International Journal of Data Informatics and Intelligent Computing (IJDIIC)
Vol. 3, No. 2, June 2024, pp. 16~26
DOI: 10.59461/ijdiic.v3i2.105

Asynchronous Federated Learning with Grey Wolf Optimization for the Heterogeneity IoT Devices

Bambang Hari Kusumo[1], Prabhdeep Singh[2]
[1]Department of Soil Science, Mataram University, Mataram, Indonesia
[2]Department of Computer Science and Engineering, BBD University, Lucknow, Uttar Pradesh, India

Article Info

Article history:

Received February 24, 2024
Revised April 09, 2024
Accepted April 20, 2024

Keywords:

Machine Learning
Heterogeneity IoT devices
Light-weight node selection
Federated Learning

ABSTRACT

The Internet of Things (IoT) creates new options for real-time data collection and Machine Learning (ML) model development. Nonetheless, it's feasible that a particular IoT device does not have sufficient computational power to train and implement a complete learning model. However, there are significant communication costs and data security and privacy concerns associated with sending actual data to a centralised server with a lot of computational power. Federated Learning (FL) is a potential way to train ML models using low-powered devices and Edge Servers (ES) since it is a distributed ML architecture. However, the vast majority of the works in existence make the unsustainable assumption of a synchronized parameters update manner with similar IoT nodes and reliable communications networks. To increase training efficiency and accelerate the speed for heterogeneous IoT devices in an unreliable network environment, we designed an Asynchronous Federated Learning strategy with Grey Wolf Optimization (AFL-GWO) in this research. In particular, we develop a Lightweight Node Selection (LNS) technique and propose an AFL-GWO model to efficiently complete learning tasks. To ensure that diverse IoT nodes with varying computational capabilities and network connectivity are represented in the global learning aggregate, the proposed technique makes node selections on an iterative basis. We show through extensive experiments that our suggested AFL-GWO system outperforms the state-of-the-art techniques on identically and independently distributed (IID) and non-IID data distribution in a variety of contexts.

Corresponding Author: Prabhdeep Singh (e-mail: prabhdeepcs@gmail.com)

1. INTRODUCTION

The condition is being advanced in a variety of fields, including driverless driving, natural language processing, deep packet inspection, phishing mail filtration, banking system research, object tracking, and many others, due to ML methods [1]. The ability to gather and use widely dispersed actual information is substantially improving as a result of recent IoT advancements. However, it is still often impractical for a single IoT device to do the complete ML training phase by itself. Whereas traditional ML techniques call for a centralised, computation-intensive training program, data from many IoT devices cannot be transferred to a remote cloud server because doing so could cause the network to become overloaded and result in unconscionable latency. IoT and ML both face urgent data security issues whose flaws have not yet been completely identified and patched [2][3]. Personalized health data, for instance, are crucial for enhancing medical diagnoses and forecasting illness risk. This information is very sensitive and intimate and may result in catastrophic harm to people if made public. By moving processing workloads from the centralised cloud to networking edges, the computation concept may help to reduce capacity problems. It makes it possible for IoT devices to use their local data to engage in computational processes. IoT nodes may gather and analyse data, connect with these other networks, and enable a range of distributed intelligent systems when seen as a system architecture [4]. The systems under investigation in this article are separated into three levels,

comprising a distant cloud, a collection of computation offloading nodes, and heterogeneity IoT devices, as illustrated in Figure 1. Although the distant cloud has a lot of processing power, it is far from nearby smaller stations. In comparison, ES engages with IoT devices at the channel's edge to minimise the cloud's computational burden and the delay in the communications of IoT devices [5].

