39. Sattler, F.; Wiedemann, S.; Müller, K.R.; Samek, W. Robust and Communication-Efficient Federated Learning From Non-i.i.d. Data. *IEEE Trans. Neural Networks Learn. Syst.* **2020**, *31*, 3400–3413. [CrossRef]
40. Gong, X.; Chen, Y.; Wang, Q.; Kong, W. Backdoor Attacks and Defenses in Federated Learning: State-of-the-Art, Taxonomy, and Future Directions. *IEEE Wirel. Commun.* **2023**, *30*, 114–121. [CrossRef]
41. Wei, K.; Li, J.; Ding, M.; Ma, C.; Yang, H.H.; Farokhi, F.; Jin, S.; Quek, T.Q.S.; Vincent Poor, H. Federated Learning With Differential Privacy: Algorithms and Performance Analysis. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 3454–3469. [CrossRef]
42. Yang, Q.; Liu, Y.; Chen, T.; Tong, Y. Federated Machine Learning: Concept and Applications. *ACM Trans. Intell. Syst. Technol.* **2019**, *10*, 1–19. [CrossRef]
43. Xu, Y.; Li, L.; Cong, P.; Zhou, J. Energy-Efficient and Reliable Federated Learning in Heterogeneous Mobile-Edge Computing. In Proceedings of the 2022 IEEE Smartworld, Ubiquitous Intelligence & Computing, Scalable Computing & Communications, Digital Twin, Privacy Computing, Metaverse, Autonomous & Trusted Vehicles (SmartWorld/UIC/ScalCom/DigitalTwin/PriComp/Meta), Haikou, China, 15–18 December 2022; pp. 689–697. [CrossRef]
44. Jiang, M.; Wei, L.; Cai, G.; Wu, X. Differentially Private Federated Learning with Heterogeneous Group Privacy. In Proceedings of the 2023 IEEE International Conferences on Internet of Things (iThings) and IEEE Green Computing & Communications (GreenCom) and IEEE Cyber, Physical & Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics), Danzhou, China, 17–21 December 2023; pp. 143–150. [CrossRef]
45. Miao, Y.; Kuang, D.; Li, X.; Xu, S.; Li, H.; Raymond Choo, K.K.; Deng, R.H. Privacy-Preserving Asynchronous Federated Learning Under Non-IID Settings. *IEEE Trans. Inf. Forensics Secur.* **2024**, *19*, 5828–5841. [CrossRef]
46. Sloss, A.N.; Gustafson, S., 2019 Evolutionary Algorithms Review. In *Genetic Programming Theory and Practice XVII*; Banzhaf, W., Goodman, E., Sheneman, L., Trujillo, L., Worzel, B., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 307–344. [CrossRef]
47. Bäck, T. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*; Oxford University Press: Oxford, UK, 1996.
48. Holland, J.H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*; MIT Press: Cambridge, MA, USA 1992.
49. Koza, J.R. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*; MIT Press: Cambridge, MA, USA 1992.
50. Koza, J.R. *Genetic Programming II: Automatic Discovery of Reusable Programs*; MIT Press: Cambridge, MA, USA 1994.
51. Beni, G.; Wang, J. Swarm Intelligence in Cellular Robotic Systems. In Proceedings of the 1989 IEEE International Conference on Robotics and Automation, Scottsdale, AZ, USA, 14–19 May 1989; Volume 1, pp. 1117–1121.
52. Dorigo, M. *Optimization, Learning, and Natural Algorithms*; MIT Press: Cambridge, MA, USA 1992.
53. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95-International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
54. Karaboga, D. An idea based on honey bee swarm for numerical optimization. In *Technical Report, Technical Report—TR06*; Department of Computer Engineering, Engineering Faculty, Erciyes University: Kayseri, Türkiye 2005.
