

## Article

# A Communication-Efficient Federated Learning Framework for Sustainable Development Using Lemurs Optimizer

Mohammed Azmi Al-Betar <sup>1,\*</sup>, Ammar Kamal Abasi <sup>2</sup>, Zaid Abdi Alkareem Alyasseri <sup>3,4</sup>, Salam Fraihat <sup>1</sup>  
and Raghad Falih Mohammed <sup>5</sup>

- <sup>1</sup> Artificial Intelligence Research Center (AIRC), College of Engineering and Information Technology, Ajman University, Ajman P.O. Box 346, United Arab Emirates; s.fraihat@ajman.ac.ae
- <sup>2</sup> Department of Machine Learning, Mohamed Bin Zayed University of Artificial Intelligence (MBZUAI), Abu Dhabi P.O. Box 131818, United Arab Emirates; ammar.abasi@mbzuai.ac.ae
- <sup>3</sup> Information Technology Research and Development Center (ITRDC), University of Kufa, Najaf 54001, Iraq; zaid.alyasseri@uokufa.edu.iq
- <sup>4</sup> College of Engineering, University of Warith Al-Anbiyaa, Karbala P.O. Box 56001, Iraq
- <sup>5</sup> Department of Business Administration, College of Administrative and Financial Sciences, Imam Ja'afar Al-Sadiq University, Baghdad P.O. Box 10011, Iraq; raghad.falih@ijsu.edu.iq
- \* Correspondence: mohbetar@bau.edu.jo

**Abstract:** The pressing need for sustainable development solutions necessitates innovative data-driven tools. Machine learning (ML) offers significant potential, but faces challenges in centralized approaches, particularly concerning data privacy and resource constraints in geographically dispersed settings. Federated learning (FL) emerges as a transformative paradigm for sustainable development by decentralizing ML training to edge devices. However, communication bottlenecks hinder its scalability and sustainability. This paper introduces an innovative FL framework that enhances communication efficiency. The proposed framework addresses the communication bottleneck by harnessing the power of the Lemurs optimizer (LO), a nature-inspired metaheuristic algorithm. Inspired by the cooperative foraging behavior of lemurs, the LO strategically selects the most relevant model updates for communication, significantly reducing communication overhead. The framework was rigorously evaluated on CIFAR-10, MNIST, rice leaf disease, and waste recycling plant datasets representing various areas of sustainable development. Experimental results demonstrate that the proposed framework reduces communication overhead by over 15% on average compared to baseline FL approaches, while maintaining high model accuracy. This breakthrough extends the applicability of FL to resource-constrained environments, paving the way for more scalable and sustainable solutions for real-world initiatives.

**Keywords:** sustainable development; federated learning (FL); communication-efficient FL; lemurs optimizer (LO); metaheuristics



**Citation:** Al-Betar, M.A.; Abasi, A.K.; Alyasseri, Z.A.A.; Fraihat, S.; Mohammed, R.F. A Communication-Efficient Federated Learning Framework for Sustainable Development Using Lemurs Optimizer. *Algorithms* **2024**, *17*, 160. <https://doi.org/10.3390/a17040160>

Received: 6 March 2024

Revised: 7 April 2024

Accepted: 11 April 2024

Published: 15 April 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

The imperative pursuit of sustainable development, encompassing the multifaceted dimensions of social, environmental, and economic well-being, demands innovative tools that effectively address complex challenges [1]. These challenges include broad issues like poverty and climate change, but also more specific and pressing concerns such as rice leaf disease [2] and implementing robust waste classification [3]. Tackling these intricacies requires a holistic approach that leverages cutting-edge solutions and embraces technological advancements to drive positive change and foster a resilient future for our planet. In this context, machine learning (ML) emerges as a potent force, enabling data-driven insights and solutions across diverse domains such as renewable energy optimization [4], climate change mitigation [5], and personalized healthcare [6]. However, traditional centralized

ML approaches encounter limitations, particularly concerning data privacy concerns and resource constraints, especially in geographically distributed settings [7].

Federated learning (FL) is a transformative paradigm for sustainable development by decentralizing the ML model training process to the edges, where data reside on individual devices or local servers [8,9]. This decentralized approach preserves data privacy and alleviates resource limitations, all while harnessing the collective power of geographically dispersed data [10]. Nonetheless, a fundamental challenge in FL lies in the communication bottleneck between the central server and the myriad of edge devices [11]. The frequent exchange of model updates during training incurs significant communication overhead, impeding scalability and sustainability in resource-constrained environments [12].

Metaheuristic algorithms, characterized by their ability to explore solution spaces efficiently, have become indispensable tools in overcoming challenges associated with optimization problems [13]. In the context of FL, where communication bottlenecks pose significant hurdles, leveraging metaheuristic algorithms offers a powerful strategy to enhance the efficiency of the training process [14].

Metaheuristic algorithms draw inspiration from natural optimization phenomena, often mirroring the evolutionary processes observed in biological systems, swarm behaviors, or physical phenomena [15,16]. These algorithms exhibit a capacity for global optimization by navigating vast solution spaces in search of optimal or near-optimal solutions [17]. Importantly, metaheuristic algorithms are particularly well-suited for complex, non-linear, and high-dimensional problems, making them invaluable in the realm of ML [18].

The essence of metaheuristic algorithms lies in their ability to strike a delicate balance between exploration and exploitation [19]. Exploration involves systematically searching the solution space to discover new potential solutions, while exploitation focuses on refining and exploiting promising solutions to improve overall performance [20]. This balance is crucial in achieving a convergence of the algorithm towards optimal solutions while avoiding premature convergence to suboptimal solutions [21,22]. In the case of FL, the communication bottleneck arises due to the constant exchange of model updates between the central server and distributed edge devices [23]. Metaheuristic algorithms offer an elegant solution by guiding the selection of relevant model updates strategically [24]. By minimizing the communication cost associated with sharing updates, these algorithms alleviate the strain on the network, enabling more scalable and sustainable FL systems [25].

Among the myriad metaheuristic algorithms, the Lemurs optimizer (LO) is particularly noteworthy for its distinctive qualities [26]. The LO is inspired by the cooperative foraging behavior of lemurs, showcasing a unique optimization approach. Its efficiency lies in its ability to dynamically adapt its exploration and exploitation strategies based on the evolving characteristics of the optimization landscape [27]. This adaptability makes the LO well-suited for dynamic environments, where the optimization landscape may change over time [28]. Furthermore, the LO efficiently identifies and exploits promising regions of the solution space, ensuring that the most relevant and high-quality model updates are selected for communication [29,30]. This reduces communication overhead and contributes to the FL framework's overall effectiveness, enhancing the convergence speed and robustness of the learning process.

This paper introduces a pioneering communication-efficient FL framework harnessing the power of the LO. The proposed framework significantly reduces communication overhead while preserving model accuracy by dynamically selecting the best client models to share and updating the global model accordingly. This breakthrough enhances the scalability and sustainability of FL in resource-constrained scenarios and extends its applicability to real-world sustainable development initiatives. Through empirical studies and comparisons, we delve into the intricate details of the proposed framework, contributing a comprehensive understanding of how the integration of FL, metaheuristic algorithms, and, specifically, the LO can advance the field and foster sustainable solutions in the era of ML. The primary contributions can be summarized as follows:

- Introduces a novel FL framework employing the LO for efficient communication.

- Achieves significant communication cost reduction compared to existing methods.
- Maintains comparable or superior model performance on diverse datasets.
- Enables more scalable and sustainable FL applications for real-world development goals.

The structure of this paper is as follows. In Section 2, the relevant literature is reviewed. A detailed explanation of the Lemurs optimizer is provided in Section 3, and, in Section 4, the FedLO method is presented. The results are analyzed and discussed in Section 5, the findings are concluded, and an outline for future work is provided in Section 6.

## 2. Related Work

The field of FL has witnessed substantial attention in recent years [31,32], driven by the challenges posed by communication between clients in the context of mobile devices. Numerous studies have focused on improving FL performance, addressing issues stemming from the unstable network environment of mobile devices [33]. Problems such as frequent node crashes, dynamic node group shifts, elevated central server overhead, and increased latency with a growing number of nodes have been identified [12,34]. Additionally, the use of multi-layer models to enhance learning accuracy has been explored, albeit accompanied by challenges related to the increasing number of weights as the layers deepen. The data size constraint in FL due to network transmission between the server and client has also been a prominent concern [35].

Efforts to enhance network performance in FL have involved innovative approaches such as low rank and random mask techniques and temporal weights [36,37]. However, while aiming to mitigate challenges, these methods may compromise accuracy in unstable network environments. Furthermore, security threats associated with traditional FL models have been identified, emphasizing the potential dangers of transmitting all model weights to the server. Minimizing the collection of weights on the server has emerged as a critical focus to address these security concerns. Simultaneously, the utilization of the Particle Swarm Optimization (PSO) algorithm in distributed environments has gained traction. Various PSO algorithms, including dynamic multi-swarm PSO (DMS-PSO), PSO for neighbor selection in peer-to-peer (P2P) networks, and gossip-based PSO for maintaining flexible P2P networks, have been explored [38,39]. Studies have demonstrated the adaptability of PSO in diverse distributed environments, showcasing its potential in addressing optimization challenges.

Park et al. [12] identified FL as a method of preserving data privacy by collecting only learned models on a server. The study highlights the challenges of limited communication bandwidth in FL clients, especially in unstable network environments. To address this, the paper introduced the Federated PSO (FedPSO) algorithm, leveraging PSO to transmit score values instead of large weights. FedPSO demonstrates a significant reduction in data used in network communication, leading to an average accuracy improvement of 9.47% and a 4% reduction in accuracy loss on an unstable network. In 2022, two papers introduced new algorithms based on FedPSO, FedGWO [25] and FedSCA [23], extending the exploration of FL techniques. FedGWO reduces data communications by employing the Grey Wolf Optimizer (GWO) algorithm, transmitting score principles rather than all client models' weights. This approach results in a 13.55% average improvement in global model accuracy while decreasing the required data capacity for network communication. Meanwhile, FedSCA proposes a Federated Sine Cosine Algorithm (SCA), utilizing the SCA mechanism for weight updating. FedSCA demonstrates a significant reduction in data used in network communication, leading to an average accuracy improvement of 9.87% over FedAvg and 2.29% over FedPSO, along with a 4.3% improvement in accuracy loss under unstable network conditions. These studies collectively highlight the evolving landscape of FL, showcasing a spectrum of innovative algorithms and techniques to address the unique challenges posed by communication, security, and accuracy in FL environments.