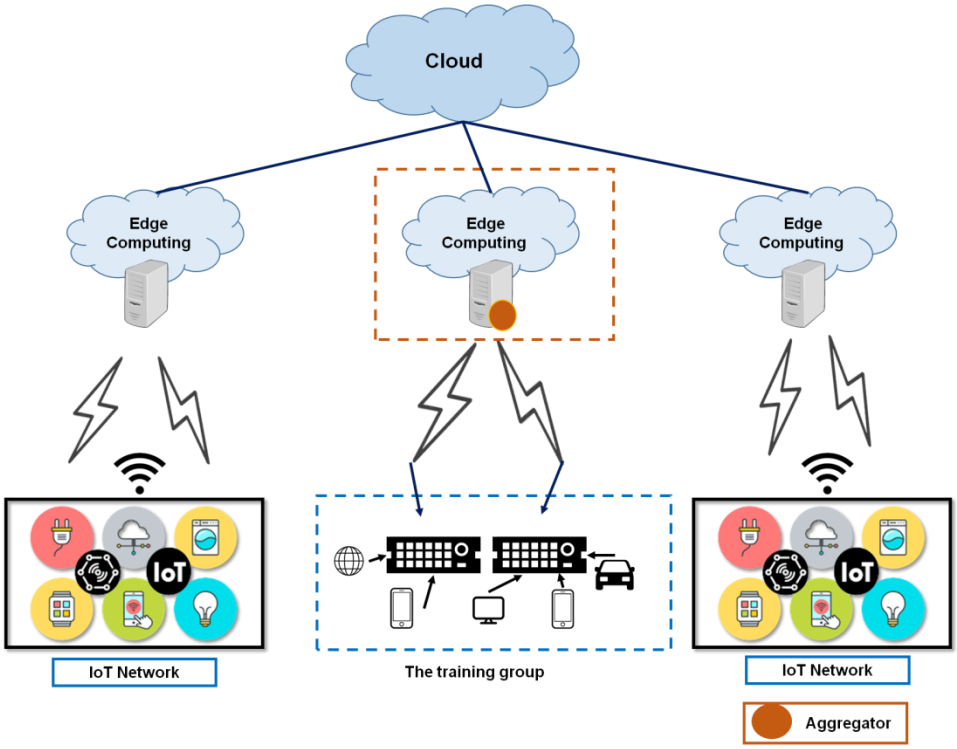

Figure 1. IoT system design for FL that is enabled by a variety of edges

Additionally, ES' computational capabilities may help analyse the data gathered from IoT devices. A wide range of approaches have been used to study cloud technology, including simultaneous radio and computational distribution of resources, dumping of processing, and management of multi-access edge supercomputers [6]. In order to cooperatively train an ML model, FL, as a decentralized education model, makes use of local computing resources and data in dispersed devices. Aggregation sites are placed at ES as part of the federated education process, interacting with different nodes as localized trainers to train the very same ML model utilising location information. The aggregation gathers the weighted inputs from nearby students and averages them for each round. Then, using the data set, the aggregators evaluate the learning efficiency of the algorithm [7]. We have provided the following contributions to the article to address this difficulty.

- We execute investigations on the MNIST 10-digit dataset and the Fashion-MNIST dataset to verify the effectiveness of our technique.
- We build an Asynchronous Federated Learning - Grey Wolf Optimization ((AFL-GWO) to boost training efficiency and speed for heterogeneous IoT devices in an unstable network.
- We design Lightweight Node Selection (LNS) to efficiently choose nodes.

The remainder of this article is as follows. Relevant literatures are discussed in Part 2. The proposed technique is outlined in Part 3. The efficacy of our approach is examined in Part 4. The study's conclusions is presented in Part 5.

2. LITERATURE REVIEW

A basic analysis of a structure that employs blockchain to manage diverse IoT devices is presented in the paper [8]. They begin by highlighting the shortcomings of earlier IoT systems and the challenges associated with fusing IoT with blockchain. Then, the authors provide a management architecture for a highly heterogeneity IoT system. The management and communication technologies are automatically configured in

the study [9] using reward learning models in a dynamic industrial context. In order for supervised learning to converge quickly, they produce three public plans on the features of industrial facilities. In order to perform extended experiments, they construct and incorporate the reinforcing learning-based founder technique on a robust wirelessly cyber-physical simulation.

The authors of the study [10] concentrate on partly visible data poisoning attacks in crowdsensing systems and demonstrate that even when hostile employees only have access to their own information, they may still use TruthFinder to develop efficient data poisoning attack tactics. They formalise the issue of partly visible data poisoning attacks against crowdsensing platforms first, and then researchers provide a deep reinforcement learning-based data poisoning attack technique that enables hostile employees to compromise TruthFinder whilst remaining undetected. The study [11] suggests deploying an IoT access platform close to the wearable sensors at the network edge. The design may make IoT applications faster and more reactive and effectively safeguard sensitive data from prying eyes. In order to provide IoT with cloud-based applications with a uniform picture of heterogeneous wearable sensors, they suggest a generic ontology-based object-relational paradigm of IoT devices. To increase the effectiveness of dispersed education, an outstanding FL strategy for the heterogeneity IoT-edge FL system is put forward in the article [12]. To address the problem of poor training effectiveness brought on by the diversity of clients, they first investigate an Iterative Self Organizing Data Analysis Techniques Algorithm (ISODATA)-based server and client schedule method.

In the article [13], they investigate and design the Resource Provisioning and Workload Assignment ((RPWA) for IoT services problem as a mixed integer algorithm to jointly choose the quantity and position of ES and apps to deploy, in addition to the teaching load allocation. Given its difficulty, they suggest a deconstruction strategy to address it, which divides RPWA into two sub-problems: the products can lead to the mobility network edge and the delayed aware workload allocation sub-problem. They provide a unique Plug-and-Play (PnP) solution for the aforementioned issue in the article [14]. SensPnP, the suggested PnP solution, is a set of integrated software and hardware that can connect third-party embedded sensors to IoT devices without any previous knowledge of the sensors or the Internet. They outline an IoT device architecture that supports several integrated peripherals communication systems and is PnP-enabled. Cross-technology communication (CTC) approaches may be broadly categorized into two types: equipment-dependent and hardware-free.

The research [15] aims to give a comprehensive state-of-the-art review of CTC from the hardware viewpoint. In hardware-based solutions, specialised hardware is needed to convey information to wireless devices to allow direct connection. On the contrary, hardware-free approaches allow heterogeneity-connected technologies to interact directly by sharing information and data without the need for special hardware. To allow blockchain in IoT, they describe Tornado in the paper [16], a slightly elevated blockchain system built on a spatial register and related algorithms. A unique consensus technique called cooperative work is designed to handle the massive heterogeneity of IoT. To increase the resource productivity of IoT devices, they also suggest the Space-structured Greedy Heaviest-Observed Subtree (S^2GHOST) method. A Proof of Concept (PoC) in a fictitious Software-Defined Networks (SDN) environment is shown in the paper [17] using the mathematical model. The suggested architecture greatly reduces variability, which aids in maintaining Quality-of-Service (QoS) and enhancing security, according to performance assessment findings.

