IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 18 May 2023, accepted 24 May 2023, date of publication 29 May 2023, date of current version 7 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3280471

RESEARCH ARTICLE

HFTL: Hierarchical Federated Transfer Learning for Secure and Efficient Fault Classification in Additive Manufacturing

MADE ADI PARAMARTHA PUTRA[1], SYIFA MALIAH RACHMAWATI[2], MIDETH ABISADO[3], (Member, IEEE), AND GABRIEL AVELINO SAMPEDRO[4,5], (Member, IEEE)

[1]Informatic Engineering, STMIK Primakara, Denpasar, Bali 80226, Indonesia
[2]Research and Development Center, Philippine Coding Camp, Manila 0922, Philippines
[3]College of Computing and Information Technologies, National University, Manila 1008, Philippines
[4]Faculty of Information and Communication Studies, University of the Philippines Open University, Laguna 4301, Philippines
[5]Center for Computational Imaging and Visual Innovations, De La Salle University, Malate, Manila 1004, Philippines

Corresponding author: Gabriel Avelino Sampedro (garsampedro@ieee.org)

ABSTRACT The technology advancement is supported by additive manufacturing industries, especially 3D printing companies, that enable fast object prototyping and development in Industry 4.0. As 3D printed products are highly adopted in various fields, the final printed product must fulfill precise requirements without any defects. Therefore, an efficient framework that simultaneously learns and detects faults during the printing process is required. Unfortunately, most state-of-the-art studies utilize a centralized approach, which is inefficient for continuous model updates. This article presents a hierarchical federated transfer learning (HFTL) framework that employs edge, fog, and cloud concepts to enable an efficient model updating mechanism. The proposed HFTL uses a well-known DL model with a new classifier to significantly reduce the distributed training process while providing high detection and classification performance. Additionally, the fog server comprehensively exploits the data collection from several edge servers and performs local training. The extensive simulation results indicate that the performance of the proposed HFTL is more efficient, with 24% faster training time, effectively detects flaws in 3D printing products with 45% accuracy, and has a 59% F1-score improvement in non-IID data distribution compared to traditional FL architecture.

INDEX TERMS Additive manufacturing, hierarchical network, federated learning, fault classification, transfer learning.

I. INTRODUCTION

The continuous development of technology in recent years has succeeded in connecting the industrial world with automation processes and bringing the industrial revolution 4.0 that enables the industrial internet of things (IIoT) [1], [2]. As a result, various machines are interconnected and communicate to boost productivity in the production process. This concept has successfully made changes in terms of effectiveness and efficiency for worldwide industries. For example, machine automation can handle massive production in the additive manufacturing industry within a short time. Additive manufacturing is one of the many fabrication processes capable of printing various types of products. For example, it is used to produce aerospace elements, accounting for 30% of components in aeroengines [3], [4]. Additionally, it finds applications in the manufacturing of radio-frequency components [5] and parts for unmanned aerial vehicles (UAVs) [6].

In order to produce an industrial-grade product, precise printing is needed in 3D printing for additive manufacturing. Therefore, the best 3D printer component and materials are mandatory to satisfy those conditions. For plastic 3D modelling, fused deposition modelling (FDM), stereolithography (SLA), and selective laser sintering (SLS) are the most popular printing techniques [7]. The FDM 3D printer's

The associate editor coordinating the review of this manuscript and approving it for publication was Mu-Yen Chen.

printing process relies on an extruder heated with a specific temperature value to melt the filament used as printing material. In addition, another aspect that is also important in FDM techniques is printing surface temperature and levelling, cooling fan and positioning motor. In general, the FDM 3D printing process rarely encounters problems since the printing path is carefully generated based on the actual object designed. However, problems such as filament not adhering to the printing surface, nozzle clogging in the extruder, or temperature fluctuation at bed and extruder might occur due to printer malfunction. These circumstances tend to construct faulty products and significantly influence the production cost.

As technology advances, several techniques, such as computer vision, machine learning (ML), and deep learning (DL), were used to mitigate 3D printing faults in the additive manufacturing industry. On the one hand, the author in [8] introduces vision-based error detection for 3D printing processes, especially FDM printers. The captured video in the form of a red-green-blue (RGB) colour model is transformed into a hue-saturation-value (HSV) and compared based on threshold values. In addition, the author also mentioned the impact of light, where image brightness might increase detection difficulty due to printing surface reflection. With a similar approach, the author in [9] presented a real-time computer vision system for automatic fault detection in FDM-based printers using the prototyping method. The proposed method that considers light intensity is sufficient to detect a failure in each layer of the printed 3D product.

On the other hand, ML and DL as emerging technologies were also utilized to detect 3D printer faults during the printing process. A supervised ML approach with a support vector machine (SVM) is used to detect FDM 3D printer status. SVM is able to differentiate printed products as good or bad as it is trained with labelled data [10]. A semi-supervised ML was introduced in [11] by investigating the laser power-bed fusion (LPBF) printer to improve the model quality without disregarding performance. The author uses labelled and unlabeled data to reduce the total cost and time to capture the labelled data. The ensemble ML approach is also used to enhance fault detection in the 3D printed surface with low computing cost by utilizing pre-trained models [12]. Finally, the layer-wise and density-wise approach is adopted to iteratively detect layer conditions during the printing process, which has been proven to improve fault detection performance.

Moreover, the feature extraction capabilities of DL are utilized to further enhance fault detection in 3D printers. A convolutional neural network (CNN)-based model is proposed in [13] to diagnose FDM 3D printer conditions based on accelerometer sensor readings. Six different classes were analyzed, and the CNN-based network can outperform the ML approaches. FDM 3D printer uses a nozzle to extract the filament into a printed product. In some cases, the nozzle might clog, preventing filament from going out and resulting in product failure. The author in [14] introduces nozzle anomaly detection using a multi-head encoder-decoder temporal neural network with high-performance accuracy. Moreover, a collaborative technique is introduced in [15] that combines computer vision with artificial intelligence (AI), specifically deep CNN, with real-world environment experiments. A machine-agnostic algorithm is employed in [16] to improve real-time anomaly detection and classification in 3D printers by utilizing a CNN-based network with pixel-wise localization.