The LO has proven highly effective in optimizing tasks across various domains. This section provides an overview of current research on the algorithm, showcasing its applications and advancements. In paddy disease detection, the Modified LO algorithm employs

a filter-based feature transformation technique inspired by the Sine Cosine Optimization method [29]. This modification has been demonstrated to improve the accuracy of paddy disease detection through ML techniques, as seen in a study using thermal images of paddy leaves. When it comes to feature selection in high-dimensional datasets, the Enhanced LO (ELO) algorithm integrates Opposition-Based Learning (OBL) and Local Search Algorithm (LSA) to achieve superior accuracy compared to other competitive algorithms across various datasets [27]. An optimal metaheuristic improvement in arrhythmia classification using ECG signals involves a hybrid deep learning model with attention-based features and weighted feature integration [30]. The Recommended Adaptive Risk Rate-based Lemurs Optimization algorithm (ARR-LO) optimizes weighted fused feature selection, resulting in superior performance compared to conventional prediction models. The LO algorithm is vital in developing a transformer-based attention LSTM network tailored for Alzheimer's disease detection using EEG signals [28]. The Enhanced Wild Geese Lemurs Optimizer (EWGLO) embeds the algorithm to optimize weight values and achieve high accuracy, surpassing other state-of-the-art methods. The LO algorithm is also applied in medical image fusion via the Enhanced Fitness-aided Election-Based and Lemur Optimizer (FEBLO) [40]. The algorithm optimizes parameters in the Adaptive Quaternion Wavelet Transform (AQWT) model, successfully fusing multi-modal medical images and overcoming challenges posed by conventional models.

Overall, the LO algorithm's versatility and effectiveness are demonstrated across diverse applications, including disease detection, feature selection, classification tasks, and medical image fusion. Its ability to optimize various computational models positions it as a valuable tool in different domains.

### 3. Lemurs Optimizer (LO)

The LO algorithm represents a cutting-edge, nature-inspired metaheuristic explicitly designed for tackling global optimization problems [26,41–43]. Taking inspiration from the intricate social behaviors observed in lemurs, a species of prosimian primates native to Madagascar and neighboring islands, LO mirrors the distinctive locomotive behaviors of these animals. In organized social groups known as troops, lemurs exhibit two notable locomotive behaviors: leaping and dance-hopping. The former involves agile, long jumps between trees, spanning considerable distances to pursue resources and shelter. On the other hand, dance-hopping refers to a coordinated, communal movement performed by lemurs within their troops.

LO incorporates these natural lemur behaviors into its optimization process, where an agent represents each potential solution termed a "lemur". The positions of these lemurs correspond to candidate solutions within the defined search space. To initiate the algorithm, a population of lemurs is randomly placed within the variable bounds, as expressed by Equation (1):

$$L_i^j = \text{rand} \times (ub_j - lb_j) + lb_j \quad \forall i \in (1, 2, \dots, n), \forall j \in (1, 2, \dots, d) \quad (1)$$

where  $\text{rand}$  is a random number within the range of 0 to 1, and  $ub$  and  $lb$  denote the upper and lower bounds for the decision variable  $j$  in solution  $i$ .

Subsequently, the fitness of each lemur is assessed using an objective function, determining the current global best ( $gbest$ ) and nearest neighbors best ( $nbest$ ).

During the exploration phase, mirroring the leaping behavior, lemurs execute long jumps according to Equation (2):

$$L_{i+1}^j = L_i^j + \text{abs}(L_i^j - gbest^j) \times (\text{rand} - 0.5) \times 2 \quad \text{if } \text{rand} \geq \text{FRR} \quad (2)$$

In the exploitation phase, reminiscent of dance-hopping, lemurs engage with neighboring lemurs as per Equation (3):

$$L_{i+1}^j = L_i^j + \text{abs}(L_i^j - nbest^j) \times (\text{rand} - 0.5) \times 2 \quad \text{if } \text{rand} < \text{FRR} \quad (3)$$

The dynamic adjustment of the risk parameter FRR using Equation (4) ensures adaptability throughout the optimization process:

$$FRR = HRR - \text{Crnt\_Iter} \times \left( \frac{(HRR - LRR)}{\text{Max\_Iter}} \right) \quad (4)$$

where  $HRR$  and  $LRR$  represent constant predefined values.  $\text{Crnt\_Iter}$  denotes the current iteration;  $\text{Max\_Iter}$  is the maximum iterations' number. This adaptive mechanism equips LO with the capability to navigate and optimize complex problem landscapes effectively.

#### 4. Federated Lemurs Optimization Algorithm (FedLO)

Federated LO (FedLO) combines the principles of the LO with the FL paradigm to address global optimization problems in a distributed manner. Inspired by the agile and coordinated behaviors of lemurs, LO is adapted to operate in an FL setting where data are distributed among multiple clients. Each client, analogous to a lemur in the natural environment, contributes to the global optimization process.

The initialization of FedLO involves randomly placing a population of lemurs (representing potential solutions) within the variable bounds of the search space. In this context, the evaluation process involves discerning the global best ( $gbest$ ) and the best among nearest neighbors ( $nbest$ ). To facilitate this, lemurs engage in a strategic exploration phase, employing inspired long jumps reminiscent of their natural leaping behavior.

Mathematically, the objective function is expressed as follows:

$$\max_{\theta} \sum_{i=1}^N \frac{1}{|\mathcal{D}_i|} \sum_{(x,y) \in \mathcal{D}_i} \mathbb{I}(f_{\theta}(x_i) = y_i) \quad (5)$$

where  $\theta$  represents the parameters of the global model to be optimized, and  $N$  is the total number of participating clients.  $\mathcal{D}_i$  denotes the local dataset of client  $i$ .  $(x, y)$  represents a data point with input  $x$  and true label  $y$ .  $f_{\theta}(x_i)$  is the prediction made by the global model with parameters  $\theta$  for input  $x_i$ .  $\mathbb{I}(\cdot)$  is the indicator function, equal to 1 if the condition inside is true, and 0 otherwise. In essence, the objective function aims to maximize the average accuracy across all clients, considering the predictions made by the global model on their respective local datasets. Optimization involves identifying the parameters  $\theta$  that maximize this accuracy measure. In doing so, we enhance the overall performance of the federated model, ensuring a collective improvement in the accuracy and efficacy of the global model as it collaboratively learns from diverse local datasets. This comprehensive approach reflects the intricacies of the fitness evaluation system, contributing to the robustness and adaptability of the federated model in a distributed learning environment.

In the exploitation phase, resembling dance-hopping, which involves collaboration between lemurs. In FedLO, this corresponds to clients sharing their knowledge with the central server, which updates the global best position for the next iteration.

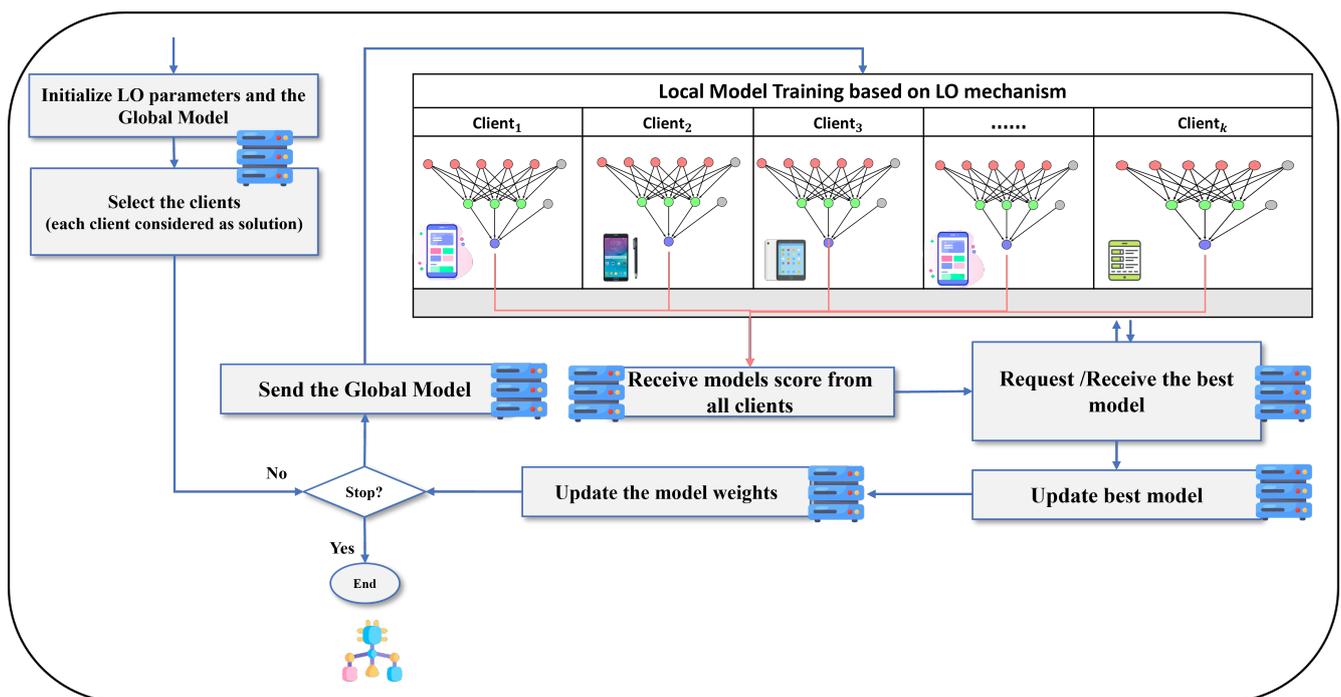
Additionally, FedLO incorporates client collaboration and information exchange mechanisms during the optimization process. After each exploration and exploitation phase, clients with improved lemurs, reflecting better potential solutions, share their findings with the central server. This collaboration enables the server to aggregate and update the global best position, promoting knowledge transfer across the federated network.

The collaboration step enhances FedLO's ability to adapt to diverse local landscapes encountered by individual clients. Lemurs exploring different regions of the search space contribute their unique insights, creating a collective intelligence that guides the optimization process. This decentralized collaboration aligns with the FL philosophy of leveraging local knowledge for global model improvement.

Furthermore, FedLO introduces adaptive mechanisms for adjusting exploration and exploitation parameters to accommodate the dynamic nature of federated environments. The risk parameter FRR, responsible for controlling the balance between exploration and exploitation, is continuously adapted based on the current iteration and the specified

maximum iteration count. This adaptability ensures that FedLO can respond effectively to changes in the optimization landscape throughout FL.