In the paper [18], they investigate the issue of job rescheduling onto such a heterogeneity Multi Processor System on a Chip (MPSoC) deployed in the IoT for maximising security quality while taking into account limits on fuel, authenticity, and project priority. In order to maximise the security system, they first provide a Mixed Integer Linear Programming (MILP) framework for assigning and scheduling dependent activities with energy and real-time restrictions on a heterogeneity MPSoC system. In order to determine the real-time resemblance of continuous monitoring of the microtubules aspect of the machines, the paper [19] presents a Heterogeneity Industrial Internet of Things (HetIoT) architecture. This idea is supposed to be shown by the examination of equipment usefulness, economy, productivity, and so forth. The suggested Advanced Machine-metameric Dimension (AmD) with HetIoT has achieved greater clarity, F1 measure, recollection, and accuracy. Researchers have created and built middleware based on a service-oriented architecture that can handle heterogeneous difficulties in the work [20]. The solution was created in three stages, beginning with the use of REST API to gather data from a variety of heterogeneity sensor devices, followed by the introduction of heterogeneity connectivity protocols, and lastly, middleware testing on gateways running various operating systems.

3. METHOD

In this paper, we propose an AFL-GWO approach, where a collection of IoT devices train a model using only the data collected at the edge. To boost the convergence of the learning process, the suggested asynchronous node selection scheme takes benefit of the current state of network and computing interaction. Figure 2 depicts the overview of suggested method.

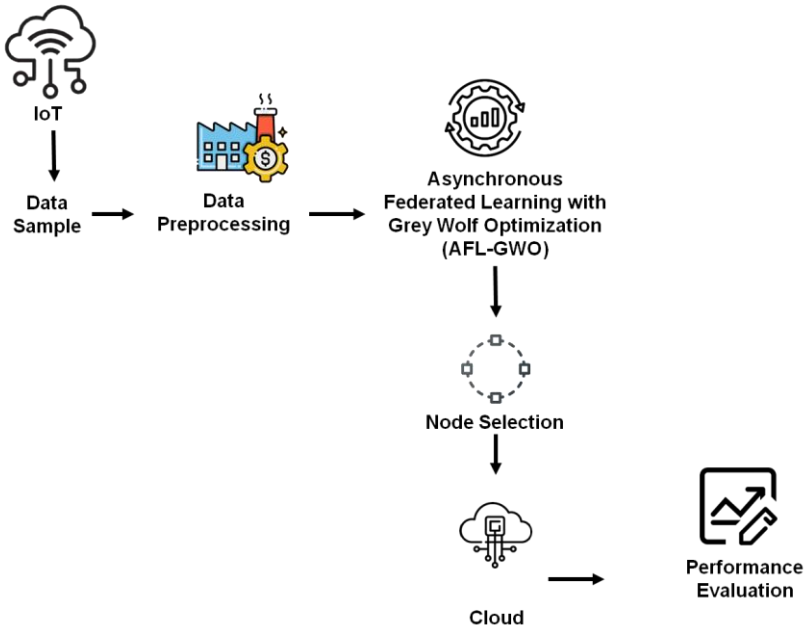

Figure 2. Overview of the suggested method

3.1. Data pre-processing

The gathered data is pre-processed using Z-score normalization. Z-score normalisation analysis of raw data uses its mean and standard deviation to get a normalised score. Equation (1) demonstrates that the z-score variable may be used to standardise the unstructured data.

$$I'_n = \frac{I_n - \bar{D}}{std(D)} \quad (1)$$

Where,
I'_n Shows the standardised Z-score values
Z-scores have been normalised and shown in I'_n.
I_n Identifies row D of the lth column where the value occurs.

$$std(D) = \sqrt{\frac{1}{(O-1)} \sum_{n=1}^{n} (I_n - \bar{D})^2} \quad (2)$$

$\bar{D} = \frac{1}{n} \sum_{n=1}^{n} I_n$ or mean value (3)

Each row may make use of the Z-score method since all the values are the same, yielding a standard deviation of 0 when the values are all set to 0 to produce standard data. Similar to Min-Max normalisation, the z-score may be used to establish a scale from 0 to 1.

3.2. Conceptual framework

This section provides a comprehensive overview of our methodology. In particular, we begin with an overview of the structure and our explanation for the structure. Following this, we present our AFL-GWO proposal. Finally, we detail the asynchronous technique we've suggested for selecting nodes for dispersed computing in an IoT system.

3.3. Problem domain and architecture explanation

The ML industry may face a number of difficulties, including those related to decentralised and centralised ML, as shown in Figure 3. The problems and demands for IoT systems cannot be met by

conventional ML approaches, which collect all training data at a centralised server with considerable computational power. We believe that FL offers a potential way to connect ML with IoT systems. To begin with, IoT devices might gather data samples and run the training process locally. Second, the public learner may save their information publically to lessen security risks and conserve expensive wireless bandwidth. The network structure of IoT systems and the FL framework are also quite similar.

