2024 IEEE 29th Asia Pacific Conference on Communications (APCC) | 979-8-3315-1781-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/APCC62576.2024.10768063

Collaborative Decentralized Learning for Detecting Bearing Faults in Industrial Internet of Things

Made Adi Paramartha Putra*, Ahmad Zainudin‡, Gabriel Avelino Sampedro§, Nengah Widya Utami*
Dong-Seong Kim†, *Senior Member, IEEE*, and Jae-Min Lee†, *Member, IEEE*,
*Faculty of Information Technology and Design, Primakara University, Denpasar, Indonesia.
†Department of IT Convergence Engineering, Kumoh National Institute of Technology, Gumi, South Korea.
‡Department of Electronic Engineering, Kumoh National Institute of Technology, Gumi, South Korea.
§Faculty of Information and Communication Studies, University of the Philippines Open University, Laguna, Philippines
adi@primakara.ac.id, zai@kumoh.ac.kr, garsampedro@ieee.org, widya@primakara.ac.id, {dskim, ljmpaul}@kumoh.ac.kr

***Abstract*—An essential aspect of Industrial Internet of Things (IIoT) systems lies in their reliability and resilience against failures. Fault detection serves as a crucial method for mitigating errors, leading to reduced downtime. Previous studies have predominantly focused on fault detection using centralized Artificial Intelligence (AI) approaches, wherein participant information is centralized and forwarded to a central server. However, Federated Learning (FL) offers a solution to these issues, enhancing the system's reliability. In this study, we propose a Decentralized FL (DFL) approach for collaborative learning in bearing fault detection. DFL is preferred over centralized FL due to its elimination of a single point of failure. By leveraging the decentralized FL concept, the vulnerability of the collaborative framework to attacks can be minimized. Our proposed DFL integrates continual learning techniques to reduce communication overhead. The results demonstrate that decentralized collaborative learning achieves satisfactory performance, with an accuracy rate of 96.08% and a learning time reduction of up to 37.52%.**

***Index Terms*—Decentralized federated learning; bearing fault detection; industrial iot;**

I. Introduction

The adoption of Industrial Internet of Things (IIoT) networks has been steadily increasing in recent years [1]. IIoT technology is utilized to monitor and control industrial and manufacturing processes continuously [2]. The capabilities of IIoT have been enhanced with the integration of Artificial Intelligence (AI) algorithms capable of determining machine faults and taking corrective actions to support fully automated industrial processes [3]. By leveraging AI, the efficiency, reliability, and robustness of the IIoT environment are significantly improved. In the traditional approach, the AI learning process adopts centralized learning, which is ineffective as it involves forwarding data to a central server and incurring substantial communication overhead. This mechanism is impractical for IIoT applications, which require low latency and efficient model updates, as centralized learning does not meet these requirements effectively [4].

Federated Learning (FL) presents a novel method for conducting AI models in a decentralized manner, addressing the limitations of centralized learning in the IIoT environment. With FL, distributed training occurs on the client side, ensuring privacy and robust model creation without necessitating the sharing of client data [5]. Nevertheless, FL must consider elements such as device diversity, access control, security, and data distribution [6]. In our previous research, we introduced an accuracy-driven client selection technique that prioritizes clients based on centralized evaluation [7]. However, the centralized approach of FL, where the aggregation process is conducted by an FL server, is vulnerable to attacks. If the FL server is compromised, the FL process can be exploited by adversaries. Therefore, a mechanism to eliminate the FL server is recommended.

Removing the FL server means that the aggregation process must be performed by each participant in the FL process. Decentralized FL (DFL) is particularly useful in this regard, as it allows participants to collect models from others and conduct local aggregation processes, which can then be shared within the same FL process. Several works related to fault detection have been conducted using the FL method. For instance, adaptive control for fault detection is discussed in [8]. Additionally, Chain FL, which utilizes blockchain technology to assign the task of storing the model to the nodes within the network instead of relying on a centralized server, is discussed in [9].