Inspired by leaping behavior, the exploration phase allows lemurs to take agile, long jumps towards potentially promising regions. This translates to clients exploring diverse areas of their local search spaces in a federated context. The proposed FedLO framework, illustrated in Figure 1, operates in a structured manner, commencing with initializing both local optimization parameters and the global model. Clients, treated as individual solutions, are strategically selected to participate in collaborative learning. The process then evaluates predefined stopping criteria; if unmet, the global model is transmitted to all chosen clients. Subsequently, each client undertakes local model training, employing local optimization mechanisms to fine-tune the model according to its specific dataset characteristics. Scores reflecting the performance of locally trained models are collected from all clients, and the framework checks whether all models have updated their weights. If not, the process continues by requesting and receiving the best-performing model among participants, updating it based on the received information. The iterative cycle persists until all models have completed their updates. Upon meeting the stopping criteria, the FL process concludes. FedLO ensures collaborative model improvement while considering the nuances of local data, fostering an effective and efficient FL environment. Notably, for each client in this study, the nearest solution corresponds to the previously identified best model in this paper.



**Figure 1.** FedLO framework.

The process of updating the weights in the FedLO algorithm is illustrated in Figure 1. First, the server initializes the model and FedLO parameters before distributing the model to clients. Each client trains the model using FedLO weights and computes a score based on the lowest loss or highest accuracy achieved. The client with the highest score sends this value to the server, which then updates the global model's weighted collection with the best client's model. The pseudocode for the FedLO is detailed in Algorithm 1, outlines a strategy for optimizing distributed client performance in FL environments.

**Algorithm 1** Federated Lemurs Optimization Algorithm (FedLO)

---

```

1: Initialize: Set algorithm parameters, client positions, and communication intervals.
2: Initialize global best position:  $g_{best}$ 
3: Initialize lemurs' positions on each client:  $L_i^j$ 
4: for each iteration  $itr = 1, 2, 3, \dots, Max_{itr}$  do
5:   for each client  $k$  do
6:     Evaluate fitness of lemurs on client  $k$ 
7:     Update  $n_{best}$  using lemurs' local best positions
8:   end for
9:   for each client  $k$  in parallel do
10:    for each lemur  $i$  on client  $k$  do
11:      Update lemurs' positions based on exploration and exploitation phases
12:    end for
13:  end for
14:  Communicate global best positions ( $g_{best}$ ) among clients
15:  for each client  $k$  do
16:    Update global best position based on received information
17:  end for
18: end for

```

---

**5. Results and Discussion**

This section meticulously examines the outcomes of the implemented methodology, serving as a critical juncture in exploring the presented research. It also provides a detailed account of the dataset utilized, offering insight into its composition and the methodology employed for its collection. A comprehensive understanding of the dataset is considered essential to contextualizing the subsequent analysis. Furthermore, a detailed overview of the sample extracted from the dataset is presented, shedding light on the characteristics that define the representative subset under scrutiny. This discussion sets the stage for a nuanced evaluation of the behavior of the proposed algorithm, accompanied by a comparative analysis against other existing algorithms. The performance of the proposed method is scrutinized alongside alternative approaches, aiming to discern distinctive patterns and efficiencies that contribute to a comprehensive understanding of the algorithmic landscape. Quantitative metrics such as accuracy and loss are delved into to gauge the efficacy of the proposed algorithm. These metrics serve as benchmarks, quantitatively measuring the algorithm's performance. Meticulous scrutiny of these values is undertaken, drawing correlations between the algorithm's predictive capabilities and its ability to minimize errors and discrepancies. The ensuing discussion section unveils the empirical outcomes and endeavors to interpret and contextualize these results within the broader scope of the research objectives.

**5.1. Dataset**

- CIFAR-10 dataset [44]: This dataset from the Canadian Institute for Advanced Research is a widely utilized dataset within computer vision (CV) and ML domains. Specifically tailored for image classification tasks, CIFAR-10 serves as a benchmark in the research and development of diverse image processing algorithms and models. Comprising 60,000 color images, each measuring  $32 \times 32$  pixels, the dataset is distributed across 10 distinct classes, with 6000 images allocated per class. These classes encompass common objects such as airplanes, automobiles, birds, cats, deer, dogs, frogs, horses, ships, and trucks. Key characteristics include each image's RGB (red, green, and blue) color channels. The dataset is divided into a training set, consisting of 50,000 images, and a testing set, containing 10,000 images. CIFAR-10's primary purpose is to serve as a standardized dataset for evaluating image classification algorithms, offering a realistic and challenging environment.

Despite the relatively small size of the images, CIFAR-10 remains instrumental for testing and comparing the efficacy of different models in addressing real-world image recognition challenges. Researchers commonly employ this dataset to develop and assess the performance of various ML models, especially convolutional neural networks (CNNs). The diverse set of classes in CIFAR-10 necessitates models to generalize effectively across different categories, contributing to its widespread use in the field.

- MNIST dataset [45]: This dataset is widely acknowledged and frequently employed within the fields of ML and CV. MNIST, an acronym for Modified National Institute of Standards and Technology, is a curated collection of handwritten digits used extensively for training and assessing image processing algorithms and models.

Essential characteristics of the MNIST dataset include 70,000 grayscale images featuring handwritten digits ranging from 0 to 9. Each image measures  $28 \times 28$  pixels. The dataset is divided into a training set comprising 60,000 images and a testing set containing 10,000 images.

MNIST serves as a benchmark for evaluating ML models' performance, particularly in recognizing handwritten digits. It is widely adopted as a standard dataset for developing and assessing image classification algorithms.

Challenges inherent in the MNIST dataset arise from the variability in writing styles, diverse penmanship, and potential noise in the images. Unlike datasets that include color images, MNIST images are grayscale, simplifying the input for models.

MNIST is notable for its historical significance. It represents one of the pioneering datasets used for testing and benchmarking ML algorithms, especially in the context of deep learning and neural networks.

Researchers commonly use MNIST as a foundational resource for experimenting with and validating image classification models. Despite its relative simplicity compared to more complex datasets, MNIST remains an essential tool for educational purposes, enabling practitioners to grasp and implement fundamental concepts in ML and computer vision.

- Rice leaf disease dataset [2]: This dataset is a comprehensive compilation of images representing various rice leaf diseases. This dataset is meticulously curated to train and evaluate models in the context of rice leaf disease recognition. A preprocessing stage was applied to each image to ensure consistency in dimensions and data quality. During this process, all images were resized to a uniform dimension of  $224 \times 224$  pixels, involving the standardization of dimensions without preserving the original aspect ratio. Additionally, normalization was performed to bring pixel values within the standardized range of  $[0, 1]$ .

The dataset is strategically divided into training, validation, and test sets, maintaining an 80%/10%/10% ratio, respectively. This division facilitates effective training and evaluation of ML models on distinct subsets of the data.

The dataset comprises images from five distinct classes, each corresponding to a specific rice leaf disease. These classes are blast, blight, brown spot, leaf smut, and tungro. Each class is represented by a varying number of images, contributing to the diversity and richness of the dataset. The distribution of images among classes is as follows: blast (80 images), blight (80 images), brown spot (40 images), leaf smut (40 images), and tungro (80 images). The dataset encompasses 320 images, providing a comprehensive and well-balanced representation of rice leaf diseases for effective model training and evaluation. As shown in Figure 2, samples of the rice leaf disease dataset highlight the visual representation of the different classes.

- WaRP dataset [46]: This dataset is designed to address the need for an extensive collection of images for training and testing the capabilities of waste recycling plants in automatically sorting recyclable items on conveyor belts. The dataset is specifically tailored to recognize and classify various waste categories found in recycling plants, including plastic and glass bottles, cardboard, detergents, canisters, and cans. The dataset, WaRP (waste recycling plant), consists of manually labeled images taken

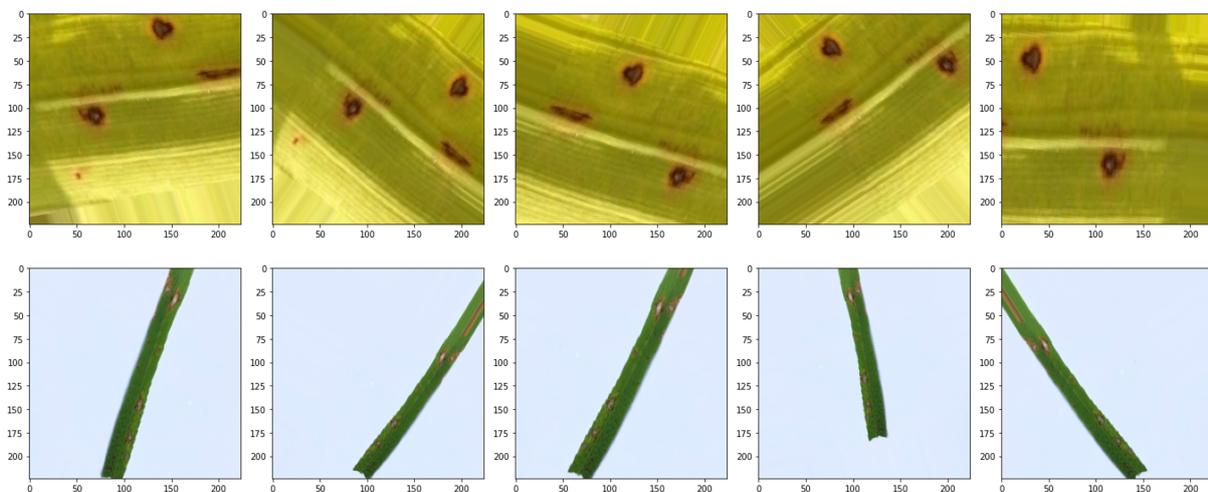
from an industrial conveyor. It encompasses 28 recyclable waste categories, with objects classified into groups such as plastic bottles, glass bottles, cardboard, detergents, canisters, and cans. Differentiation among categories is essential, considering the specific recycling technologies associated with each.

Unique features of the WaRP dataset include indicating whether a plastic bottle is filled with air (“full” postfix), which is crucial for correctly operating the conveyor’s manipulator. Unlike other datasets, the WaRP dataset allows for object overlap, heavy deformation, and variations in lighting conditions, simulating real-world scenarios encountered in recycling plants.