Furthermore, most state-of-the-art studies in fault detection and classification for the 3D printer in additive manufacturing adhere to a centralized learning concept. This approach is remarkable for its high performance since the amount of training data is significant. However, in some scenarios, sharing data might threaten user privacy [17]. Especially for the industry use cases, where data is a crucial component of the business process. Despite the fact that a company is capable of providing enough data for detecting faults in their 3D printers, a larger size of the dataset could improve the model's robustness. Federated learning (FL) becomes a promising solution for conducting distributed learning while preserving user privacy [18]. However, the architecture of distributed learning is entirely different from centralized learning. For distributed learning, an independent server is placed to control the client selection mechanism for the training process and aggregate parameters from training results from those selected clients. Utilizing the FL approach, the training process is conducted locally by the client with a private dataset using a global model parameter from the server. After local training is completed, the client forwards the results parameter to the central server without sharing the private dataset. Therefore, the data privacy of clients is wholly secured. The aggregated parameter from several clients is used for the following FL training rounds. Those processes are repeated continuously to achieve higher fault detection and classification.

Considering the requirements for advanced fault detection additive manufacturing, especially for FDM 3D printers with privacy-preserving capability in the industrial network environment, this study proposes the following contributions:

- We present a distributed fault classification in additive manufacturing using FL. The proposed technique exploits an image dataset with four different classes, specifically captured and preprocessed with data augmentation to deliver a large-scale dataset.
- We employed a federated transfer learning algorithm to learn various features from clients with different data distributions in a short period and generalize the DL model. Moreover, we analyzed several pre-trained models, such as VGG16, ResNet50V2, and MobileNetv2, to explore the most efficient pre-trained model for additive manufacturing.
- We proposed the edge, fog, and cloud concept to create a hierarchical architecture in FL called hierarchical federated transfer learning (HFTL). The proposed HFTL assumes that each company maintains a fog server

connected to a cloud server and multiple factory branches with 3D printers. This technique will preserve the data privacy of all companies during the distributed learning process.
- We investigated the proposed model's robustness in the simulation by analyzing the fault classification performance with and without hierarchical architecture in FL using accuracy, loss, precision, recall, F1-score, and training time.

The remainder of the paper is structured as follows: Section II delivers the state-of-the-art of fault detection and classification techniques for the FDM 3D printer. The proposed architecture and pre-trained model are described in Section III. The experimental results and discussion are presented in Section IV. Finally, Section V concludes the study and provides future research trends.

II. RELATED WORK

Generally, fault classification for additive manufacturing uses three techniques: computer vision, ML, and DL approaches. However, most of these approaches utilize the centralized learning method, which is unsuitable for secure and efficient fault classification in the additive manufacturing industry. This section provides state-of-the-art approaches used for a 3D printer the fault classification.

A. COMPUTER VISION APPROACHES

In the computer vision techniques, 3D printer product is collected based on RGB colours and then transformed into HSV value due to RGB format is not ideal for algorithmic segmentation [8], [9]. Therefore, the author in [19] introduced two modules for fault detection, namely first layer verification and nozzle analyzer with a threshold value. These combined methods were able to detect 3D printer defects during the printing process. Nevertheless, some mistakes are not detected because the detection did not reach the minimum threshold value. Another approach is layer-wise anomaly detection, where computer vision plays a critical role in continuously detecting faults of a running 3D printer layer-by-layer [20]. However, the drawback of using computer vision is the complexity of the image preprocessing, which resulting a longer detection time.

B. ML APPROACHES

In the area of machine learning, supervised and semi-supervised learning approaches were employed [21]. For example, the author in [10] proposed collaborative learning using image preprocessing and supervised machine learning to detect defects of semi-finished 3D printed parts. The proposed model can handle 3D printer classification for two printing materials, such as acrylonitrile butadiene styrene (ABS) and polylactide (PLA). Similarly, the author in [22] presented an ML-enabled prediction of a 3D-printed microneedle using image processing and ML. The prediction was conducted in time series; hence the model can indicate any possible anomalies that might occur in the near future. On the contrary, the author in [11] employed a semi-supervised learning approach to detect the LPBF 3D printer faults using photodiodes data. The time-series data was pre-processed and downsampled using randomized singular value decomposition. The performance shows acceptable accuracy of 77% and states that using semi-supervised learning can reduce the dataset's size. Ensemble ML models were utilized in [23]. The author evaluated numerous ensemble models for the prediction of form-tap wear. The results indicate that combining multiple ML models can enhance the prediction performance based on lower root mean square error. Lastly, a layer-wise combined ML approach is also studied in [12]. The author also uses transfer learning to fasten the processing time, hence applicable for real-time fault detection.

C. DL APPROACHES

For the deep learning approaches, sensor reading and image data were employed to classify the defect in the 3D printing process. The author in [13] uses an accelerometer sensor in three axes (x, y, and z) to improve classification performance with a CNN-based model. Comparison between artificial neural networks (ANN) and SVM was also investigated. The performance shows that the proposed CNN model can achieve the highest accuracy in different class variations. Fault prevention is another critical aspect of a 3D printer monitoring system. For example, the author in [14] continuously uses nozzle temperature to anticipate future nozzle values and prevent printing failure. Regarding 3D printer faults uncertainty, the proposed model was built with temporal CNN and successfully yielded an accuracy of 97.2%.