Most previous studies in fault detection for IIoT have focused on centralized AI or centralized FL and have overlooked the possibility of attacks on the FL server. Additionally, the applicability of DFL has not been fully explored. In this study, we utilize the continual DFL method for fault detection in IIoT networks. The major contributions of this work can be summarized as follows:

1) We propose a continual DFL which removes the central server role as an aggregator for the industrial IoT scenario, specifically for bearing fault detection.
2) We introduce a threshold time for the continual DFL algorithm. This parameter can be used to strictly control the learning process time, thus enhancing efficiency.
3) To investigate the performance of the proposed method, various evaluations were conducted. First, a comparison between centralized and decentralized FL, followed by the evaluation of increasing the total number of participants in the continual DFL process.

The structure of this work is detailed as follows: In Section II, recent work on fault detection mechanisms is covered, followed by an examination of research gaps. The proposed continual DFL is described in Section III, including the algorithm, dataset, and configuration for the simulation. Section IV provides a detailed comparative analysis with state-of-the-art work under various conditions. Finally, the conclusion and future work of this article are provided in

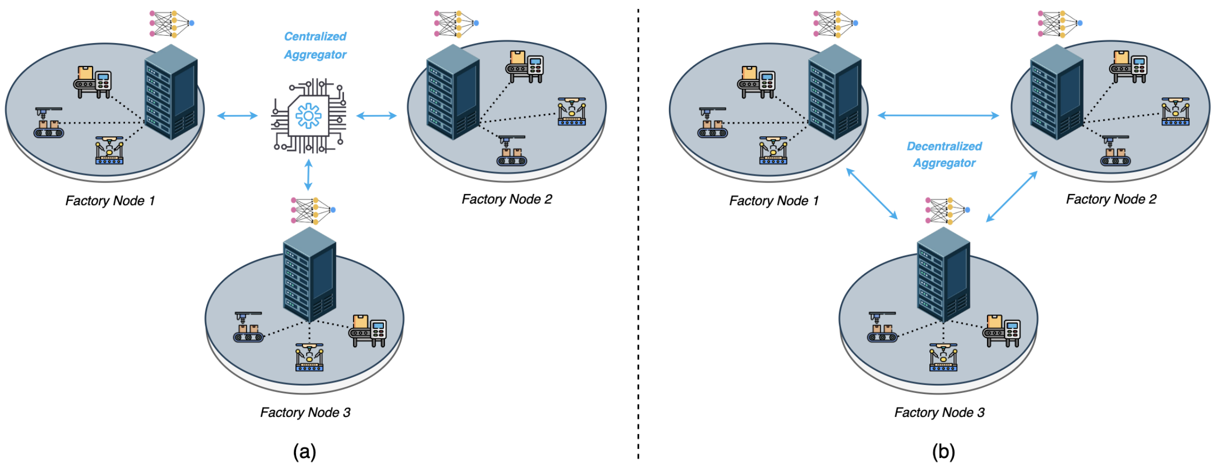


Fig. 1. Two different learning processes of the FL approach. (a) Centralized FL; (b) Decentralized FL.

Section V.

II. Related Works

Referring to bearing fault detection problems in the IIoT environment is widely discussed in academia. However, most state-of-the-art techniques mainly focus on centralized FL approaches. For instance, Huang et al. presented the first FL-based approach for fault detection, which varied the total number of participants involved in the federation process. The authors claim that the designed method is able to achieve high accuracy [10]. Another method, presented by Chen et al., discusses a technique for detecting bearing fault diagnosis using FL and combines it with dynamic weighted averaging to address imbalanced data distributions [11]. Performance comparisons using different aggregation strategies are investigated in [12]. The authors compare the performance of FedAvg and FedProx algorithms under various data distributions. Additionally, an aggregation strategy based on precision difference is also evaluated and shows promising results.

Not only is centralized FL used, but the DFL method is also utilized for detecting bearing faults in industrial machinery. The author in [8] introduced a mechanism to adaptively control fault detection by using an approximation model. This model approximates the upper bound of the fault, ensuring quick recovery to stability. Another work, presented by Korkmaz et al., introduces the integration of blockchain technology with the FL approach. In this model, updates are stored in the blockchain, and the aggregation process is conducted via the blockchain network instead of a centralized FL server [9].