It is worth noting that the dataset is intentionally unbalanced, mirroring the naturally uneven distribution of household waste on a conveyor belt due to the varying frequency of use of different objects. For instance, there are more bottle images than canisters in the dataset.

The WaRP dataset is divided into three parts: WaRP-D for training and quality assessment of detection, WaRP-C for training and objective quality assessment of classification, and WaRP-S for validating weakly supervised segmentation methods. Each part serves a specific purpose, contributing to the overall effectiveness of the dataset in training and evaluating waste recognition models.

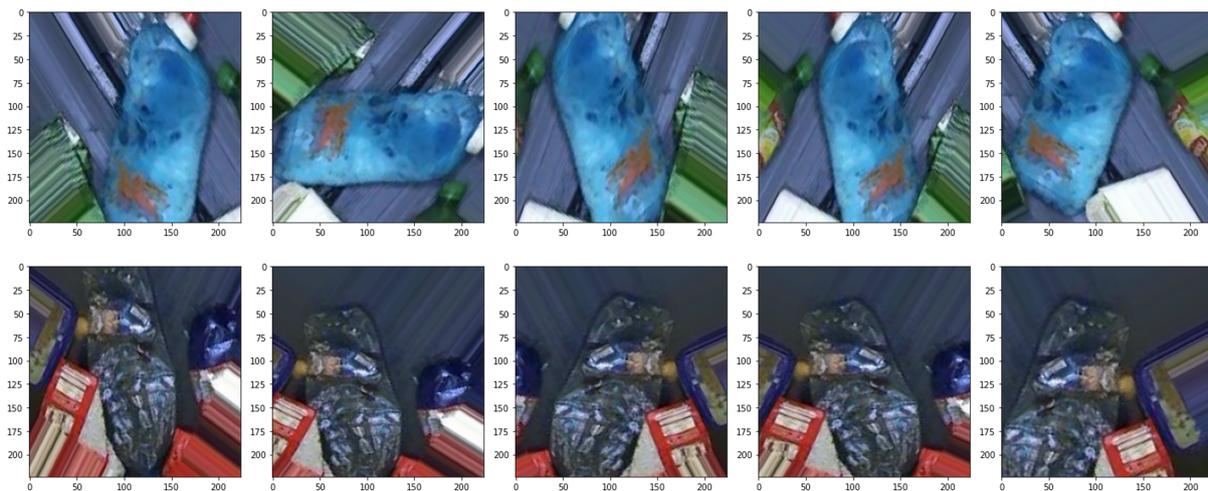
The dataset statistics reveal the distribution of images across different categories, providing insights into the number of training and testing samples for each waste category in the WaRP-D and WaRP-C parts. This comprehensive dataset aims to facilitate developing and testing methods for detecting, classifying, and segmenting waste in recycling plants, addressing the specific challenges encountered in real-world scenarios. As illustrated in Figure 3, dataset samples visually depict the variety of waste categories in the WaRP-C part.



**Figure 2.** Samples of rice leaf disease dataset.

It is important to note that the data partitioning strategy used in the proposed FedLO framework is crucial to the training process. Essentially, the training data are divided into subsets at random, ensuring that each client receives a diverse portion of the dataset for training. This method fosters collaborative learning while still maintaining data privacy.

An FL strategy is employed to handle heterogeneity in data distribution among clients. Each client trains their model independently using their local data subset. This decentralized approach enables clients to use their local data while also contributing to enhancing the global model. Additionally, model aggregation at the server allows for collaborative learning, which helps the model benefit from the collective knowledge of all clients.



**Figure 3.** Samples of WaRP dataset.

### 5.2. Comprehensive Model Architecture and Constant Configuration

This section presents a detailed analysis of the proposed model for dataset classification, illustrated in Figure 4. This architecture comprises a total of fourteen layers, including four convolutional layers (`conv2d`, `conv2d_1`, `conv2d_2`, `conv2d_3`) responsible for extracting hierarchical features. Additionally, four max pooling layers (`max_pooling2d`, `max_pooling2d_1`, `max_pooling2d_2`, `max_pooling2d_3`) contribute to spatial dimension reduction.

Layer (type)	Output Shape	Param #
<code>conv2d</code> (Conv2D)	(None, 222, 222, 16)	448
<code>max_pooling2d</code> (MaxPooling2D)	(None, 111, 111, 16)	0
<code>conv2d_1</code> (Conv2D)	(None, 109, 109, 32)	4640
<code>max_pooling2d_1</code> (MaxPooling2D)	(None, 54, 54, 32)	0
<code>conv2d_2</code> (Conv2D)	(None, 52, 52, 64)	18496
<code>max_pooling2d_2</code> (MaxPooling2D)	(None, 26, 26, 64)	0
<code>conv2d_3</code> (Conv2D)	(None, 24, 24, 128)	73856
<code>max_pooling2d_3</code> (MaxPooling2D)	(None, 12, 12, 128)	0
<code>flatten</code> (Flatten)	(None, 18432)	0
<code>dropout</code> (Dropout)	(None, 18432)	0
<code>dense</code> (Dense)	(None, 128)	2359424
<code>dropout_1</code> (Dropout)	(None, 128)	0
<code>dense_1</code> (Dense)	(None, 5)	645

**Figure 4.** Detailed summary of the model architecture for dataset classification.

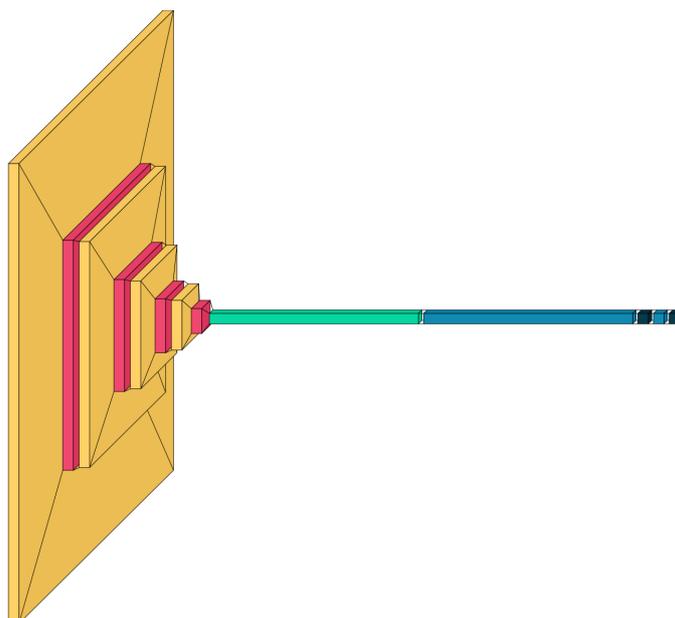
Furthermore, two flatten layers (`flatten`, `dropout`) transform the output into a one-dimensional vector, while two dense layers (`dense`, `dropout_1`) perform the final classification based on the learned features. The convolutional layers exhibit a progressive filter

increase, enabling the model to capture increasingly complex patterns. Output shapes, such as (None, 222, 222, 16) for the initial convolutional layer, represent the size of the data after each layer.

The model demonstrates a moderate number of parameters, indicating a balanced level of complexity. The abundance of trainable parameters enhances the model's adaptability during training. The proposed CNN model is tailored to effectively learn intricate features from images, balancing complexity and efficiency during training. Including dropout layers contributes to regularization, mitigating the risk of overfitting. A thorough model summary, encompassing activation functions and evaluation metrics, would be imperative to assess its performance comprehensively.

In this paper, a set of constants has been meticulously defined to shape the behavior and performance of the proposed framework. These constants encapsulate pivotal aspects such as batch processing, training epochs, and unique considerations for the FedAvg and the optimization algorithms. Notably, the batch size, client-epoch parameter, total number of epochs, "C" parameter (specifically for FedAvg), and the number of clients collectively represent the foundational building blocks of the model's configuration. The batch size, a critical determinant of the number of training instances processed in each iteration, is consistently set at 10 for both the optimization algorithms and the FedAvg approach. This choice reflects a strategic balance between computational efficiency and model performance. The client-epoch parameter, governing the number of training epochs completed by each client in the FL setup, is maintained at 5 for both optimization methods. This standardization ensures a uniform learning experience across clients, contributing to the overall stability of the training process. The total number of epochs, a fundamental factor influencing the duration and depth of the model training, is universally established at 30 for both the optimization algorithms and the FedAvg approach. This parameter plays a crucial role in determining the overall convergence and performance of the model. The "C" parameter, a specific consideration for the FedAvg, assumes values of 0.1, 0.2, 0.5, and 1.0. This parameter controls the number of clients during the aggregation process in FL. Furthermore, the number of clients participating in the FL process is constant at 10. This stability ensures consistent collaboration and diversity in training, contributing to the model's robustness and generalization capability.

Using Keras, the model architecture is visually represented in Figure 5. This diagram provides an intuitive overview of the connections and flow within the neural network, emphasizing layers' arrangement and respective operations.



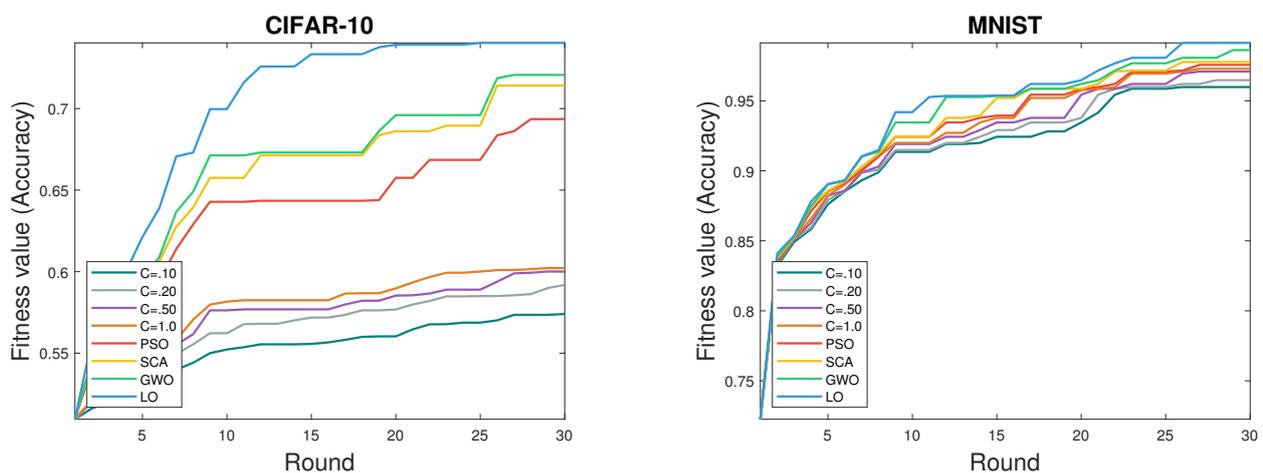
**Figure 5.** Visual representation of the proposed CNN model architecture using Keras.