To comply with additive manufacturing privacy and security. The author in [24] introduced an FL-based semantic segmentation for the LPBF printer by considering the pixel-wise effect. In addition, the author utilizes U-Net to differentiate the defect during the printing process and states that the proposed method results are comparable to the centralized learning approach. To the best of our knowledge, the aforementioned studies not concerning the hierarchical architecture with transfer learning using well-known DL model scenarios and detailed communication flow between the server and clients. In this paper, we proposed a hierarchical federated transfer learning architecture for the additive manufacturing industry concerning the data diversity among numerous companies while also preserving the privacy issue.

III. PROPOSED SYSTEM

This section describes the system model first, followed by the proposed HFTL architecture and algorithm that covers the concepts of FL combined with transfer learning and hierarchical architecture. Additionally, the custom dataset collection method for AM is also provided.

A. SYSTEM MODEL

The main idea of the proposed method in this paper is presented in Fig. 1. Initially, the cloud server C_s store the

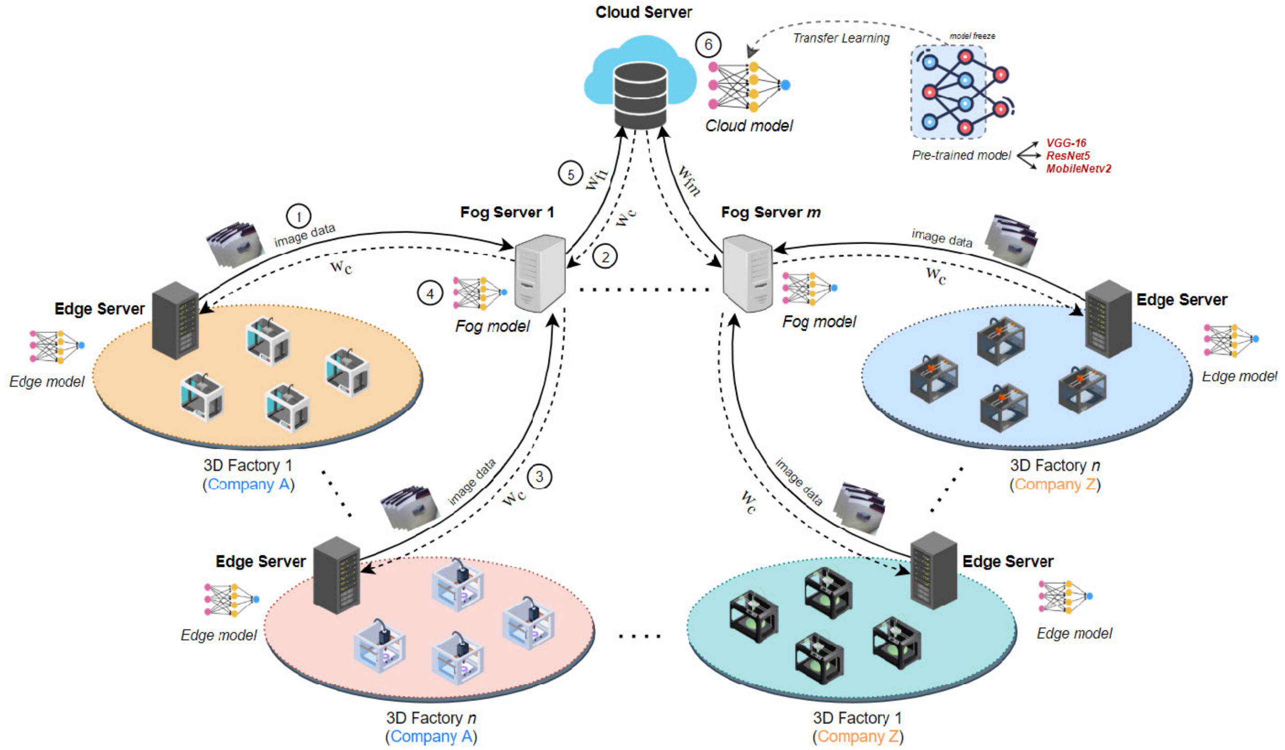


FIGURE 1. The proposed architecture of hierarchical federated transfer learning (HFTL).

cloud model as a global DL model is connected to m 3D printer company fog server $Fs_m = \{fs_1, fs_2, .., fs_m\}$. Each fog server is linked to the edge server $Es_n = \{es_1, es_2, .., es_n\}$, representing the n factory branch. Then, each branch operates p number of 3D printer $Es_{np} = \{es_{n1}, esn_{n2}, .., es_{np}\}$, where each 3D printer is able to produce i images. Therefore, the total images data produced by each branch is denoted as es_{np_i}, whereas the total images from fog server with n edge servers and p printers are calculated using (1).

$$fs_{m_i} = \sum_{j=1}^{n} \sum_{k=1}^{p} es_{(j)(k)}. \tag{1}$$

Each company uses fs_{mi} images to conduct local training in the fog server using cloud model parameters ω_c within the t_l period. After the training, the fog server forwards the trained weights ω_f to the cloud server. Hence, the list of total weight variation received by the cloud server from m fog servers can be computed using (2)

$$\omega_{c\{1,..,m\}} = \sum_{l=1}^{m} \omega f_{(l)} C. \tag{2}$$

where ω_c expressed the list of weights acquired by the cloud server, whereas C denotes the fraction of clients that joined the training process in FL. The received weights are then aggregated to obtain an updated cloud server DL model using the FederatedAveraging (FedAvg) algorithm.

This work considers the learning process to produce a robust DL model while preserving company privacy in 3D printer image data sharing. Most previous studies focused on centralized learning with a large number of datasets and left privacy issues behind. As an emerging technology, FL enables distributed training without sharing any dataset information. The iterative procedure of FL is depicted in Fig. 1.