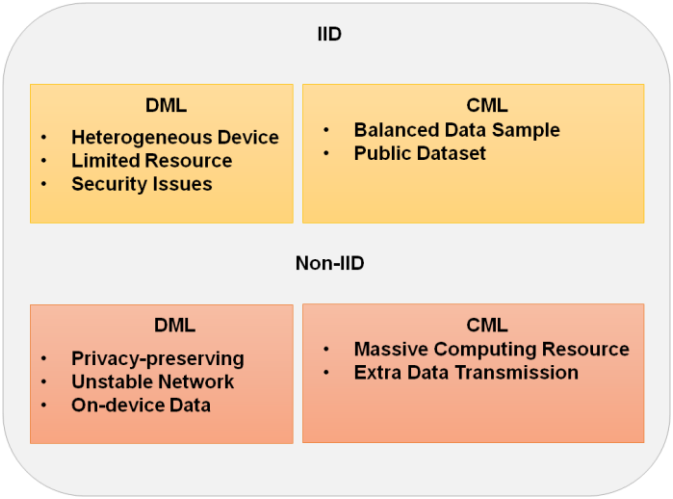

Figure 3. Challenges in the ML sector

As can be seen in Figure 3, the FL system is implemented on edge computing nodes linked by an IoT network. In this scenario, several IoT devices work together to train a single learning model, with the edge node serving as the centralised parameter server (or aggregator). While most data samples may not adhere to the IID, IoT devices may still collect original data from their environment of usage. A distributed learning system, FL makes use of distributed computational capabilities across several nodes. Heterogeneity nodes can all train identical ML model using their own data. Still, the vast majority of current FL initiatives assume a homogenous system where every node has identical processing power, trains synchronously, and has stable connectivity. Despite this, synchronous learning is inefficient in the real world since IoT devices are heterogeneity. Waiting for the weakest nodes to update their weight vector (WV) is an inefficient use of time due to the varied computational power and dynamic connectivity circumstances of IoT devices. To solve this problem, we develop an AFL-GWO model, where computational resources are collected, and nodes are dynamically chosen for global aggregation in order to boost the performance of the designated learning task.

3.4. FL model

The scenario under study in FL consists of several aggregators and IoT devices. These nodes in training are all responsible for training the same ML model. The paper's most important notations are tabulated in Table 1.

Table 1. Description of notation

Notations	Description
$\boldsymbol{M}$	*The number of devices in the network*
$\boldsymbol{J}$	*The set of nodes*
$\boldsymbol{U}$	*The weight vector of the machine learning model*
$\boldsymbol{L}()$	*The loss function*
$\boldsymbol{\mathcal{T}_m}$	*The dataset on node m*
$\boldsymbol{T_m}$	*The number of data samples on node m*
$\boldsymbol{B}$	*The selected nodes in global aggregation*
$\boldsymbol{D}$	*The local training period*
$\boldsymbol{\lambda_m}$	*The number of local updates on the node m*
$\boldsymbol{b_m}$	*Communication rate on node m*

We refer to IoT nodes in the set $J = \{1,2,\ldots.,m,\ldots..,M\}$ and$T_m = \{d_1, d_2, \ldots d_i, \ldots. d_{T_m}$ as the data source on particular node m, where $T_m$indicates the total response rate among all local dataset T_m`. The initial goal was FL to find a WV $U_S \in \Re^r$ that minimises wide loss function $L_S()$ across the training session. Here, r indicates the dimensions the W . FL presents a learning challenge that may be expressed as

$$U_S^* = argmin\ L_S(U_S) \tag{4}$$

The creation of a local update is then described. To be more accurate, during each iteration, a certain number of IoT networks are selected to participate in the globe aggregate. The local loss function $L_m()$ is minimised across each node's individual dataset T_m by training the obtained global WV U_S^{d-1} with $A_m$epochs. At the conclusion of the iteration, every node has a local WV U_S^d. The local update δ_m^d is then calculated locally and transmitted back to the central parameter server as follows:

$$\delta_m^d = U_m^d - U_S^{d-1} \tag{5}$$

All obtained local updates are aggregated using the weight-averaging approach on the centralized parameter server, which is stated as follows:

$$U_S^d = U_S^{d-1} + \sum_{m=1}^{b^d} \frac{T_m}{T_{b^d}} \delta_m^d \tag{6}$$

It should be noted that B^d ($B^d \subseteq J$ is a group of node that shows the nodes with adaptively contribute to the total aggregated during round d. The case of synchronous updating is covered by the stated issue (Equations (4)–(6)). The updating process is synchronous, that is, $B^d = J$, while every node inside the collection of nodes is J become a part of the global aggregate.

The issue of accurately calculating the amount of time and space required for FL activities remains unsettled. Computing expenses during training is thus an important topic to cover initially. We assume the ML model's computational complexity is constant across all iterations.

The compute contribution $\lambda_m$on to the ML task for a given number of local updates A_m is thought to be linked to the minibatch size employed in those updates. It is anticipated that the parameters servers (i.e., aggregator) will set and maintain the same minibatch size (hyperparameter) throughout the FL process. By multiplying A_m^d (the total number of updates) by t_g (the minibatch size), we can calculate λ_m^d. Then, we have

$$\lambda_m^d = A_m^{d*} t_g \tag{7}$$

For the learning task, it is important to note that the size of the mini-batch is determined empirically. $N_D = \{\lambda_1, \lambda_2, \ldots, \lambda_n, \ldots, \lambda_M\}$ is the collection of local computing contributions across the FL process. Furthermore, we think of the set MT as an accumulation. If node m was chosen to take part in the last global aggregate, its value of λ_m will be reset to 0. For this purpose, we have

$$\lambda_m = \begin{cases} \lambda_m + \lambda_m^d, G_m^d = 1 \\ 0, \qquad G_m^d = 0 \end{cases} \tag{8}$$

Where $G_m^d = 1$ if node m is included in the global aggregate and $G_m^d = 0$ otherwise. The speed of WV transfer among nodes and the aggregators serves as a representation of the connectivity costs for aggregating at each round, i.e., $P = \{p_1^d, p_2^d, \ldots. p_m^d, \ldots. p_M^d\}$. This allows us to simulate the flexible communications system in the IoT context.