The FedLO framework utilizes fixed values for the HRR and LRR parameters, as specified in the original LO algorithm. Specifically, the HRR parameter is set to 0.5, and the LRR parameter is set to 0.1. These parameters dictate the risk-taking behavior of the optimization process during training.

### 5.3. Performance Analysis of FedLO

This subsection examines the performance of the LO algorithm in the FL scenario. The focus is on comparing LO's behavior with other optimization algorithms across various datasets.

Figure 6a presents the accuracy results of the proposed algorithm and other algorithms in the context of FL, specifically across 30 rounds, using the CIFAR-10 dataset, examining the accuracy trends across various regularization parameter values ( $C$ ) in the context of FedAVG (Fed Aggregating) [47], where  $C$  represents the proportion of clients participating in the FedAVG process. A value of  $C = 1$  indicates the involvement of all clients,  $C = 0.10$  implies 10% participation,  $C = 0.20$  implies 20% participation, and  $C = 0.50$  implies 50% participation. This analysis shows that the LO algorithm consistently exhibits competitive or superior accuracy compared to other optimization algorithms under different  $C$  values in the FedAVG framework.



(a) Model accuracy over 100 epochs for CIFAR-10 dataset.

(b) Model accuracy over 100 epochs for MNIST dataset.

**Figure 6.** Model accuracy over 100 epochs for different datasets.

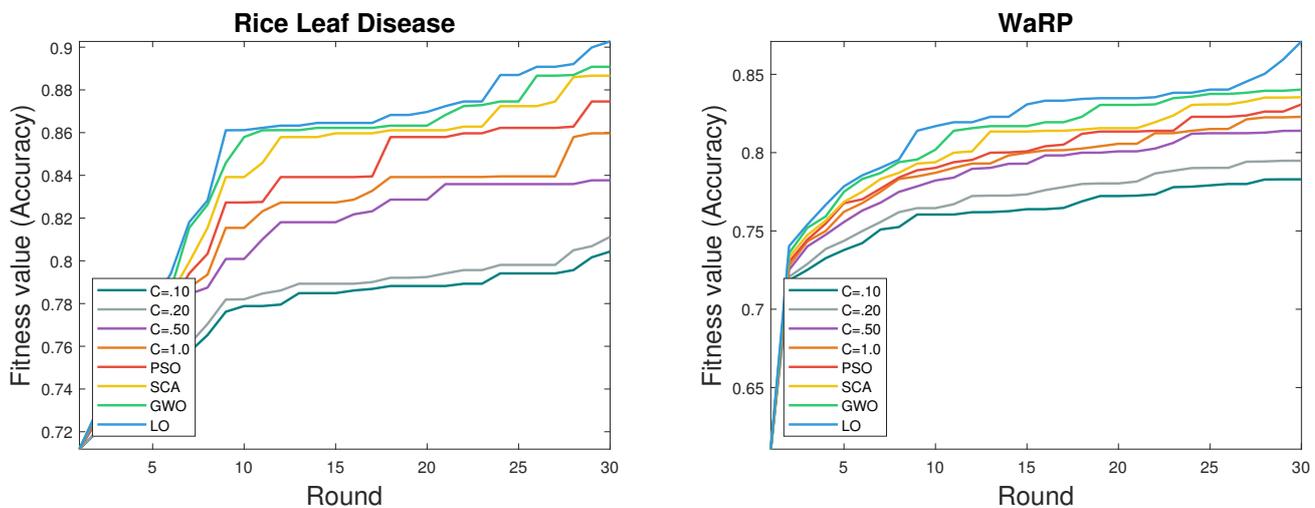
For FedAVG, increasing  $C$  from  $C = 0.10$  to  $C = 1.0$  results in a steady improvement in accuracy. Specifically, at  $C = 1.0$ , FedAVG exhibits a notable increase in accuracy, reaching 0.602195028, making it a robust contender in the FL context. Similarly, at  $C = 0.20$ , FedAVG maintains a competitive edge, achieving an accuracy of 0.59187885 over the 30 rounds. Although FedAVG's accuracy improves with higher  $C$  values, the focus remains on evaluating LO's performance. Compared to optimization algorithms like PSO, SCA, and GWO, LO is consistently accurate or superior in all 30 rounds; while PSO displays gradual improvement and SCA maintains consistent accuracy, LO delivers impressive results, making it an ideal choice for FL tasks.

In Figure 6b, the performances of various algorithms in FL using the MNIST dataset over 30 rounds are compared. The proposed FedLO algorithm and other algorithms like PSO, SCA, and GWO are evaluated.

The presented results highlight the superior performance of the proposed LO algorithm, FedLO, compared to other algorithms, particularly regarding accuracy. For instance, at  $C = 0.5$ , FedAVG achieves an accuracy of 96.2%, surpassing the accuracies achieved at  $C = 0.2$  and  $C = 0.1$ . Similarly, at  $C = 1.0$ , FedAVG attains an accuracy of 98.1%, outperforming the accuracy obtained at  $C = 0.5$ . Notably, the PSO method consistently enhances its accuracy, peaking at 0.9757, indicating its adaptability and effectiveness in optimizing

model parameters within the FL framework. The SCA method exhibits a steady accuracy increase, reaching a maximum of 0.9778. This consistent improvement throughout rounds underscores SCA's reliability and adaptability in FL scenarios. The GWO method demonstrates a noteworthy accuracy of 0.9863, showcasing its effectiveness in optimizing model parameters. The competitive performance of GWO suggests its suitability for FL tasks. The key strength of the LO algorithm lies in its ability to exploit local areas within the search space. This strategic approach leverages the inherent capability of the algorithm as a metaheuristic, allowing it to navigate and exploit the search space efficiently. This proficiency in exploring the solution space at a more localized level enhances its adaptability, enabling the algorithm to optimize model parameters based on the nuanced characteristics of each client's data. In contrast, global algorithms like PSO and SCA may struggle to capture and exploit these localized features, contributing to the observed superior performance of the LO algorithm in the context of metaheuristic optimization.

Figure 7a shows the performance of various algorithms in FL using the rice leaf disease dataset over 30 rounds.  $C = 0.1$  exhibits consistent improvement, reaching a peak accuracy of 0.8043. Similarly,  $C = 0.2$  demonstrates a positive trend, achieving a maximum accuracy of 0.8112.  $C = 0.5$  showcases substantial improvement, reaching an accuracy of 0.8377, while  $C = 1.0$  achieves a remarkable 0.8597 accuracy. PSO displays steady improvement, reaching an accuracy of 0.8746, demonstrating compelling exploration and exploitation. SCA exhibits a consistent positive trend, achieving an accuracy of 0.8867, indicating its adaptability to the federated data from the rice leaf disease dataset. GWO demonstrates gradual improvement, reaching an accuracy of 0.8908, showcasing a compelling exploration of the federated data space. The proposed LO method shows competitive performance, achieving an accuracy of 0.9027 and outperforming some established algorithms in convergence and accuracy.



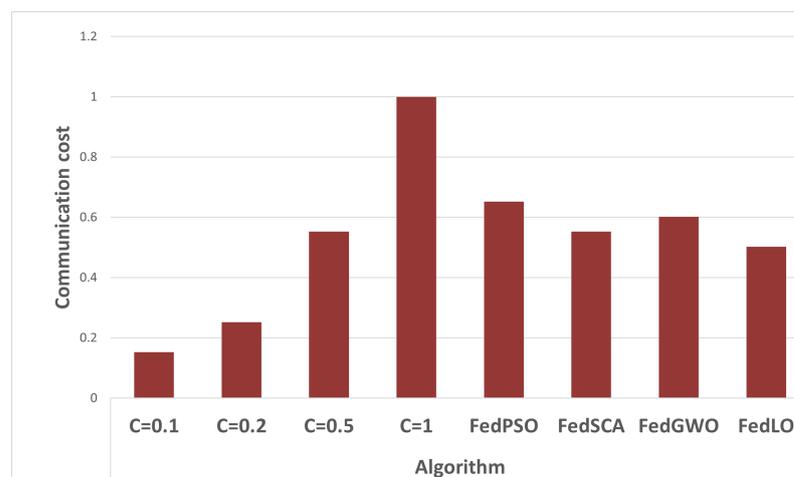
(a) Model accuracy over 100 epochs for Rice Leaf Disease dataset. (b) Model accuracy over 100 epochs for WaRP dataset.

**Figure 7.** Model accuracy over 100 epochs for different datasets.

Figure 7b shows the performance of various algorithms in FL using the WaRP dataset over 30 rounds.  $C = 0.1$  exhibits consistent improvement, reaching a peak accuracy of 0.7829. Similarly,  $C = 0.2$  demonstrates positive trends, achieving a maximum accuracy of 0.7948.  $C = 0.5$  showcases substantial improvement, reaching an accuracy of 0.8139, while  $C = 1.0$  achieves notable performance with a maximum accuracy of 0.8229. PSO demonstrates steady improvement, reaching an accuracy of 0.8308, effectively exploring and exploiting the WaRP dataset. SCA exhibits consistent positive trends, achieving an accuracy of 0.8354, showcasing its adaptability and convergence. GWO demonstrates gradual improvement, reaching an accuracy of 0.8402, effectively exploring the WaRP dataset. The proposed LO method shows competitive performance, achieving an accuracy

of 0.8709, surpassing some established optimization algorithms in terms of convergence and accuracy with the WaRP dataset.

Figure 8 shows the communication cost of different algorithmic approaches to FL on distributed datasets. It plots the communication cost for different values of  $C$  and several federated learning algorithms: FedPSO, FedSCA, FedGWO, and FedLO. According to the figure, the proposed algorithm has the lowest communication cost of 0.50 across all values of  $C$ . This result suggests that the proposed algorithm draws inspiration from cooperative foraging behavior in lemurs and is more efficient in communication than other algorithms based on different metaheuristic techniques. In particular, the FedLO algorithm can lower the communication overhead by up to 15% compared to standard federated learning approaches.