1) Initially, each edge server gathers all 3D printer image data on each factory branch from p 3D printer and forward those data via a secured channel to their fog server.
2) After image data is received from edge devices. The selected fog server automatically downloads the current model parameter from the cloud server for the federated training process.
3) The downloaded parameter is forwarded to the edge server; hence the local model on the edge server that detects any fault during printing is also updated.
4) The downloaded parameter is also used to configure the fog model and train the local dataset from multiple edge servers. After the training process, an evaluation is performed locally.
5) All fog servers send the updated model parameter after the local training to the cloud server.
6) The parameter aggregation process is conducted in the cloud server using the FedAvg algorithm to deliver an updated version of the cloud model.

Those processes are continuously conducted for r communication rounds, or the cloud server model is convergence (e.g., accuracy=100%).

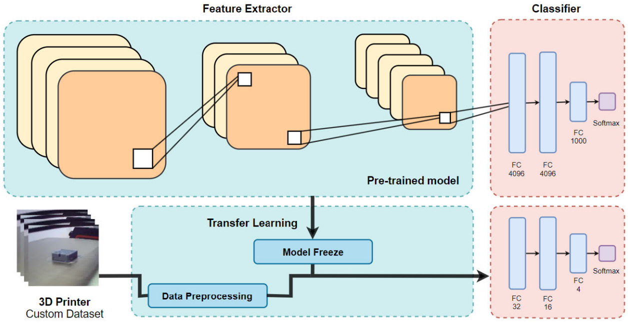


FIGURE 2. The concept of transfer learning from a pre-trained model to a specific application with a custom dataset.

B. FEDERATED TRANSFER LEARNING

Transfer learning opens opportunities to utilize a pre-trained model; hence it can be reused in different applications by eliminating the model classifier. Then a new classifier for a specific task is added to the pre-trained model. This technique significantly reduces the training time because the trainable parameters are converted into non-trainable parameters. The conversion process froze the pre-trained model feature extraction and used the initial model weight from the previous training process. Therefore, the training time mostly influenced by the new model classifier design. Overview of transfer learning illustrated in Fig. 2. The pre-trained model feature extractor is utilized for training a custom 3D printer dataset:

1) The data preprocessing is performed to ensure that the custom image data's input shape fits the pre-trained model's input size.
2) The pre-trained parameter is disabled, and a new classifier with a specific number of classes is added before training is performed.
3) In this work, the classifier is built by flattening the feature extractor output into a one-dimensional vector. Then two fully connected layers are added with sizes 32 and 16, respectively. Lastly, four neurons of dense layers are inserted to classify faults in the 3D printing process.

In this work, we employ federated transfer learning to reduce training time while maintaining high accuracy. Specifically, a pre-trained model is designed with a new classifier, as depicted in Fig. 2. Then, the model is distributed to all FL clients with random initial parameters for the classifier. Each client uses the transfer learning model in the distributed training process, which reduces the local training time. Minimizing local training time is crucial to reduce the overall training time of FL and improve model performance (e.g., lower loss, higher accuracy). Finally, based on the FL concept, those transfer learning models are stored in cloud, fog, and edge servers, as depicted in Fig. 1. Moreover, selecting a proper model for transfer learning is important since the extracted feature from the pre-trained model might differ from the additive manufacturing characteristic. Several pre-trained models have been released in the past several years through extensive training on a large dataset with more than 1000 classes. This paper investigates three popular pre-trained models built based on a convolutional neural network without any knowledge transfer.

1) VGG

The first popular model for large-scale image classification is VGG, built by increasing the network depth ranging from 16-19 with (3×3) convolution filters. First, the VGG network is trained using the ImageNet and VOC datasets with $1,000$ and $2,000$ classes, respectively. Next, a convolutional network with a max pooling layer in between uses by the VGG network, followed by a stacked fully connected layer and classifier with softmax as an activation function [25]. In this work, we employ VGG16 to minimize computational complexity.

2) ResNet

Microsoft Research introduced ResNet in 2015 to mitigate deeper neural networks, which are challenging to train. The residual connection is proposed to improve the model accuracy while maintaining low complexity. ResNet was evaluated with the ImageNet, CIFAR-10, and COCO datasets consisting of $1,000$, 80, and 10 classes. The results show that a network with residual learning efficiency is higher than a plain network without any residual connection [26]. Among many interpretations, ResNet50V2 is selected due to its performance compared to other layer variations [27] with moderate parameter size.

3) MobileNet

Google Inc. proposed MobileNet in 2017 to provide a lightweight model for the mobile user and embedded devices with limited computing power. MobileNet is built with a convolutional layer followed by 28 layers of depthwise and pointwise convolutions. The batch normalization and rectified linear unit as activation function is installed in all layers, followed by the softmax layer for classification. MobileNet, trained with the Stanford Dogs dataset, also offers a trade-off between accuracy and model complexity by varying the model shrinking hyperparameters [28]. MobileNetV2 is used in this paper as a pre-trained model considering the accuracy and model complexity trade-off.

Those pre-trained models were then utilized as feature extraction in an additive manufacturing environment with the addition of flattened, fully connected and classifier layers. Due to the pre-trained model's capability to process many variations of input size, the model input shape is configured to $(192 \times 192 \times 3)$.

C. HIERARCHICAL FEDERATED TRANSFER LEARNING

The traditional architecture of the FL process selects the specific user with their private data to conduct the training process. However, the edge, fog, and cloud concept is adopted