3.5. AFL scheme

The AFL system, which comprises the following three phases, is fully detailed. In step 1, the initialization process takes place. After the initialization aggregation process takes place in step 2, the parameters are updated.

Step 1: The server, sometimes referred to as the parameter aggregator, creates the random initial parameter vector $U_{initial}$ and the global ML model. It creates node set J. After that, the aggregators deliver the original WV $U_{initial}$ to node set J.

Step 2: The ACK frame that each node in set J provides to the aggregators includes an evaluation of the current computational contribution (λ_m) and communications status (p_M^d). Then, a suggested node selection

technique is shown in the section that follows. It uses the weight averaging approach to choose nodes for set B^d to obtain the global aggregate at each iteration d.

3.6. Node selection

We construct a step 2-based algorithm to choose which nodes should take part in the aggregate at the specified time. D_{tf}, where D_{tf} is a set period of time utilised to gather node update data. To enhance the performance of the ML task, the suggested method seeks to collect as much computational contribution as possible with the available communication time (D_{tf}). Consequently, the formulation of the node selection issue at iteration t is

$$\underset{G_m^d, m \in J}{max} \sum_{m=1}^{M} \lambda_m G_m^d \tag{9}$$

$$s.t \begin{cases} \sum_{m=1}^{M} k_m^d G_m^d \leq D_{tf}, (m = 1,2, \dots, M) \\ G_m^d \in \{0,1\}, \quad (m = 1,2, \dots, M) \end{cases} \tag{10}$$

Due to the diversity of IoT devices, we calculate the transmission time$k_m^d = \frac{J}{p_m^d}$, where f represents the packet size of the WV, based on the communication conditions in set B.

The $0-1$ $Knapsack\ problem$ represented by equations (9) and (10) is well-known. The suggested method must handle a high-density (IoT) ecosystem. A heuristic algorithm built on the greedy approach is developed to estimate the solution to the $0-1$ $Knapsack\ problem$, finding a compromise between the ideal answer and the efficiency of the operations.

Algorithm 1. The aggregate node classification approach (in stage 2)

Initialize: $B^d = \{\}, O - \{\}, J^d = J, n = 0, D_{tf}, N_D, D$;
Verify the most recent communication situation P;
Evaluate the efficiency of the contributions calculation $o_n^d = \lambda_m * p_m^d$;
$D_{tf}^d = D_{tf}$;
while $J^d \neq \{\emptyset\}$ do
    $\arg\max(O)\ \forall m \in J^d$;
    $n = J_m^d$;
    Remove$\in J_m^d$ from J^d;
    if $k_m^d \leq D_{tf}^d$ then
        $D_{tf}^d = D_{tf}^d - k_m^d$;
        $B^d = B^d \cup n$
End while;

To accelerate the training speed for heterogeneity IoT devices in an unreliable network environment, we use GWO.

The GWO is a new optimization method with biological underpinnings. The GWO approach uses a group of search agents to identify the optimal answer to a problem. The GWO algorithm stands out from the crowd because of the social dominance hierarchy that generates the candidate solution at every optimization iteration. The hunting mechanism consists of three stages: monitoring, approaching, and striking the prey. To solve difficult optimization issues, grey wolves employ a mathematical hunting strategy known as GWO. As a consequence, a victim is thought to be the greatest way to solve an issue.

According to the following formula, the victim is being surrounded by GW when they perform the three higher level movements.

$$\vec{G} = \overrightarrow{F}.\overrightarrow{A_s(w)} - \vec{A}(w) \tag{11}$$

The vector A_s denotes where the prey is located, the vector A denotes where the GW is, and the vector F denotes the coefficient. The following equation is employed to relocate a given element closer to or farther from the region containing the ideal solution (representing the prey).

$$\overrightarrow{A(v+1)} = \left|\overrightarrow{A_t(v) - \vec{e}.\vec{f}}\right|, with\ \vec{e} = 2\vec{e}.\overrightarrow{n_1} - \vec{e} \tag{12}$$

Where n_1 is randomly selected from the range [0, 1] and an is decreased from 2 to 0 over a predetermined number of iterations. When$|D|$ exceeds 1, this mimics prey attack behaviour and is consistent with exploitation behaviour. When $|D| > 1$, the wolf's distance from the victim is copied. The suggested range for C is [-2, 2]. Three higher levels, $c, d, and\ e$, will be calculated using the mathematical formulas below.