III. Proposed System

A. System Model

Concerning the FL approach, two distinct methods govern how the federation works to securely train the model without disclosing actual information. Fig. 1 illustrates these two methods, namely centralized and decentralized FL (DFL). In centralized FL, an additional entity, known as the FL server, manages aggregation, client selection, and other mechanisms. The second approach, decentralized FL, eliminates the central aggregator due to the risk of a single point of failure resulting from attacks or other factors. The aggregation process is conducted individually by each participant in the federation. In certain scenarios, the implementation of decentralized FL is considered superior to centralized techniques.

In this work, we focused on DFL techniques and deliberately ignored the existence of a central FL server traditionally used as an aggregator. For the IIoT network, reliability and robustness are mandatory. Therefore, relying on a central server, which is prone to attack, represents a very dangerous scenario. The proposed DFL method is depicted in Fig. 2. Here, 'the factory' refers to the IIoT nodes that collect sensory data and store it on the edge server. After data collection, the edge server not only stores the data but also participates in the DFL process. It is worth noting that the edge server plays a crucial role, acting as the aggregator, as detailed previously.

Let the factories be grouped into n groups denoted by *F(N)* = 1, 2, 3, ..., n, and each factory relies on the others to perform the DFL mechanism. The total number of aggregation processes for each communication round can be given by n. This is because each factory is required to perform the aggregation before the other participants are able to continue the learning process.

B. Proposed Continual DFL Algorithm

Based on the previous description, the DFL method proposed in this work utilizes the continual approach. This technique is selected over the aggregate method to reduce communication overhead, which is crucial in the IIoT network. The larger the factories that participate in the DFL process, the longer it takes to finish the learning process, which is also linear with the total number of communication rounds.

In Fig. 2, the learning process of the proposed DFL is initialized by factory node 1. Then, the updated local model is used by factory node 2 to train the model based on their dataset. Training of the local model can be denoted as follows:

$$\omega_l^n \leftarrow \omega_g - \eta \nabla \ell \left(\omega_g; b\right) . \tag{1}$$