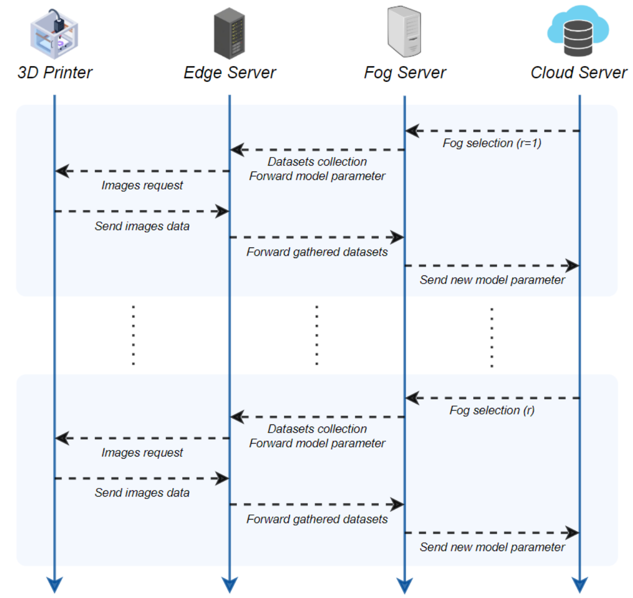


FIGURE 3. The communication flow between 3D printers, edge, fog, and cloud servers of the proposed HFTL.

in our proposed work. The communication diagram of the proposed HFTL is depicted in Fig. 3. First, the client selection is shifted from the user to the fog server. This modification allows multiple users (3D printers) within the same company to preserve their privacy by sending local data to the edge server and their fog server. Then, the fog server that acts as a client interacts with the cloud server and involve in the FL process with another company's fog server. Moreover, the proposed architecture concept enables fog model parameters to be easily forwarded to the edge server and ensures the edge model is updated. Hence, the edge server's capability to detect faults in additive manufacturing is consistently improved based on its fog server knowledge.

We proposed a hierarchical federated transfer learning (HFTL) algorithm to provide efficient fault classification in the additive manufacturing industry. The overall pseudocode of the proposed HFTL is described in the Algorithm 1. During the training process, the fog server parameter is continuously updated following the local training epoch e by using (3).

$$\omega_f \leftarrow \omega_c - \eta \nabla \ell\left(\omega_c; b\right). \tag{3}$$

the learning rate denoted by η, where $\ell\left(\omega_c; b\right)$ represent prediction loss of batch size b from local dataset B. After all training results from various fog server fs_m are received, the cloud server updates its DL model parameter $\omega_{c_{t+1}}$ by using the FedAvg aggregation method, which is expressed as (4):

$$\omega_{c_{t+1}} \leftarrow \sum_{k=1}^{K} \frac{\eta\left(k\right)}{\eta} \omega l_{t+1}^{(k)}. \tag{4}$$

Algorithm 1 Pseudocode for Hierarchical Federated Transfer Learning

```
Initialize pre-trained model {VGG16, ResNet50V2, MobileNetV2
Initialize new classifier
Initialize transfer learning model parameter
Initialize client fraction $C \leftarrow 0.9$
Initialize other parameters: $m,\ r,\ b,\ \eta$
Cloud Server executes:
for each communication round 1 to $r$ do
    $Sf_m \leftarrow m \,.\, C$
    for each $m \in Sf_m$ in parallel do
        $\omega_{fs_m} \leftarrow FogServerTraining(\omega_c)$
    end for
    $\omega_{c_{t+1}} \leftarrow$ aggregate new cloud model parameters using (4)
    $F_1 \leftarrow cloudmodel(\omega_{c_{t+1}}).evaluate()$
end for
Function $FogServerTraining(\omega_c)$
$B \leftarrow$ gather 3D printer images data from all edge servers using (1)
Forward $\omega_c$ to all edge servers
Initialize dataset preprocessing
for local epoch 1 to $e$ do
    for batch $b \in B$ do
        $\omega_f \leftarrow$ update local parameters using (3)
    end for
end for
return updated parameter $\omega_f$
```

Furthermore, adding a hierarchical strategy into an architecture might affect the overall network performance. The total training time for FL is affected by the edge, fog, and cloud server. The time needed for conducting FL training with r rounds can be calculated using (5):

$$t_{total} = \sum_{u=1}^{r} t(u)_{edge} + t(u)_{fog} + t(u)_{cloud} \tag{5}$$

where tu_{edge} is edge server processing time, tu_{fog} and tu_{cloud} represent fog and server processing time, respectively. Specifically, on the edge server side, the main task is to gather and send 3D printer images data to the fog server, which is detailed in (6):

$$tu_{edge} = tu_{gather} + tu_{send} \tag{6}$$

The fog server processing time tu_{fog} is affected by multiple processes, such as server parameter downloading, dataset processing, and local training time. Therefore, the total processing duration in the fog server is calculated with (7):

$$tu_{fog} = tu_{download} + tu_{preprocess} + tu_{training} \tag{7}$$

Lastly, parameter aggregation and model evaluation are performed on the cloud server, and the total period to complete those actions can be defined as (8):