$$\overrightarrow{G_c} = \left|\overrightarrow{F_1}.\overrightarrow{A_c} - \vec{A}\right| with\ \overrightarrow{A_1} = \overrightarrow{A_c} - \overrightarrow{A_c}.\left(\vec{G}_c\right) \quad (13)$$

$$\overrightarrow{G_d} = \left|\overrightarrow{F_2}.\overrightarrow{A_c} - \vec{A}\right| with\ \overrightarrow{A_2} = \overrightarrow{A_d} - \overrightarrow{A_d}.\left(\overrightarrow{G_d}\right). \quad (14)$$

$$\overrightarrow{G_e} = \left|\overrightarrow{F_3}.\overrightarrow{A_c} - \vec{A}\right| with\ \overrightarrow{A_3} = \overrightarrow{A_e} - \overrightarrow{A_e}.\left(\overrightarrow{G_e}\right) \quad (15)$$

Consider that b, c, and d have enough knowledge about the victim's expected location to statistically recreate the grey wolf's hunting strategy. Additionally, the top 3 solutions are kept, which forces the remaining agents to adjust their values to match those of the top 3 agents, b, c, and d.

For the purposes of argument, let's say that c, d, and e have enough intelligence to mathematically replicate the grey wolf's hunting technique. And the top three solutions are kept, so the other agents have to adjust their locations to match those of c, d, and e.

$$\vec{A}(w+1) = \frac{\overrightarrow{a_1}+\overrightarrow{a_2}+\overrightarrow{a_3}}{3} \quad (16)$$

4. RESULTS AND DISCUSSION

Using the "MNIST 10-digit and Fashion-MNIST" datasets, we conduct tests to verify the effectiveness of our proposed AFL-GWO. We compare our proposed method to a few modern techniques, such as Synchronous Federated Learning (SFL) [21] and Random Nodes Selection Federated Learning (RNS-FL) [22]. As we compare the two state-of-the-art methods in various contexts, we first outline the assessment approach before presenting the evaluation findings.

4.1. Accuracy

We compare the accuracy performance across the various strategies in Figure 4(a)(b). Accuracy rises as the number of nodes grows, and our suggested asynchronous strategy achieves the requisite accuracy more quickly (i.e., with less training time). This is due to the fact that the suggested node selection technique may optimally choose the IoT nodes with enough computational power in each cycle.

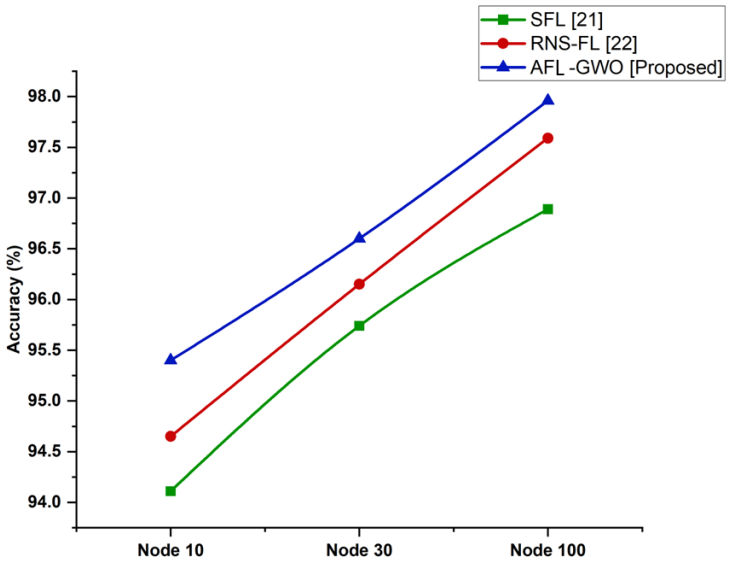

Figure 4(a). Accuracy for MNIST 10-digit

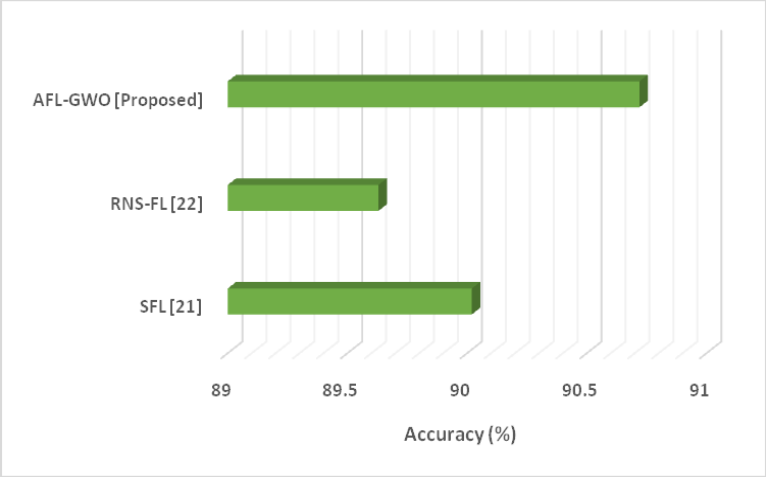

Figure 4(b). Accuracy for Fashion-MNIST

4.2. Computing Time

The amount of time needed to complete a computing operation is known as computation time. When a calculation is represented as a series of rule deployments, the computation time is inversely correlated with the quantity of rule implementations. The computation durations for our technique and other techniques are shown in Figure 5. It demonstrates that AFL-GWO requires less processing time than other approaches.