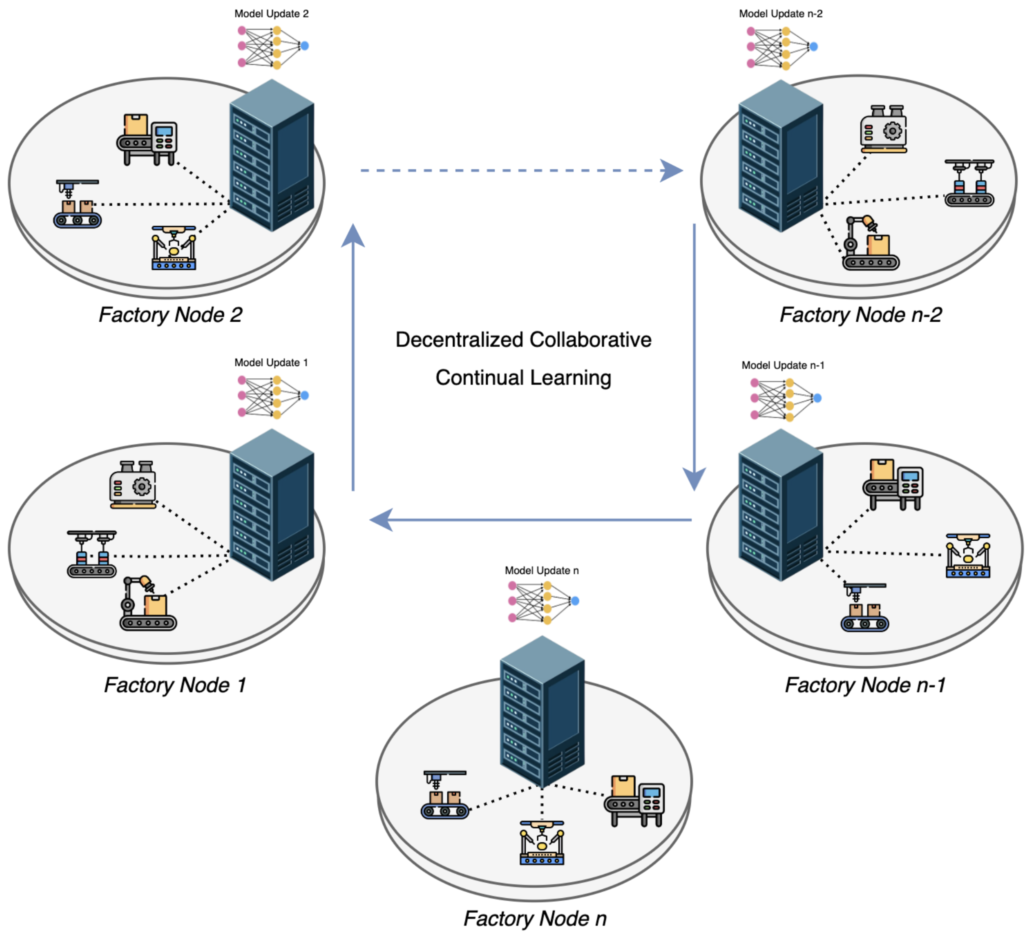


Fig. 2. The algorithmic flow of the proposed continual decentralized FL in the Industrial IoT network.

In this context, the updated local model ω_l^n originating from factory n is derived using the global model ω_g and the outcomes of training on a batch b of local data β, utilizing a learning rate η.

After the training is completed, factory node 2 starts the aggregation by combining the parameters from factory node 1 with the trained parameters to form a new aggregated parameter for the next factory training. The aggregation process can be performed by using the following equation:

$$\omega_{g_{t+1}} \leftarrow \sum_{n=1}^{N} \frac{\eta\, n}{\eta} \omega_{l_{t+1}}^{n} \,. \tag{2}$$

This process occurs iteratively until all available factories train their own datasets under the DFL communication rounds. In most state-of-the-art scenarios, even if some factories fail to deliver results, the DFL process still continues. In this work, we introduce a threshold time, denoted by τ. This period is used to determine whether the subsequent factory is able to deliver the new model update or not. If, during the period, no model update is stored, then the next factory will continue the process and ignore the previous factory's model update. By utilizing this technique, the time constraint for the training process using DFL can be strictly guaranteed.

To better understand the process of the proposed continual DFL, the pseudocode of the method is detailed in Algorithm 1. First, the initialization of the global model is mandatory, followed by the initialization of several aspects such as the total number of participants, communication rounds, batch size, epoch size, learning rate, as well as the threshold time period. After all initializations, the iterative process is performed. Firstly, under the circumstances of each communication round, followed by the order of the factory. For the sake of simplicity, the order of continual DFL in this work is conducted in an ascending manner. After a specific iteration of communication rounds is finished, the model is evaluated to collect the final model performance after the learning process.

C. Dataset Description and Simulation Details

The performance of the proposed system was assessed with an industrial IoT-related dataset, specifically bearing fault detection data obtained from the Case Western Reserve University (CWRU) [13]. This dataset consists of data gathered from a machinery motor equipped with a torque transducer, dynamometer, and control electronics, recorded at two distinct sampling rates: 12,000 and 48,000 samples per second. The collected data saved in the .mat file format and MATLAB was used for preprocessing. In this work, we focused on a shaft rotation speed of 1,772 rpm, with accelerometers operating at a sampling frequency of 48 kHz.

In total, 50 rounds of communication are used to conduct the federation process for the fault detection model. It is worth mentioning that the proposed DFL method is evaluated

Algorithm 1: Proposed Continual DFL Algorithm

Initialization of global model parameter: ω_g
Initialization of other parameters: $N,\ t,\ b,\ \eta,\ e,\ \tau$

for each communication round t **do**
    for each $n \in F(N)$ in ascending **do**
        $IndividualTraining(\omega_g,\ t)$
    end for
    $\omega_n \leftarrow$ Collect parameter from previous participant
    $\omega_{g_{t+1}} \leftarrow$ aggregate new global model via equation (2)
    $Acc, F_1 \leftarrow globalmodel(\omega_{g_{t+1}}).evaluate()$
end for

Function $IndividualTraining(\omega_g,\ t)$
$\omega_l \leftarrow$ Collect ω_g from previous participant
for epoch 1 to e **do**
    for batchsize $b \in B$ **do**
        $\omega_l \leftarrow$ update local model via equation (1)
    end for
end for
$\omega_{l_{t+1}} \leftarrow$ aggregate new global model via equation (2)
Store aggregated model $\omega_{l_{t+1}}$, t for subsequent participant

TABLE I
THE SIMULATION PARAMETER CONFIGURATIONS FOR THE PROPOSED CONTINUAL DFL.