$$tu_{cloud} = tu \sum_{v=0}^{m} \omega(v)_{c_{t+1}} + F_1(v) \tag{8}$$

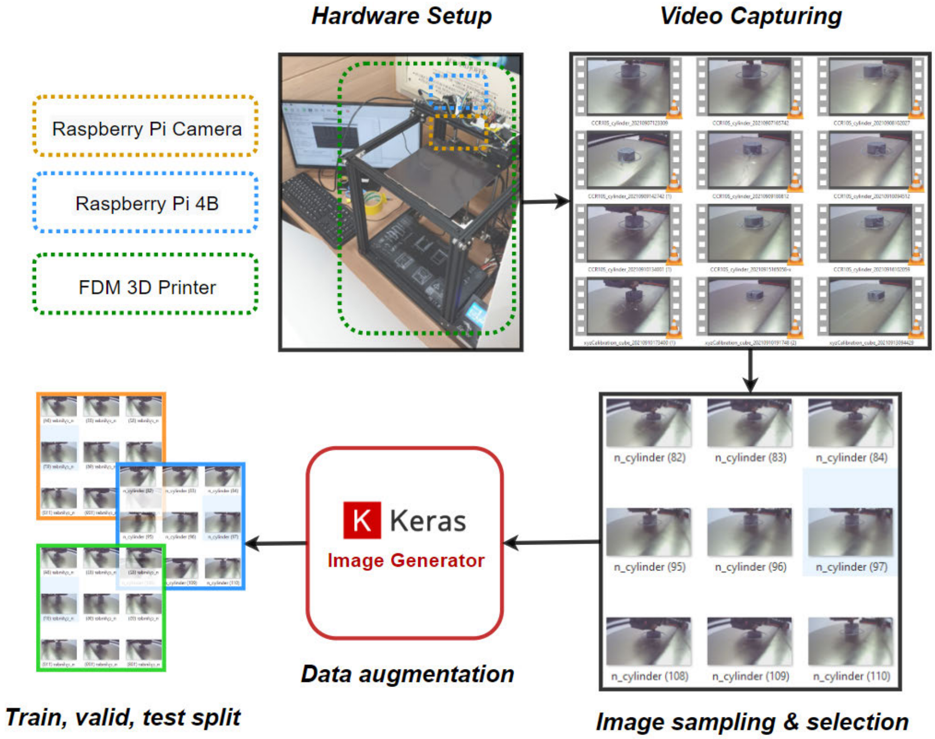


FIGURE 4. Dataset collection and preprocessing process using Creality Ender 5 Pro 3D Printer with Keras image generator.

where m signifies the total number of fog layers that joined the FL learning at each communication round.

D. FDM 3D PRINTER DATASET COLLECTION AND PREPROCESSING

A large dataset with multi-class labels is needed to provide a reliable fault detection scenario in additive manufacturing, specifically the 3D printing process. However, the availability of the printing process dataset is limited and mainly records sensory data (e.g., accelerometer, temperature). Moreover, more than applying sensor data is required to provide accurate predictions due to the different paths and temperatures characteristic of 3D objects. For instance, the accelerometer path to print a rectangle with a length of 10cm is different compared to the path used to produce a 50cm long rectangle. The fault detection accuracy might decrease if those two sensor data are compared. Therefore, the fault detection accuracy might decrease if those two sensor data are compared. Hence, we captured an image-based dataset from the 3D printing process that consists of four classes, namely `normal cube`, `faulty cube`, `normal cylinder`, and `faulty cylinder`. A printed product is classified as a fault if the final printed product has one of the following errors: stringy, unstick surface, burned layer, or immature adhesion between layers. Otherwise, the printed product is categorized as normal. The dataset was collected using a 3D printer with FDM type, specifically Creality Ender 5 Pro with PLA material. The single extruder attached to the printer is able to create a printing object using a.gcode file with a maximum dimension of 220 x 220 x 300mm, and a printing precision of ±0.1 mm.

The dataset collection process is illustrated in Fig. 4. First, a Raspberry Pi 4B that operates on top of the Raspbian operating system is attached to the Raspberry Pi camera to capture video files during the printing process. The dimension of the printed product covered in this paper is 20 × 20 × 20 mm. Each printed product generates a single video file of 25 minutes duration. The video output is saved in the form of.mp4 files with a resolution of 480p (640 × 480). For each 3D shape, five printed product with five video files is produced. Therefore, a total of 20 video files are obtained for four different classes investigated in this paper. Then, those videos were carefully sampled layer by layer to generate an image dataset using the OpenCV library, considering the image's brightness and blurriness. A total of 2, 500 images are produced from 20 video files. To further increase the size of the datasets, the data augmentation process is conducted using the Keras image generator library. A combination of image rotation, width shift, height shift, zoom range, and horizontal flip techniques was used. Based on the data augmentation process, the total images in the dataset expanded to 8, 068 images. Lastly, the dataset is divided into train, validation, and test with a percentage of 70%, 10%, and 20%, respectively.

TABLE 1. Hardware and software specification for simulation work.

Component Name	Description
Processors	27x Intel(R) Core(TM) i9-10940X CPU
Graphic Cards	3x NVIDIA GeForce RTX 3090
Random Access Memory	125.5 Gibibytes
Operating System	Ubuntu 20.04.4 LTS
Environment	Virtualenv with Python 3.7.12. (Flower 1.0 & TensorFlow 2.9.1)

In this paper, the primary focus is on the utilization of an FDM 3D Printer. However, advanced dataset collection and preprocessing are necessary when considering various printing techniques in additive manufacturing, such as LPBF or directed energy deposition (DED). For example, the LPBF process leads to the formation of columnar grain structures in alloys [29], particularly in low-stiffness components [30], which increases the difficulty of defect detection. Therefore, solely relying on a layer-wise approach is inadequate, and additional feature information, such as the heat distribution of the printing object and its surface, must be carefully considered.

IV. RESULTS AND DISCUSSIONS

This section covers simulation configuration, performance metrics to evaluate the proposed architecture, and various scenarios used to investigate the effectiveness of the HFTL algorithm compared to the traditional FL approach.

A. SIMULATION SETUP

Flower [31] as an FL-based framework is utilized in this work to demonstrate the proposed algorithm. Flower supports python-based programming language, enabling large-scale FL simulation and experiments with independent and identically distributed (IID) and non-IID data. In addition, Flower support various distributed parameter aggregation algorithm, such as FedAvg, FedBN, FedAdam, and others, with a connection to TensorFlow [32], PyTorch [33], and scikit-learn [34]. In this work, the hardware and software details to execute the proposed HFTL algorithm in Flower with TensorFlow are detailed in Table 1.