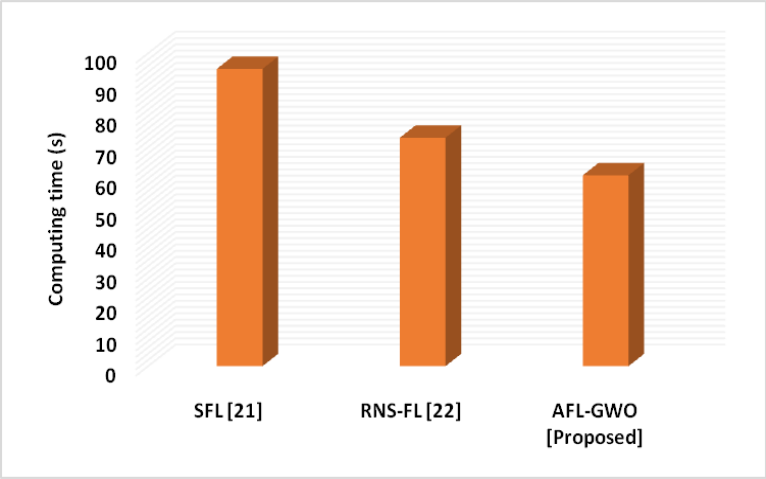

Figure 5. Comparison of computing time

4.3. Security Level

Security is a technique that protects IoT devices that are linked over a network by using protection mechanisms while also limiting cyber threats. The security level of our technique and other techniques is shown in Figure 6. It is evident that AFL-GWO exhibits a high level of security when compared to other strategies.

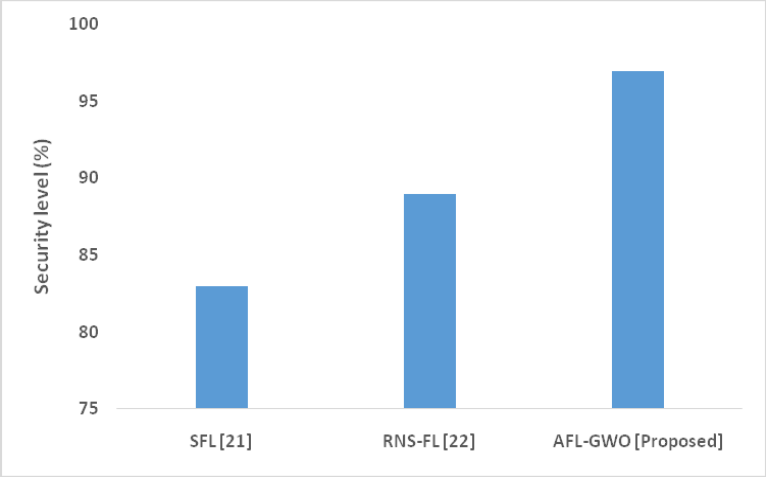

Figure 6. Comparison of security level

5. CONCLUSION

In this research, we have discussed the difficulties in implementing IoT systems driven by the edge spread over dynamic communications infrastructure. We've designed an Asynchronous Federated Learning with Grey Wolf Optimization (AFL-GWO) that can function well with various IoT devices. We have suggested a Lightweight Node Selection (LNS) approach to provide a rough answer with minimal computing cost for adaptively deciding which nodes participate every time there is a global aggregate. We have tested the effectiveness of the suggested scheme by a number of quantitative tests on big datasets, and we have furthermore verified clearly the two conventional state-of-the-art systems are not as effective as our suggested solution, taking into account both IID and Non-IID data types.

In the future, we will address a number of techniques for federated training communications reduction, including model compression and local updating. It's critical to comprehend how these methods interact each other and to rigorously examine how each method trades off communication for accuracy. The most effective strategies will, in particular, show advances at the Pareto frontier, reaching an accuracy larger than any other strategy while spending the same amount on communications, and, ideally, over a broad variety of interaction characteristics.

DATA AVAILABILITY STATEMENT

Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest in this work.

REFERENCES

[1] H. Wang, M. Daneshmand, and H. Fang, "Artificial Intelligence (AI) Driven Wireless Body Area Networks: Challenges and Directions," in *2019 IEEE International Conference on Industrial Internet (ICII)*, IEEE, Nov. 2019, pp. 428–429. doi: 10.1109/ICII.2019.00079.

[2] D. Wu, H. Shi, H. Wang, R. Wang, and H. Fang, "A Feature-Based Learning System for Internet of Things Applications," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1928–1937, Apr. 2019, doi: 10.1109/JIOT.2018.2884485.

[3] F. Liang, W. G. Hatcher, W. Liao, W. Gao, and W. Yu, "Machine Learning for Security and the Internet of Things: The Good, the Bad, and the Ugly," *IEEE Access*, vol. 7, pp. 158126–158147, 2019, doi: 10.1109/ACCESS.2019.2948912.

[4] H. Hu, D. Wang, and C. Wu, "Distributed Machine Learning through Heterogeneous Edge Systems," *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 05, pp. 7179–7186, Apr. 2020, doi: 10.1609/aaai.v34i05.6207.

[5] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1125–1142, Oct. 2017, doi: 10.1109/JIOT.2017.2683200.

[6] F. Liang, W. Yu, X. Liu, D. Griffith, and N. Golmie, "Toward Edge-Based Deep Learning in Industrial Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4329–4341, May 2020, doi: 10.1109/JIOT.2019.2963635.

[7] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated Learning: Challenges, Methods, and Future Directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020, doi: 10.1109/MSP.2020.2975749.