**Figure 8.** Comparing communication costs of different algorithms.

#### 5.4. Model Training Dynamics and Efficacy: A Deep Dive into Rice Leaf Disease and WaRP Datasets with FedLO Methodology

In the context of the rice leaf disease dataset, the presented results underscore the effectiveness of the FedLO method in enhancing model performance (see Figures 9 and 10). Across 30 training epochs, training and validation accuracy exhibited noticeable improvement. The training accuracy initiated at 21.48% in the first epoch and demonstrated a consistent upward trend, culminating at 87.50% in Epoch 30. Concurrently, the validation accuracy progressed from 28.12% to 82.81% over the same period. Complementing the accuracy metrics, the loss function, a crucial indicator of model efficacy, consistently decreased during training. The training loss started at 1.54 and steadily diminished to 0.29 by Epoch = 30. Similarly, the validation loss showed a reduction from 1.37 to 0.46. Occasional marginal increments in validation loss, notably in Epoch 2 and Epoch 8, suggested potential episodes of overfitting (see Figure 10). The general observations highlighted initial challenges in achieving satisfactory accuracy, which were gradually overcome as the model adapted and refined its predictions. The model attained a peak training accuracy of 87.50% and a validation accuracy of approximately 82.81% by Epoch 30. The diminishing loss function underscored the model's ability to extract meaningful insights from this dataset, enhancing its predictive capabilities. In conclusion, these results, obtained through the application of FedLO in this dataset, affirm the methodology's efficacy in progressively refining model performance over multiple training epochs. Further insights could be gained by delving into additional details on this dataset's characteristics, the employed model architecture, and specific nuances in implementing FedLO.

In the investigation of the WaRP dataset, the FedLO was applied to assess and enhance model performance (see Figures 11 and 12). The training spanned 30 epochs, and both training and validation accuracies and the loss metrics were tracked.

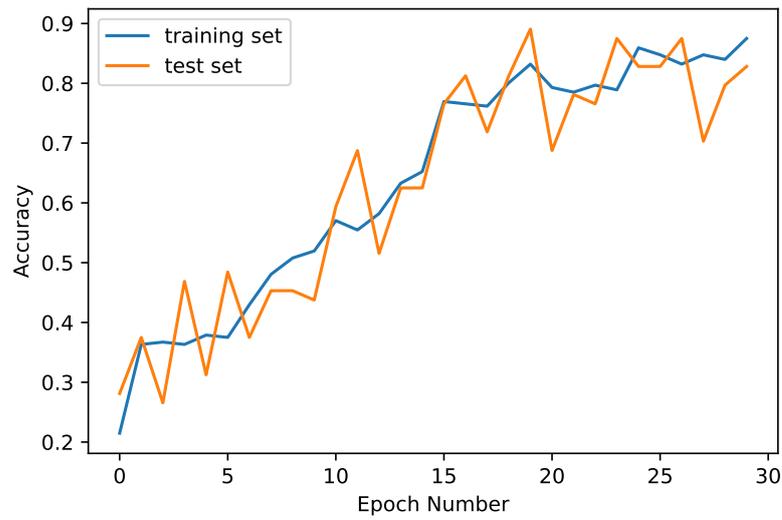


Figure 9. Training and validation accuracy—rice leaf disease dataset.

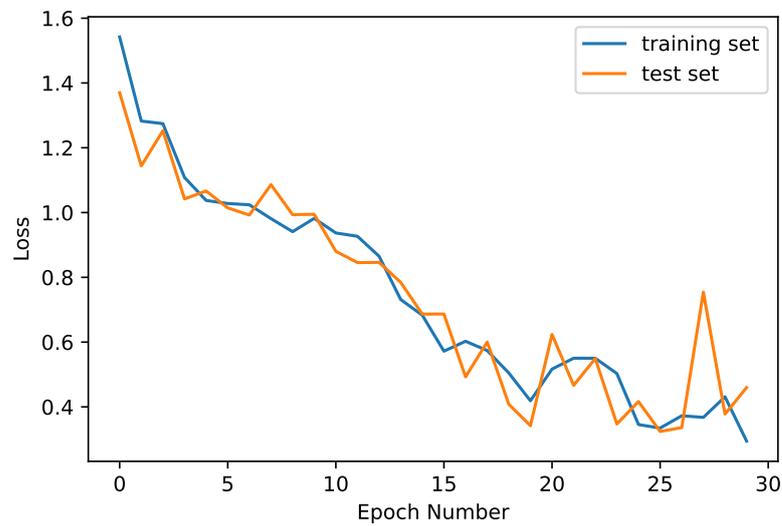


Figure 10. Training and validation loss—rice leaf disease dataset.

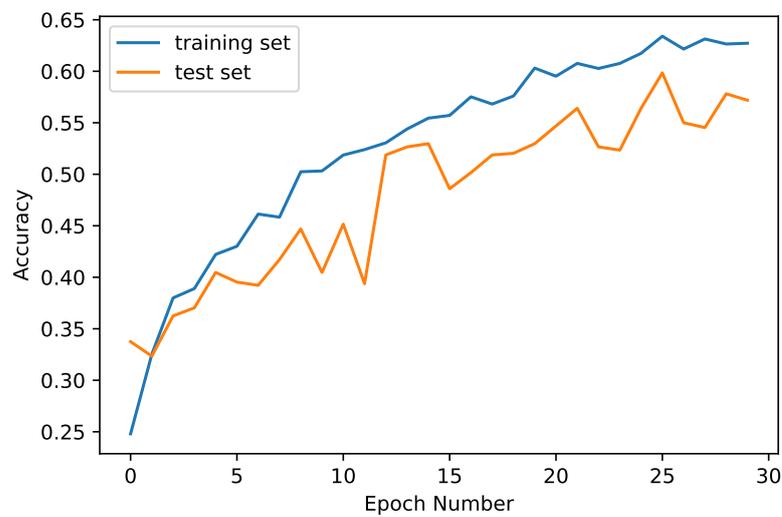


Figure 11. Training and validation accuracy—WaRP dataset.

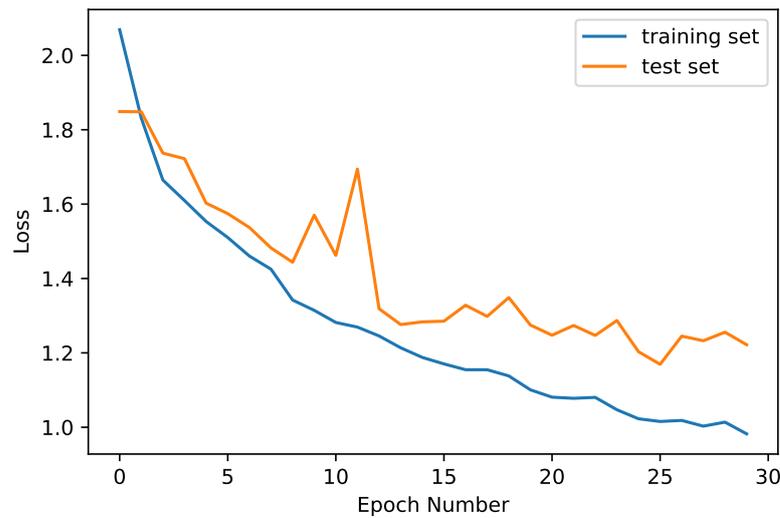


Figure 12. Training and validation loss—WaRP dataset.

The training accuracy commenced at 24.54% in the first epoch and steadily increased to 64.49% by Epoch 30. Simultaneously, the validation accuracy started at 31.56%, showing variability throughout training and reaching 53.28% by the conclusion of Epoch 30.

The loss function consistently reduced during training, with the training loss starting at 2.0912 and diminishing to 0.9842 by Epoch 30. Similarly, the validation loss decreased from 1.8839 to 1.2788, signifying effective learning and adaptation of the model to the DS4 dataset.

Observations indicate successes and challenges, with occasional fluctuations in accuracy and instances of potential overfitting, as suggested by the validation loss increments in certain epochs. However, the overall trajectory indicates a positive model learning and generalization trend.

The experimental findings presented in Table 1 shed light on the comparative performance of various FL algorithms across four datasets. Notably, the results highlight FedLO’s superior test accuracy compared to other algorithms, with the peak accuracy values presented in the table. At 30 epochs, FedLO consistently outperformed FedAvg across all datasets, showcasing its efficacy in achieving higher accuracy. This is a crucial observation, indicating that FedLO may offer a more robust and accurate solution in FL scenarios. In the context of FedAvg, the impact of the C on test accuracy is particularly noteworthy. The results in this table reveal that increasing the C leads to higher accuracy for FedAvg, reaching its peak at C = 1.0. However, this improvement comes at the cost of increased data transmission between the server and clients. The trade-off between accuracy and data transfer is evident, emphasizing the need for a balanced approach in FL settings. Looking at the individual datasets, FedLO consistently outperforms other algorithms, achieving the highest accuracy across CIFAR-10, MNIST, rice leaf disease, and WaRP datasets. Particularly in MNIST and WaRP datasets, FedLO demonstrates exceptional accuracy, reaching 99.14% and 87.10%, respectively.

Table 1. Test accuracy.