B. PERFORMANCE METRICS

Various performance metrics were used to validate the effectiveness of the proposed HFTL algorithm. First, we investigate the performance of the proposed HFTL compared with the non-hierarchical approach. Then, we vary the different data distributions among multiple clients by using IID and non-IID data. The performance evaluation was performed on the cloud server to generalize the model performance, as depicted in the Algorithm 1. Not only accuracy A and loss L were investigated, but also model precision P, recall R, F1-score F_1, and overall network delay t_{total} were evaluated using the following formulas:

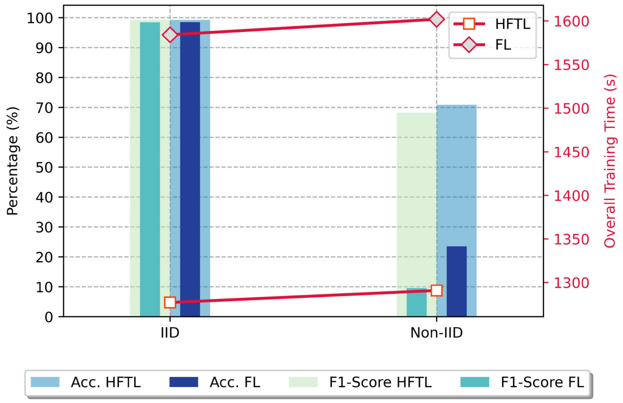


FIGURE 5. Performance of proposed HFTL compared to traditional FL without hierarchical architecture in terms of accuracy, F1-score, and overall network delay using IID and non-IID data.

$$A = \frac{T_p + T_n}{T_p + T_n + F_n + F_p} \times 100. \tag{9}$$

$$P = \frac{T_p}{T_p + F_p}. \tag{10}$$

$$R = \frac{T_p}{T_p + F_n}. \tag{11}$$

$$F_1 = 2 \times \left(\frac{P \times R}{P + R}\right). \tag{12}$$

where T_p denotes true positive, T_n refers true negative, followed by F_p and F_n that represents false positive and false negative, respectively.

C. HIERARCHICAL ARCHITECTURE ANALYSIS

The proposed HFTL architecture is compared with traditional FL without a hierarchical structure for the first investigation. Several parameters were used to investigate this scenario, such as the total fog servers being set to 10 with two edge devices and adopting the MobileNetV2 structure as the pre-trained model. Fig. 5 depicts the evaluation results from various performance metrics for IID and non-IID data allocation. In terms of accuracy, the proposed hierarchical architecture can enhance IID data's performance from 98.10% to 99.26%. Then, after the proposed HFTL is implemented using non-IID data, the accuracy improvement is enhanced significantly. For example, the traditional FL only produces 23.54% accuracy, whereas the proposed HFTL is able to achieve 68.81%.

Moreover, we calculate precision and recall to obtain the F1-score for both IID and non-IID data. As shown in Fig. 5, the F1-score of the traditional FL network achieves 98.49%, whereas the proposed HFTL successfully improved to 99.26%. On the other hand, traditional FL with non-IID data that represents the real-world data distribution provides an F1-score of 9.5%. For the proposed HFTL architecture, superior improvement is obtained by giving an F1-score of

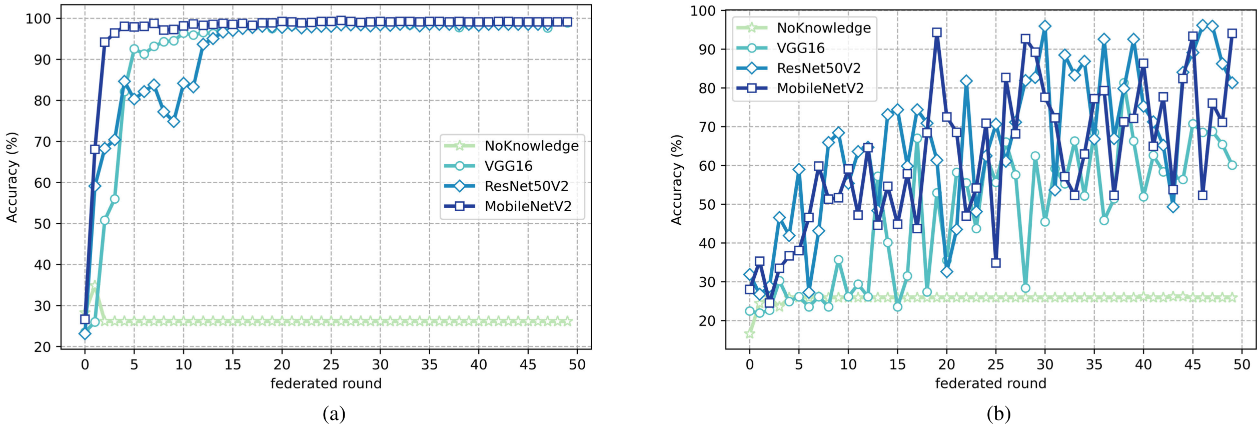


FIGURE 6. Accuracy performance among different federated transfer learning models in 50 federated rounds for 10 fog servers using: (a) IID Data; and (b) non-IID Data.

68.81%. The performance advancement obtained in this work is due to the proposed HFTL architecture's ability to perform an efficient training process by gathering multiple image data from different edge servers. Also, the data collection process performed in the fog server is proven effective since the non-similarity boundaries between FL participants are reduced.

Furthermore, the overall federated training process of the proposed HFTL and traditional FL is addressed. Using a similar configuration, Fig. 5 also shows the time required by HFTL and FL to accomplish 50 rounds of iterative distributed training. Again, the proposed HFTL is superior to traditional FL architecture by using IID and non-IID data. As a result, the average training time of the proposed HFTL and traditional FL is 1, 283.51 s and 1, 592.36 s, respectively. Thereby total training time reduction by adopting the proposed HFTL is 24.06%. Considering the number of communication rounds and the number of participants who joined the distributed learning process, the proposed HFTL can significantly reduce federated training time by gathering edge server data and conducting local training on the fog server.

Additionally, data privacy and security is not an issue considering the edge and fog server within the same company. Since the proposed HFTL architecture outperforms the traditional FL in terms of accuracy, F1-score, and training time different data distribution, the remaining analysis is conducted using the HFTL network structure.