Parameter	Symbol	Value
Communication Round	t	50
Total Factory	K	10-50
Batch Size	b	32
Local Epochs	e	1
Threshold time (s)	τ	300
Aggregation Technique	n/a	FedAvg
Model Configuration	n/a	CNN & MLP

using two different models: Convolutional Neural Network (CNN) and Multilayer Perceptron (MLP). These models were selected due to their capabilities to handle numerical data for data extraction in case of classification and detection problems. The complete parameter configuration for the continual DFL model is detailed in Table I.

IV. Performance Evaluation

For the evaluation process, this work conducts a comparison utilizing two distinct models: MLP [14] and CNNMLP, which combines MLP and CNN [15]. Initially, the comparison is performed between the centralized and decentralized approaches using MLP model. Figure 3 illustrates the performance comparison of the two approaches investigated in this work. It can be observed that the performance of centralized and decentralized FL is similar. In our simulation, we observe that the performance of CFL is slightly higher in the first three communication rounds, whereas for subsequent federations, DFL achieves better model accuracy.

For completeness, we further investigated the impact of the total number of factories participating in the learning

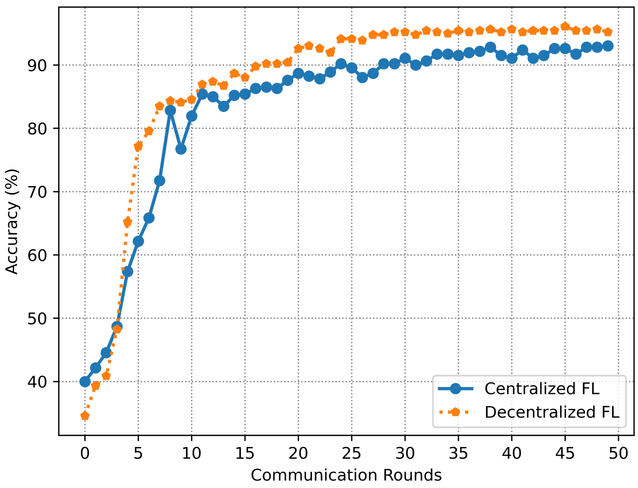


Fig. 3. The performance comparison between centralized FL and decentralized FL techniques using MLP model.

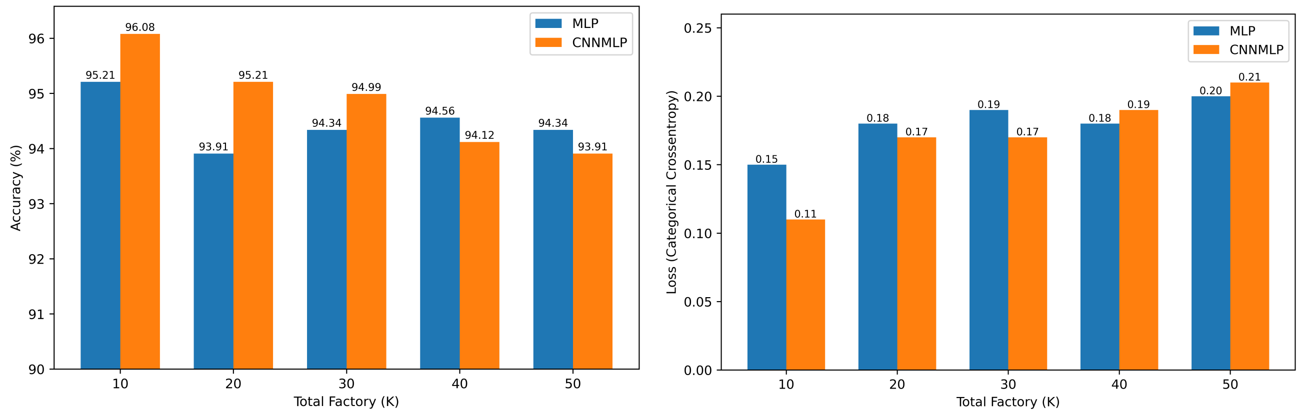
TABLE II
TIME COMPARISON BETWEEN TRADITIONAL METHODS AND THE PROPOSED DFL.