Algorithm \ Dataset	FedAvg, C = 0.1	C = 0.2	C = 0.5	C = 1.0	FedPSO	FedSCA	FedGWO	FedLO
CIFAR-10	0.5740	0.5919	0.6001	0.6022	0.6936	0.7141	0.7207	<b>0.7403</b>
MNIST	0.9598	0.9648	0.9709	0.9729	0.9757	0.9778	0.9863	<b>0.9914</b>
Rice leaf disease	0.8043	0.8112	0.8377	0.8597	0.8746	0.8867	0.8908	<b>0.9027</b>
WaRP	0.7829	0.7948	0.8139	0.8229	0.8308	0.8354	0.8402	<b>0.8710</b>

### 5.5. Unstable Network Experimental Result

Random data drops were introduced during communication rounds between the client and server to simulate an unstable network environment. The evaluation involved varying degrees of data drop (0%, 10%, 20%, and 50%) across different datasets (CIFAR-10, MNIST, rice leaf disease, and WaRP). To ensure the validity of the results, the outcomes were averaged over ten experiments. Table 2 displays the accuracy of FedAvg and alternative algorithms at different failure rates. FedAvg consistently declined accuracy as the failure rate increased across all datasets. For example, in CIFAR-10, accuracy ranged from 60.22% (0% failure) to 54.62% (50% failure). Similarly, in MNIST, accuracy varied from 97.29% (0% failure) to 92.31% (50% failure), indicating a significant impact of communication failures on model performance. In the rice leaf disease dataset, FedAvg showed accuracy ranging from 85.97% (0% failure) to 81.57% (50% failure), highlighting the algorithm's vulnerability to increased failure rates across diverse datasets. Additionally, in the WaRP dataset, the accuracy of FedAvg ranged from 82.29% (0% failure) to 77.48% (50% failure), further underscoring the consistent trend of declining accuracy under adverse network conditions. FedPSO consistently outperforms FedAvg across datasets and failure rates, indicating its potential for maintaining accuracy in the face of communication failures. For instance, in CIFAR-10, FedLO achieved an accuracy range of 69.36% (0% failure) to 64.11% (50% failure). Similarly, in MNIST, FedLO outperformed FedAvg with an accuracy range of 97.57% (0% failure) to 91.54% (50% failure), showcasing its resilience in unstable network environments. Furthermore, FedSCA and FedGWO also demonstrate competitive performance. In the WaRP dataset, FedGWO achieves an accuracy range of 84.02% (0% failure) to 79.00% (50% failure), showcasing its ability to cope with communication challenges. However, it is notable that FedLO consistently outperforms the other algorithms, emphasizing its superiority in maintaining accuracy under varying failure rates. In the WaRP dataset, FedLO achieves an accuracy range of 87.10% (0% failure) to 82.33% (50% failure), consistently surpassing FedAvg, FedPSO, FedSCA, and FedGWO. This highlights FedLO as a promising alternative for robust FL in unstable network conditions. These instances emphasize the importance of considering alternative FL algorithms, particularly FedLO, to mitigate the impact of communication failures.

**Table 2.** Accuracy against communication failure probability.

Dataset	Algorithm	Failure Rate 50%	20%	10%	0%
CIFAR-10	FedAvg, C = 1.0	0.5462	0.5731	0.5911	0.6022
	FedPSO	0.6411	0.6674	0.6812	0.6936
	FedSCA	0.6637	0.6923	0.7052	0.7141
	FedGWO	0.6676	0.6907	0.7092	0.7207
	FedLO	0.6879	0.7150	0.7320	0.7403
MNIST	FedAvg, C = 1.0	0.9231	0.9517	0.9619	0.9729
	FedPSO	0.9154	0.9448	0.9629	0.9757
	FedSCA	0.9354	0.9564	0.9680	0.9778
	FedGWO	0.9378	0.9602	0.9715	0.9863
	FedLO	0.9422	0.9665	0.9814	0.9914
Rice leaf disease	FedAvg, C = 1.0	0.8157	0.8411	0.8521	0.8597
	FedPSO	0.8184	0.8416	0.8612	0.8746
	FedSCA	0.8454	0.8695	0.8796	0.8867
	FedGWO	0.8435	0.8729	0.8857	0.8908
	FedLO	0.8617	0.8825	0.8932	0.9027
WaRP	FedAvg, C = 1.0	0.7748	0.8025	0.8151	0.8229
	FedPSO	0.7851	0.8071	0.8207	0.8308
	FedSCA	0.7805	0.8060	0.8227	0.8354
	FedGWO	0.7900	0.8144	0.8320	0.8402
	FedLO	0.8233	0.8530	0.8655	0.8710

While the proposed FedLO framework shows promising results, it is essential to acknowledge certain limitations that should be addressed in future research:

- **Scalability Challenges:** The scalability of FedLO may face challenges in extremely-large-scale federated environments. Exploring strategies to optimize the algorithm's performance as the number of edge devices increases is crucial for broader applicability.
- **Sensitivity to Hyperparameters:** The performance of FedLO may be sensitive to the choice of hyperparameters.
- **Dependency on Network Conditions:** FedLO's performance relies on stable communication networks between the central server and edge devices. Adverse network conditions or high latency may impact the algorithm's effectiveness. Investigating techniques to mitigate the impact of unreliable networks is essential.
- **Model Heterogeneity:** The current version of FedLO assumes homogeneity among edge devices, considering them equal participants. Addressing the challenges posed by heterogeneous models on edge devices, such as varying architectures or capabilities, is an important avenue for future research.
- **Security Concerns:** FL, in general, raises security and privacy concerns, and FedLO is not exempt. Future work should include robust security mechanisms to safeguard sensitive data during the FL process.
- **Real-World Implementation Challenges:** The deployment of FL frameworks, including FedLO, in real-world scenarios may encounter practical challenges such as device dropout, varying data distributions, or diverse data types. Addressing these challenges is crucial for successfully adopting FL in practical applications.

In addition to the mentioned restrictions, it is crucial to consider the extent of the FedLO framework in contrast to conventional methods. Traditional approaches frequently depend on centralized data processing, which may only sometimes be practical or desirable in certain circumstances due to concerns regarding privacy, data locality, or regulatory restrictions. FedLO, which utilizes FL, presents a decentralized approach that permits model training to occur locally on edge devices, thereby mitigating privacy risks linked to centralizing data. Additionally, FedLO facilitates collaborative model training across distributed edge devices while safeguarding data privacy, making it well-suited for applications where data cannot be conveniently centralized.

## 6. Conclusions

This paper addressed the critical challenges associated with ML for sustainable development and introduced a novel communication-efficient FL framework, FedLO, designed to enhance scalability and sustainability in resource-constrained environments. The pressing need for sustainable solutions necessitated innovative data-driven tools, and FL emerged as a transformative paradigm by decentralizing ML training to edge devices. However, communication bottlenecks hindered its scalability. The proposed FedLO framework leverages the power of the LO, a nature-inspired metaheuristic algorithm, to strategically select relevant model updates, significantly reducing communication overhead. The key contributions of this work include the introduction of the FedLO framework, which achieved a significant communication cost reduction compared to existing methods while maintaining high model accuracy. Through rigorous evaluation of diverse datasets representing various areas of sustainable development, FedLO demonstrated its superiority over other FL approaches. The experiments showcased FedLO's exceptional accuracy, scalability, and sustainability performance, making it a promising solution for real-world sustainable development initiatives. The study delved into the dynamics of model training using FedLO on the rice leaf disease and waste recycling plant datasets, providing insights into the algorithm's effectiveness in progressively refining model performance over multiple training epochs.

Moreover, the robustness of FedLO was demonstrated in an unstable network environment, where random data drops were introduced. FedLO consistently outperformed other FL algorithms, maintaining higher accuracy under varying failure rates. However,

FedLO has limitations and areas where its performance can be enhanced. These include its adaptability to different FL scenarios, optimization efficiency, and robustness in unstable network environments.

To address these limitations and enhance FedLO's performance, several avenues for future research are proposed:

- **Algorithmic Modifications for LO:** Investigating and implementing modifications to the LO algorithm could be explored to improve its efficiency and adaptability to various FL scenarios. Fine-tuning the parameters or introducing novel strategies may lead to further performance gains.
- **Hybridization with Other Optimization Algorithms:** Exploring hybrid approaches by combining LO with other optimization algorithms could offer a synergistic effect, leveraging the strengths of different algorithms. Hybridizing LO with popular optimization techniques might enhance the overall performance and robustness of the FedLO framework.
- **Dynamic Adaptation of Communication Strategies:** Developing dynamic communication strategies within FedLO that adapt to the changing network conditions could be crucial. Investigating methods to dynamically adjust the communication frequency or prioritize specific updates based on network stability can improve the framework's resilience in fluctuating environments.

**Author Contributions:** Conceptualization, M.A.A.-B. and A.K.A.; methodology, M.A.A.-B., Z.A.A.A. and A.K.A.; software, M.A.A.-B. and A.K.A.; validation, M.A.A.-B. and A.K.A.; investigation, M.A.A.-B.; resources, M.A.A.-B., S.F. and R.F.M.; writing—original draft, M.A.A.-B., A.K.A., Z.A.A.A., S.F. and R.F.M.; Visualization, M.A.A.-B., A.K.A., Z.A.A.A., S.F. and R.F.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by grant from the Deanship of Graduate Studies and Research (DGSR) at Ajman University, Ajman, UAE (Grant No. 2023-IRG-ENIT-15).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data are contained within the article.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Abulibdeh, A.; Zaidan, E.; Abulibdeh, R. Navigating the confluence of artificial intelligence and education for sustainable development in the era of industry 4.0: Challenges, opportunities, and ethical dimensions. *J. Clean. Prod.* **2024**, *437*, 140527. [[CrossRef](#)]
2. Abasi, A.K.; Makhadmeh, S.N.; Alomari, O.A.; Tubishat, M.; Mohammed, H.J. Enhancing Rice Leaf Disease Classification: A Customized Convolutional Neural Network Approach. *Sustainability* **2023**, *15*, 15039. [[CrossRef](#)]
3. Malik, M.; Sharma, S.; Uddin, M.; Chen, C.L.; Wu, C.M.; Soni, P.; Chaudhary, S. Waste Classification for Sustainable Development Using Image Recognition with Deep Learning Neural Network Models. *Sustainability* **2022**, *14*, 7222. [[CrossRef](#)]
4. Zaferani, S.P.G.; Amiri, M.K.; Emami, M.R.S.; Zahmatkesh, S.; Hajiaghahi-Keshteli, M.; Panchal, H. Prediction and optimization of sustainable fuel cells behavior using artificial intelligence algorithms. *Int. J. Hydrogen Energy* **2024**, *52*, 746–766. [[CrossRef](#)]
5. Singh, R.; Biradar, C.; Behera, M.; Prakash, A.; Das, P.; Mohanta, M.; Krishna, G.; Dogra, A.; Dhyani, S.; Rizvi, J. Optimising carbon fixation through agroforestry: Estimation of aboveground biomass using multi-sensor data synergy and machine learning. *Ecol. Inform.* **2024**, *79*, 102408. [[CrossRef](#)]
6. da Silva, N.C.; Albertini, M.K.; Backes, A.R.; das Graças Pena, G. Machine learning for hospital readmission prediction in pediatric population. *Comput. Methods Programs Biomed.* **2024**, *244*, 107980. [[CrossRef](#)] [[PubMed](#)]
7. Beltrán, E.T.M.; Gómez, Á.L.P.; Feng, C.; Sánchez, P.M.S.; Bernal, S.L.; Bovet, G.; Pérez, M.G.; Pérez, G.M.; Celdrán, A.H. Fedstellar: A platform for decentralized federated learning. *Expert Syst. Appl.* **2024**, *242*, 122861. [[CrossRef](#)]
8. Bao, G.; Guo, P. Federated learning in cloud-edge collaborative architecture: Key technologies, applications and challenges. *J. Cloud Comput.* **2022**, *11*, 94. [[CrossRef](#)] [[PubMed](#)]
9. Juan, P.H.; Wu, J.L. Enhancing Communication Efficiency and Training Time Uniformity in Federated Learning through Multi-Branch Networks and the Oort Algorithm. *Algorithms* **2024**, *17*, 52. [[CrossRef](#)]