55. Yang, X.S. Firefly Algorithms for Multimodal Optimization. In *Stochastic Algorithms: Foundations and Applications*; Watanabe, O., Zeugmann, T., Eds.; Springer: Berlin/Heidelberg, Germany, 2009; pp. 169–178. [CrossRef]
56. Hayyolalam, V.; Pourhaji Kazem, A.A. Black Widow Optimization Algorithm: A novel meta-heuristic approach for solving engineering optimization problems. *Eng. Appl. Artif. Intell.* **2020**, *87*, 103249. [CrossRef]
57. ALRahhal, H.; Jamous, R. AFOX: A new adaptive nature-inspired optimization algorithm. *Artif. Intell. Rev.* **2023**, *56*, 15523–15566. [CrossRef]

58. Salih, S.Q.; Alsewari, A.A. A new algorithm for normal and large-scale optimization problems: Nomadic People Optimizer. *Neural Comput. Appl.* **2020**, *32*, 10359–10386. [CrossRef]
59. Połap, D.; Woźniak, M. Polar bear optimization algorithm: Meta-heuristic with fast population movement and dynamic birth and death mechanism. *Symmetry* **2017**, *9*, 203. [CrossRef]
60. Mirjalili, S. Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Comput. Appl.* **2016**, *27*, 1053–1073. [CrossRef]
61. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [CrossRef]
62. Joshi, A.S.; Kulkarni, O.; Kakandikar, G.M.; Nandedkar, V.M. Cuckoo search optimization-a review. *Mater. Today: Proc.* **2017**, *4*, 7262–7269. [CrossRef]
63. Bairwa, A.K.; Joshi, S.; Singh, D. Dingo Optimizer: A Nature-Inspired Metaheuristic Approach for Engineering Problems. *Math. Probl. Eng.* **2021**, *2021*, 2571863. [CrossRef]
64. Mirjalili, S.; Mirjalili, S.M. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Adv. Eng. Softw.* **2017**, *114*, 163–191. [CrossRef]
65. Shami, T.M.; El-Saleh, A.A.; Alswaitti, M.; Al-Tashi, Q.; Summakieh, M.A.; Mirjalili, S. Particle Swarm Optimization: A Comprehensive Survey. *IEEE Access* **2022**, *10*, 10031–10061. [CrossRef]
66. Gad, A.G. Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review. *Arch. Comput. Methods Eng.* **2022**, *29*, 2531–2561. [CrossRef]
67. Wang, D.; Tan, D.; Liu, L. Particle swarm optimization algorithm: An overview. *Soft Comput.* **2018**, *22*, 387–408. [CrossRef]
68. Ashrafzadeh, M.; Taheri, H.M.; Gharehgozlou, M.; Hashemkhani Zolfani, S. Clustering-based return prediction model for stock pre-selection in portfolio optimization using PSO-CNN+MVF. *J. King Saud Univ. Comput. Inf. Sci.* **2023**, *35*, 101737. [CrossRef]
69. Kwok, N.M.; Wang, D.; Ha, Q.P.; Fang, G.; Chen, S.Y. Locally-Equalized Image Contrast Enhancement Using PSO-Tuned Sectorized Equalization. In *Computational Intelligence in Image Processing*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 21–36. [CrossRef]
70. Hosseini, M.S.; Jahanshahlou, F.; Akbarzadeh, M.A.; Zarei, M.; Vaez-Gharamaleki, Y. Formulating research questions for evidence-based studies. *J. Med. Surg. Public Health* **2024**, *2*, 100046. [CrossRef]
71. Carrera-Rivera, A.; Ochoa, W.; Larrinaga, F.; Lasa, G. How-to conduct a systematic literature review: A quick guide for computer science research. *MethodsX* **2022**, *9*, 101895. [CrossRef]
72. Haddaway, N.R.; Page, M.J.; Pritchard, C.C.; McGuinness, L.A. PRISMA2020: An R package and Shiny app for producing PRISMA 2020-compliant flow diagrams, with interactivity for optimised digital transparency and Open Synthesis. *Campbell Syst. Rev.* **2022**, *18*, e1230. [CrossRef]
73. Abboud, A.; Abouaissa, A.; Shahin, A.; Mazraani, R. A Hybrid Aggregation Approach for Federated Learning to Improve Energy Consumption in Smart Buildings. In Proceedings of the 2023 International Wireless Communications and Mobile Computing (IWCMC), Marrakesh, Morocco, 19–23 June 2023; pp. 854–859. [CrossRef]
74. Supriya, Y.; Victor, N.; Srivastava, G.; Gadekallu, T.R. A Hybrid Federated Learning Model for Insurance Fraud Detection. In Proceedings of the WS23 IEEE ICC 2023 Workshop on CyberNet: Cyber-Physical Security in Mission-Critical Tactical Networks, Rome, Italy, 28 May–1 June 2023; pp. 1516–1522.
75. Xing, P.; Zhang, H.; Derbali, M.; Sefat, S.M.; Alharbi, A.H.; Khafaga, D.S.; Sani, N.S. An efficient algorithm for energy harvesting in IIoT based on machine learning and swarm intelligence. *Heliyon* **2023**, *9*. [CrossRef]
76. Houssein, E.H.; Sayed, A. Boosted federated learning based on improved Particle Swarm Optimization for healthcare IoT devices. *Comput. Biol. Med.* **2023**, *163*, 107195. [CrossRef]
77. Fan, X.; Wang, Y.; Huo, Y.; Tian, Z. CB-DSL: Communication-efficient and byzantine-robust distributed swarm learning on non-iid data. *IEEE Trans. Cogn. Commun. Netw.* **2023**, *10*, 322–334. [CrossRef]
78. Shang, C.; Gu, F.; Jiang, J. Evolutionary Multi-model Federated Learning on Malicious and Heterogeneous Data. In Proceedings of the 2023 IEEE International Conference on Data Mining Workshops (ICDMW), Shanghai, China, 1–4 December 2023; pp. 386–395.