Method	Average Round Time	Total Training Time
Traditional DFL	27.109 s	1355.450 s
Proposed DFL	**16.938 s**	**846.931 s**

process. Five different numbers were explored, starting from $10, 20, 30, 40$, and 50. Based on the results depicted in Fig. 4(a), the performance of the MLP and CNNMLP models is comparable. For total participants fewer than 30, the CNNMLP model achieves higher accuracy. For instance, with a 10-factory configuration, MLP achieves an accuracy of 95.21%, while CNNMLP delivers 96.08%. However, for a total number of participants equal to 50, the total accuracy produced by the MLP is higher than CNNMLP. The MLP model is able to achieve 94.34% accuracy, whereas CNNMLP obtains an accuracy of 93.91

Additionally, the losses of the model are also calculated for each configuration during the evaluation based on the categorical cross-entropy approach detailed in Fig. 4(b). The results indicate the same trends as the accuracy presented in Fig. 4(a). Based on the two metrics evaluated in this work, we find that the size of the participant is important in determining the final model accuracy. If the scenario requires a large number of participants, it is recommended to use the MLP model compared to CNNMLP.

Finally, to demonstrate the advancement of the proposed continual DFL, the evaluation focused on learning time. The scenario was designed to mimic device failure and showcase the effectiveness of the time threshold in this work. In total, 10 nodes were tested with 50 communication rounds, with the results detailed in Table II. The findings show the efficiency of the proposed DFL in terms of average round time and total training time, with a 37.52% improvement. This is supported by the threshold time, which ignores malfunctioning nodes in the DFL system.



(a) Accuracy

(b) Loss

Fig. 4. The proposed continual DFL performance among two different approaches is investigated in this work with a variation in the total number of factories and 50 communication rounds.

V. Conclusion and Future Work

The article outlines a technique for continual DFL. The proposed method is able to perform continuous learning without the need for any assistance from a central server. It is worth noting that the proposed continual DFL also includes a threshold time to mitigate any node failures during the federation process. The evaluation was conducted on an industrial-related dataset, specifically the bearing fault dataset. The findings show that the proposed continual DFL can deliver significant performance when compared with centralized FL. The proposed DFL can also be paired with any model to deliver adequate results. In this work, we investigate the MLP and CNNMLP models, which show satisfactory results with up to 96.08% accuracy and a 0.11 loss. Additionally, our main contribution is to limit the learning process time, which was successfully achieved with a time threshold, resulting in a total reduction of 37.52%.

Future research in DFL holds several promising avenues to explore. For example, investigating dataset distribution among various participants, appropriately selecting the order of continual aggregation processes, optimizing hyperparameters for each participant locally, considering hierarchical DFL [16], and integrating blockchain-based approaches [17] such as Ethereum-based network or zero-knowledge proof (ZKP) hold compelling potential.

Acknowledgment

This work was partly supported by Innovative Human Resource Development for Local Intellectualization program through the Institute of IITP grant funded by the Korea government(MSIT) (IITP-2024-RS-2020-II201612, 50%) and by the IITP(Institute of Information & Coummunications Technology Planning & Evaluation)-ICAN(ICT Challenge and Advanced Network of HRD) grant funded by the Korea government(Ministry of Science and ICT)(IITP-2024-2022-00156394*, 50%)

References

[1] M. A. P. Putra, A. P. Hermawan, D.-S. Kim, and J.-M. Lee, "Data prediction-based energy-efficient architecture for industrial iot," *IEEE Sensors Journal*, vol. 23, no. 14, pp. 15 856–15 866, 2023.