[8] H. Xu, X. Liu, W. Yu, D. Griffith, and N. Golmie, "Reinforcement Learning-Based Control and Networking Co-

Design for Industrial Internet of Things," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 5, pp. 885–898, May 2020, doi: 10.1109/JSAC.2020.2980909.
[9] M. Li, Y. Sun, H. Lu, S. Maharjan, and Z. Tian, "Deep Reinforcement Learning for Partially Observable Data Poisoning Attack in Crowdsensing Systems," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6266–6278, Jul. 2020, doi: 10.1109/JIOT.2019.2962914.
[10] L. Tseng, L. Wong, S. Otoum, M. Aloqaily, and J. Ben Othman, "Blockchain for Managing Heterogeneous Internet of Things: A Perspective Architecture," *IEEE Netw.*, vol. 34, no. 1, pp. 16–23, Jan. 2020, doi: 10.1109/MNET.001.1900103.
[11] L. Lan, R. Shi, B. Wang, and L. Zhang, "An IoT Unified Access Platform for Heterogeneity Sensing Devices Based on Edge Computing," *IEEE Access*, vol. 7, pp. 44199–44211, 2019, doi: 10.1109/ACCESS.2019.2908684.
[12] Y. Cui, K. Cao, G. Cao, M. Qiu, and T. Wei, "Client Scheduling and Resource Management for Efficient Training in Heterogeneous IoT-Edge Federated Learning," *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 41, no. 8, pp. 2407–2420, Aug. 2022, doi: 10.1109/TCAD.2021.3110743.
[13] N. Kherraf, H. A. Alameddine, S. Sharafeddine, C. M. Assi, and A. Ghrayeb, "Optimized Provisioning of Edge Computing Resources With Heterogeneous Workload in IoT Networks," *IEEE Trans. Netw. Serv. Manag.*, vol. 16, no. 2, pp. 459–474, Jun. 2019, doi: 10.1109/TNSM.2019.2894955.
[14] S. K. Roy, S. Misra, and N. S. Raghuwanshi, "SensPnP: Seamless Integration of Heterogeneous Sensors With IoT Devices," *IEEE Trans. Consum. Electron.*, vol. 65, no. 2, pp. 205–214, May 2019, doi: 10.1109/TCE.2019.2903351.
[15] Y. Chen, M. Li, P. Chen, and S. Xia, "Survey of cross-technology communication for IoT heterogeneous devices," *IET Commun.*, vol. 13, no. 12, pp. 1709–1720, Jul. 2019, doi: 10.1049/iet-com.2018.6069.
[16] R. Kumar and R. Sharma, "Leveraging blockchain for ensuring trust in IoT: A survey," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 10, pp. 8599–8622, Nov. 2022, doi: 10.1016/j.jksuci.2021.09.004.
[17] Y. Liu, K. Wang, K. Qian, M. Du, and S. Guo, "Tornado: Enabling Blockchain in Heterogeneous Internet of Things Through a Space-Structured Approach," *IEEE Internet Things J.*, vol. 7, no. 2, pp. 1273–1286, Feb. 2020, doi: 10.1109/JIOT.2019.2954128.
[18] H. Yang *et al.*, "FedRich: Towards efficient federated learning for heterogeneous clients using heuristic scheduling," *Inf. Sci. (Ny).*, vol. 645, p. 119360, Oct. 2023, doi: 10.1016/j.ins.2023.119360.
[19] M. A. Khan and K. A. Abuhasel, "Advanced metameric dimension framework for heterogeneous industrial Internet of things," *Comput. Intell.*, vol. 37, no. 3, pp. 1367–1387, Aug. 2021, doi: 10.1111/coin.12378.
[20] Y. Mesmoudi, M. Lamnaour, Y. El Khamlichi, A. Tahiri, A. Touhafi, and A. Braeken, "A Middleware based on Service Oriented Architecture for Heterogeneity Issues within the Internet of Things (MSOAH-IoT)," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 32, no. 10, pp. 1108–1116, Dec. 2020, doi: 10.1016/j.jksuci.2018.11.011.
[21] S. Wang *et al.*, "When Edge Meets Learning: Adaptive Control for Resource-Constrained Distributed Machine Learning," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, IEEE, Apr. 2018, pp. 63–71. doi: 10.1109/INFOCOM.2018.8486403.
[22] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," Feb. 2016, [Online]. Available: http://arxiv.org/abs/1602.05629

BIOGRAPHIES OF AUTHORS

Bambang Hari Kusumo, SH., M.Agr.Sc., Ph.D. Professor University of Mataram City Lombok Indonesia NTB. Specialist in Soil and Environmental Science. Result publication citation 536 and H-index 12. Many papers have been published in international journals. He can be contacted at email: media@unram.ac.id

Prabhdeep Singh received a B.Tech (Computer Science and Engineering) from Saroj Institute of Technology and Management, Lucknow, affiliated with U.P.T.U., U.P., India, M. Tech. (Computer Science) and Pursuing Ph.D. (Computer Science and Engineering) from Amity School of Engineering and Technology Lucknow, Amity University Uttar Pradesh, India. His research interests include cloud computing, data security, machine learning, data mining and warehousing, scheduling and optimization. He has published a number of research papers in journals of national and international reputation. He can be contacted at email: prabhdeepcs@gmail.com