79. Mendes, N.; Moura, P.; Mendes, J.; Antunes, C.H.; Mohammadi, J. Federated Learning Optimization for Energy Communities in Smart Cities. In Proceedings of the 2023 IEEE International Smart Cities Conference (ISC2), Bucharest, Romania, 24–27 September 2023; pp. 1–7.
80. Park, S.; Suh, Y.; Lee, J. FedPSO: Federated learning using particle swarm optimization to reduce communication costs. *Sensors* **2021**, *21*, 600. [CrossRef] [PubMed]
81. Victor, N.; Bhattacharya, S.; Maddikunta, P.K.R.; Alotaibi, F.M.; Gadekallu, T.R.; Jhaveri, R.H. FL-PSO: A Federated Learning approach with Particle Swarm Optimization for Brain Stroke Prediction. In Proceedings of the 23rd IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing Workshops, CCGridW 2023, Bangalore, India, 1–4 May 2023; pp. 33–38. [CrossRef]
82. Li, Z.; Li, H.; Zhang, M. Hyper-parameter Tuning of Federated Learning Based on Particle Swarm Optimization. In Proceedings of the 7th IEEE International Conference on Cloud Computing and Intelligence Systems, CCIS 2021, Xi'an, China, 7–8 November 2021; pp. 99–103. [CrossRef]
83. Devi, R. IPSHO-Fed: A hybrid federated learning and spotted hyena optimization approach for trust assessment. *Neural Comput. Appl.* **2024**, *36*, 5571–5594. [CrossRef]

84. Połap, D.; Woźniak, M. Meta-heuristic as manager in federated learning approaches for image processing purposes. *Appl. Soft Comput.* **2021**, *113*, 107872. [CrossRef]
85. Qolomany, B.; Ahmad, K.; Al-Fuqaha, A.; Qadir, J. Particle Swarm Optimized Federated Learning for Industrial IoT and Smart City Services. In Proceedings of the GLOBECOM 2020—2020 IEEE Global Communications Conference, Taipei, Taiwan, 7–11 December 2020. [CrossRef]
86. Kishore, P.; Barisal, S.K.; Prasad Mohapatra, D. Particle Swarm Optimized Federated Learning for Securing IoT Devices. In Proceedings of the 2021 IEEE Globecom Workshops, GC Wkshps 2021—Proceedings, Madrid, Spain, 7–11 December 2021; pp. 1–6. [CrossRef]
87. Torra, V.; Galván, E.; Navarro-Arribas, G. PSO + FL = PAASO: Particle swarm optimization + federated learning = privacy-aware agent swarm optimization. *Int. J. Inf. Secur.* **2022**, *21*, 1349–1359. [CrossRef]
88. Bakir, N.; Samrouth, A.; Samrouth, K. PSO-GA-based Federated Learning for Predicting Energy Consumption in Smart Buildings. In Proceedings of the International Conference on Microelectronics, ICM, Abu Dhabi, United Arab Emirates, 17–20 December 2023; pp. 234–238. [CrossRef]
89. Liu, Z.; Zhu, T.; Liu, Z.; Ning, H.; Chen, L. Reducing Communication Costs of Federated Contrastive Learning by Particle Swarm Optimization. In Proceedings of the 3rd International Academic Exchange Conference on Science and Technology Innovation, IAECST 2021, Guangzhou, China, 10–12 December 2021; pp. 687–692. [CrossRef]
90. Abdul Salam, M.; Badr, E.S.; Monir, E.; Atef, A. Schizophrenia Diagnosis using Optimized Federated Learning Models. *Int. J. Comput. Sci. Netw. Secur.* **2022**, *22*, 829–838. [CrossRef]
91. Miyajima, H.; Shigei, N.; Miyajima, H.; Shiratori, N. Securely Distributed Computation with Divided Data and Parameters for Hybrid Particle SwarmOptimization. *IAENG Int. J. Appl. Math.* **2022**, *52*.
92. Cai, W.; Duan, F. Task Scheduling for Federated Learning in Edge Cloud Computing Environments by Using Adaptive-Greedy Dingo Optimization Algorithm and Binary Salp Swarm Algorithm. *Future Internet* **2023**, *15*, 357. [CrossRef]
93. Geyer, R.C.; Klein, T.; Nabi, M. Differentially private federated learning: A client level perspective. *arXiv* **2017**, arXiv:1712.07557.
94. De Falco, I.; Cioppa, A.D.; Koutny, T.; Ubl, M.; Krcma, M.; Tarantino, U.S.E. A Federated Learning-Inspired Evolutionary Algorithm: Application to Glucose Prediction. *Sensors* **2023**, *23*, 2957. [CrossRef]
95. Mahlool, D.H.; Abed, M.H. A comprehensive survey on federated learning: Concept and applications. In *Mobile Computing and Sustainable Informatics: Proceedings of ICMCSI 2022*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 539–553. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.