[2] G. A. Sampedro, M. A. P. Putra, J.-M. Lee, and D.-S. Kim, "Industrial internet of things-based fault mitigation for smart additive manufacturing using multi-flow bilstm," *IEEE Access*, vol. 11, pp. 99 130–99 142, 2023.

[3] G. Kim, J. G. Choi, M. Ku, H. Cho, and S. Lim, "A multimodal deep learning-based fault detection model for a plastic injection molding process," *IEEE Access*, vol. 9, pp. 132 455–132 467, 2021.

[4] S. R. Saufi, Z. A. B. Ahmad, M. S. Leong, and M. H. Lim, "Challenges and opportunities of deep learning models for machinery fault detection and diagnosis: A review," *IEEE Access*, vol. 7, pp. 122 644–122 662, 2019.

[5] M. Aledhari, R. Razzak, R. M. Parizi, and F. Saeed, "Federated learning: A survey on enabling technologies, protocols, and applications," *IEEE Access*, vol. 8, pp. 140 699–140 725, 2020.

[6] L. Li, Y. Fan, and K.-Y. Lin, "A survey on federated learning," in *2020 IEEE 16th International Conference on Control & Automation (ICCA)*, 2020, pp. 791–796.

[7] M. A. P. Putra, A. R. Putri, A. Zainudin, D.-S. Kim, and J.-M. Lee, "Acs: Accuracy-based client selection mechanism for federated industrial iot," *Internet of Things*, vol. 21, p. 100657, 2023.

[8] S. Zhou, J. Bai, and F. Wu, "Decentralized fault detection and fault-tolerant control for nonlinear interconnected systems," *Processes*, vol. 9, no. 4, 2021.

[9] C. Korkmaz, H. E. Kocas, A. Uysal, A. Masry, O. Ozkasap, and B. Akgun, "Chain fl: Decentralized federated machine learning via blockchain," in *2020 Second International Conference on Blockchain Computing and Applications (BCCA)*, 2020, pp. 140–146.

[10] G.-Y. Huang and C.-H. Lee, "Federated learning architecture for bearing fault diagnosis," in *2021 International Conference on System Science and Engineering (ICSSE)*, 2021, pp. 408–411.

[11] J. Chen, J. Li, R. Huang, K. Yue, Z. Chen, and W. Li, "Federated learning for bearing fault diagnosis with dynamic weighted averaging," in *2021 International Conference on Sensing, Measurement & Data Analytics in the era of Artificial Intelligence (ICSMD)*, 2021, pp. 1–6.

[12] D. Geng, H. He, X. Lan, and C. Liu, "Bearing fault diagnosis based on improved federated learning algorithm," *Computing*, vol. 104, no. 1, pp. 1–19, 2022. [Online]. Available: https://doi.org/10.1007/s00607-021-01019-4

[13] X. Chen, B. Zhang, and D. Gao, "Bearing fault diagnosis base on multi-scale CNN and LSTM model," *Journal of Intelligent Manufacturing*, vol. 32, no. 4, pp. 971–987, 2021. [Online]. Available: https://doi.org/10.1007/s10845-020-01600-2

[14] G.-Y. Huang and C.-H. Lee, "Federated learning architecture for bearing fault diagnosis," in *2021 International Conference on System Science and Engineering (ICSSE)*, 2021, pp. 408–411.

[15] W. Zhang, X. Li, H. Ma, Z. Luo, and X. Li, "Federated learning for machinery fault diagnosis with dynamic validation and self-supervision," *Knowledge-Based Systems*, vol. 213, p. 106679, 2021.

[16] M. A. P. Putra, S. M. Rachmawati, M. Abisado, and G. A. Sampedro, "Hftl: Hierarchical federated transfer learning for secure and efficient fault classification in additive manufacturing," *IEEE Access*, vol. 11, pp. 54 795–54 807, 2023.

[17] A. Zainudin, M. A. P. Putra, R. N. Alief, R. Akter, D.-S. Kim, and J.-M. Lee, "Blockchain-inspired collaborative cyber-attacks detection for securing metaverse," *IEEE Internet of Things Journal*, pp. 1–1, 2024.