10. Asad, M.; Shaukat, S.; Hu, D.; Wang, Z.; Javanmardi, E.; Nakazato, J.; Tsukada, M. Limitations and Future Aspects of Communication Costs in Federated Learning: A Survey. *Sensors* **2023**, *23*, 7358. [[CrossRef](#)]
11. Al Amin, A.; Hong, J.; Bui, V.H.; Su, W. Emerging 6G/B6G wireless communication for the power infrastructure in smart cities: Innovations, challenges, and future perspectives. *Algorithms* **2023**, *16*, 474. [[CrossRef](#)]
12. Park, S.; Suh, Y.; Lee, J. FedPSO: Federated learning using particle swarm optimization to reduce communication costs. *Sensors* **2021**, *21*, 600. [[CrossRef](#)] [[PubMed](#)]
13. Nassef, A.M.; Abdelkareem, M.A.; Maghrabie, H.M.; Baroutaji, A. Review of Metaheuristic Optimization Algorithms for Power Systems Problems. *Sustainability* **2023**, *15*, 9434. [[CrossRef](#)]
14. Połap, D.; Woźniak, M. Meta-heuristic as manager in federated learning approaches for image processing purposes. *Appl. Soft Comput.* **2021**, *113*, 107872. [[CrossRef](#)]
15. Awadallah, M.A.; Al-Betar, M.A.; Doush, I.A.; Makhadmeh, S.N.; Alyasseri, Z.A.A.; Abasi, A.K.; Alomari, O.A. CCSA: Cellular Crow Search Algorithm with topological neighborhood shapes for optimization. *Expert Syst. Appl.* **2022**, *194*, 116431. [[CrossRef](#)]
16. Shokouhifar, M.; Hasanvand, M.; Moharamkhani, E.; Werner, F. Ensemble Heuristic–Metaheuristic Feature Fusion Learning for Heart Disease Diagnosis Using Tabular Data. *Algorithms* **2024**, *17*, 34. [[CrossRef](#)]
17. Alyasiri, O.M.; Cheah, Y.N.; Abasi, A.K.; Al-Janabi, O.M. Wrapper and hybrid feature selection methods using metaheuristic algorithms for English text classification: A systematic review. *IEEE Access* **2022**, *10*, 39833–39852. [[CrossRef](#)]
18. Rajwar, K.; Deep, K.; Das, S. An exhaustive review of the metaheuristic algorithms for search and optimization: Taxonomy, applications, and open challenges. *Artif. Intell. Rev.* **2023**, *56*, 13187–13257. [[CrossRef](#)]
19. Morales-Castañeda, B.; Zaldivar, D.; Cuevas, E.; Fausto, F.; Rodríguez, A. A better balance in metaheuristic algorithms: Does it exist? *Swarm Evol. Comput.* **2020**, *54*, 100671. [[CrossRef](#)]
20. Etaati, B.; Neshat, M.; Dehkordi, A.A.; Pargoo, N.S.; El-Abd, M.; Sadollah, A.; Gandomi, A.H. Shape and sizing optimisation of space truss structures using a new cooperative coevolutionary-based algorithm. *Results Eng.* **2024**, *21*, 101859. [[CrossRef](#)]
21. Velpula, P.; Pamula, R. EBGO: An optimal load balancing algorithm, a solution for existing tribulation to balance the load efficiently on cloud servers. *Multimed. Tools Appl.* **2022**, *81*, 34653–34675. [[CrossRef](#)]
22. Sulaiman, A.T.; Bello-Salau, H.; Onumanyi, A.J.; Mu’azu, M.B.; Adedokun, E.A.; Salawudeen, A.T.; Adekale, A.D. A Particle Swarm and Smell Agent-Based Hybrid Algorithm for Enhanced Optimization. *Algorithms* **2024**, *17*, 53. [[CrossRef](#)]
23. Abasi, A.K.; Aloqaily, M.; Guizani, M.; Karray, F. Sine cosine algorithm for reducing communication costs of federated learning. In Proceedings of the 2022 IEEE International Mediterranean Conference on Communications and Networking (MeditCom), Athens, Greece, 5–8 September 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 55–60.
24. Khan, A.A.; Hussain, S.; Chandra, R. A Quantum-Inspired Predator–Prey Algorithm for Real-Parameter Optimization. *Algorithms* **2024**, *17*, 33. [[CrossRef](#)]
25. Abasi, A.K.; Aloqaily, M.; Guizani, M. Grey Wolf Optimizer for Reducing Communication Cost of Federated Learning. In Proceedings of the GLOBECOM 2022–2022 IEEE Global Communications Conference, Rio de Janeiro, Brazil, 4–8 December 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1049–1154.
26. Abasi, A.K.; Makhadmeh, S.N.; Al-Betar, M.A.; Alomari, O.A.; Awadallah, M.A.; Alyasseri, Z.A.A.; Doush, I.A.; Elnagar, A.; Alkhamash, E.H.; Hadjouni, M. Lemurs optimizer: A new metaheuristic algorithm for global optimization. *Appl. Sci.* **2022**, *12*, 10057. [[CrossRef](#)]
27. Ra’ed, M.; Al-qudah, N.E.A.; Jawarneh, M.S.; Al-Khateeb, A. A novel improved lemurs optimization algorithm for feature selection problems. *J. King Saud Univ.-Comput. Inf. Sci.* **2023**, *35*, 101704.
28. kumar Ravikanti, D.; Saravanan, S. EEGAlzheimer’sNet: Development of transformer-based attention long short term memory network for detecting Alzheimer disease using EEG signal. *Biomed. Signal Process. Control* **2023**, *86*, 105318. [[CrossRef](#)]
29. Bharanidharan, N.; Chakravarthy, S.S.; Rajaguru, H.; Kumar, V.V.; Mahesh, T.; Guluwadi, S. Multiclass Paddy Disease Detection Using Filter Based Feature Transformation Technique. *IEEE Access* **2023**, *11*, 109477–109487. [[CrossRef](#)]
30. Admass, W.S.; Bogale, G.A. Arrhythmia classification using ECG signal: A meta-heuristic improvement of optimal weighted feature integration and attention-based hybrid deep learning model. *Biomed. Signal Process. Control* **2024**, *87*, 105565. [[CrossRef](#)]
31. Li, J.; Liu, A.; Han, G.; Cao, S.; Wang, F.; Wang, X. FedRDR: Federated Reinforcement Distillation-Based Routing Algorithm in UAV-Assisted Networks for Communication Infrastructure Failures. *Drones* **2024**, *8*, 49. [[CrossRef](#)]
32. Wu, C.; He, S.; Yin, Z.; Guo, C. Towards Client Selection in Satellite Federated Learning. *Appl. Sci.* **2024**, *14*, 1286. [[CrossRef](#)]
33. 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. Tutor.* **2020**, *22*, 2031–2063. [[CrossRef](#)]
34. Tang, C.; Wei, X.; Zhu, C.; Wang, Y.; Jia, W. Mobile vehicles as fog nodes for latency optimization in smart cities. *IEEE Trans. Veh. Technol.* **2020**, *69*, 9364–9375. [[CrossRef](#)]
35. Tufail, S.; Riggs, H.; Tariq, M.; Sarwat, A.I. Advancements and Challenges in Machine Learning: A Comprehensive Review of Models, Libraries, Applications, and Algorithms. *Electronics* **2023**, *12*, 1789. [[CrossRef](#)]
36. 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 Int. Things J.* **2020**, *8*, 5476–5497. [[CrossRef](#)]
37. Chen, Y.; Sun, X.; Jin, Y. Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *31*, 4229–4238. [[CrossRef](#)] [[PubMed](#)]

38. Miao, Z.; Yong, P.; Mei, Y.; Quanjun, Y.; Xu, X. A discrete PSO-based static load balancing algorithm for distributed simulations in a cloud environment. *Future Gener. Comput. Syst.* **2021**, *115*, 497–516. [[CrossRef](#)]
39. Zhenying, Y. Towards optimal neighbour selection mechanisms in peer-to-peer networks. *J. Opt.* **2023**, 1–10. [[CrossRef](#)]
40. Babu, B.S.; Venkatanarayana, M. AQWT: Adaptive quaternion wavelet transform and hybrid heuristic improvement for multi-modal medical image fusion model. *Signal Image Video Process.* **2023**, *18*, 1041–1051. [[CrossRef](#)]
41. Li, G.; Wu, H.; Yang, H. A multi-factor combination prediction model of carbon emissions based on improved CEEMDAN. *Environ. Sci. Pollut. Res.* **2024**, *31*, 20898–20924. [[CrossRef](#)]
42. Wajgi, R.; Brammya, G.; Ramesh, G.; Deotare, V.V.; Govindasamy, C.; Preetha, N.N. A novel enhanced lemurs optimization-based hybrid attention network for classifying arrhythmia from ECG signal using optimal weighted feature. *Aust. J. Electr. Electron. Eng.* **2024**, 1–25. [[CrossRef](#)]
43. Lalitha, K.S.; Sri, K.S.; Soundarya, G.; Koushik, K.; Sai, M.H. Lemurs Algorithm for Optimal Location and Sizing of Distributed Generation in Electrical Distribution Network. In Proceedings of the 2024 Third International Conference on Power, Control and Computing Technologies (ICPC2T), Raipur, India, 18–20 January 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 1–6.
44. Krizhevsky, A. *Learning Multiple Layers of Features from Tiny Images*; Technical Report; University of Toronto: Toronto, ON, Canada, 2009.
45. Deng, L. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Process. Mag.* **2012**, *29*, 141–142. [[CrossRef](#)]
46. Yudin, D.; Zakharenko, N.; Smetanin, A.; Filonov, R.; Kichik, M.; Kuznetsov, V.; Larichev, D.; Gudov, E.; Budenny, S.; Panov, A. Hierarchical waste detection with weakly supervised segmentation in images from recycling plants. *Eng. Appl. Artif. Intell.* **2024**, *128*, 107542. [[CrossRef](#)]
47. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 20–22 April 2017; PMLR: London, UK, 2017; pp. 1273–1282.

**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.