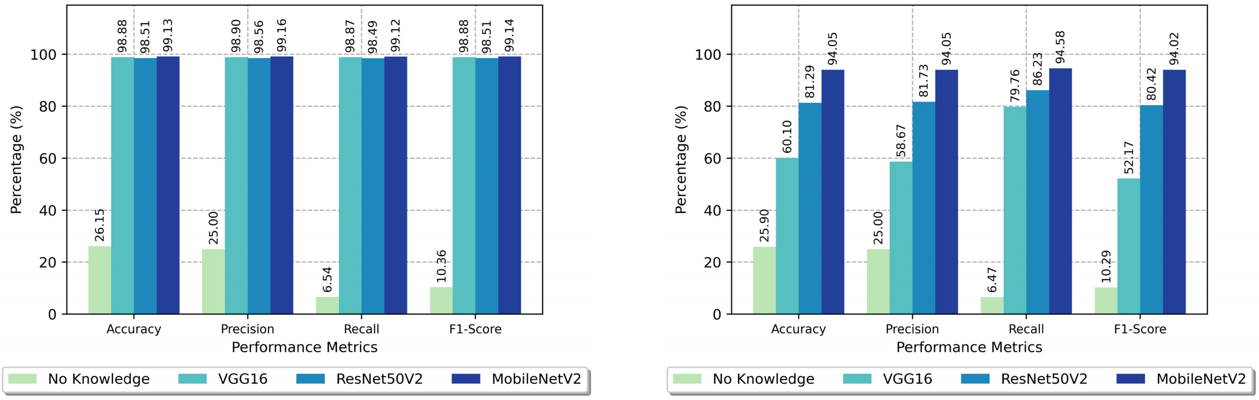
D. FEDERATED TRANSFER LEARNING EFFECTIVENESS

To show the robustness of the transfer learning technique in the proposed HFTL, we compared the performance of three different pre-trained models and the proposed classifier with a CNN-based model without knowledge transfer. Similarly, the total number of fog servers is set to 10 with 20 edge servers and 50 rounds of federated iteration. Finally, the distribution of the IID and non-IID datasets were investigated with four classes to provide extensive results.

Fig. 6 shows the iterative learning process in the FL with a total communication round of 50. The cloud server model performance, based on accuracy with IID data depicted in Fig. 6 (a). It can be observed that the performance of DL pre-trained models is superior to the CNN-based model without knowledge transfer. For example, MobileNetV2 successfully learn the 3D printer dataset with an accuracy of 99.13%. Besides, VGG16 and ResNet50V2 are able to achieve acceptable accuracy of 98.88% and 98.10%, respectively. MobileNet can achieve the highest performance in the first five iterations based on three pre-trained models investigated in this study. On the other hand, the CNN-based model cannot learn complex features from the image dataset despite the fact that IID data distribution is used to evaluate the model performance. For 50 federated rounds, the basic model without knowledge transfer only provides 26.15% accuracy for four different classes. This result indicates that the basic model is not able to learn image features during the federated process.

Moreover, federated learning was also performed using non-IID data with results depicted in Fig. 6 (b) to validate the previous results using IID data. As expected, the performance of all pre-trained models is degraded. This phenomenon occurs due to different classes owned by each fog server incorporated in the FL process. Therefore, the cloud server suffers from an aggregation process of multiple fog server parameters characteristic and influences the classification accuracy. MobileNetV2, which is able to deliver steady performance using IID data for 50 rounds of FL, provides inconsistent results while using non-IID data. For instance, MobileNetV2 yielded 94.05% accuracy for the last communication round, whereas VGG16 and ResNet50V2 accuracies degraded to 60.10% and 81.29%, respectively. Besides, the CNN-based model produces unsatisfactory performance with a constant accuracy of 25.90% without knowledge transfer.

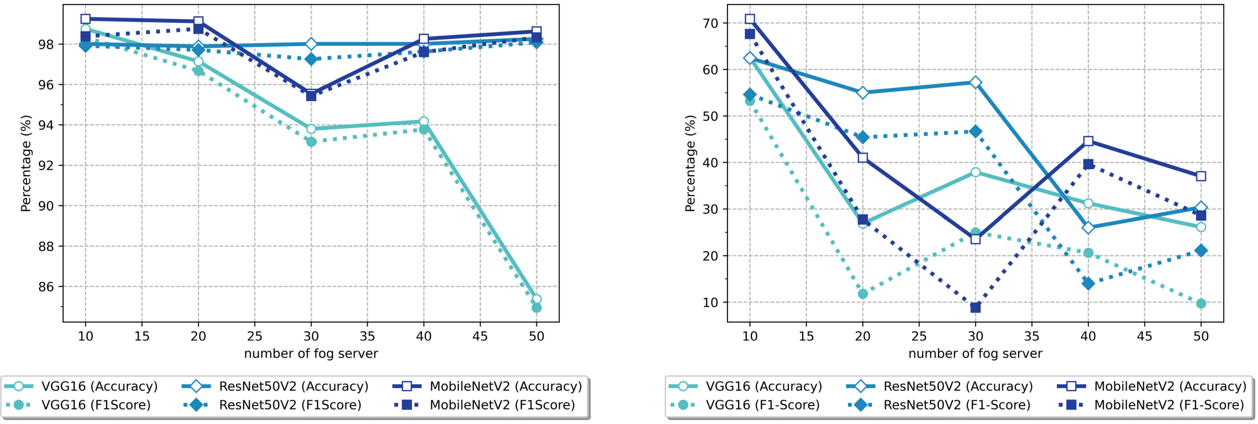
Furthermore, precision, recall, and F1-score were also used to evaluate model robustness. Fig. 7 (a) depicts the performance of all models studied in this work using IID data. Generally, the pre-trained models achieve outstanding performance compared to the CNN-based model. The



(a)

(b)

FIGURE 7. Various performance results of the pre-trained compared to CNN-based models without knowledge transfer for 10 fog servers using: (a) IID Data; and (b) non-IID data.



(a)

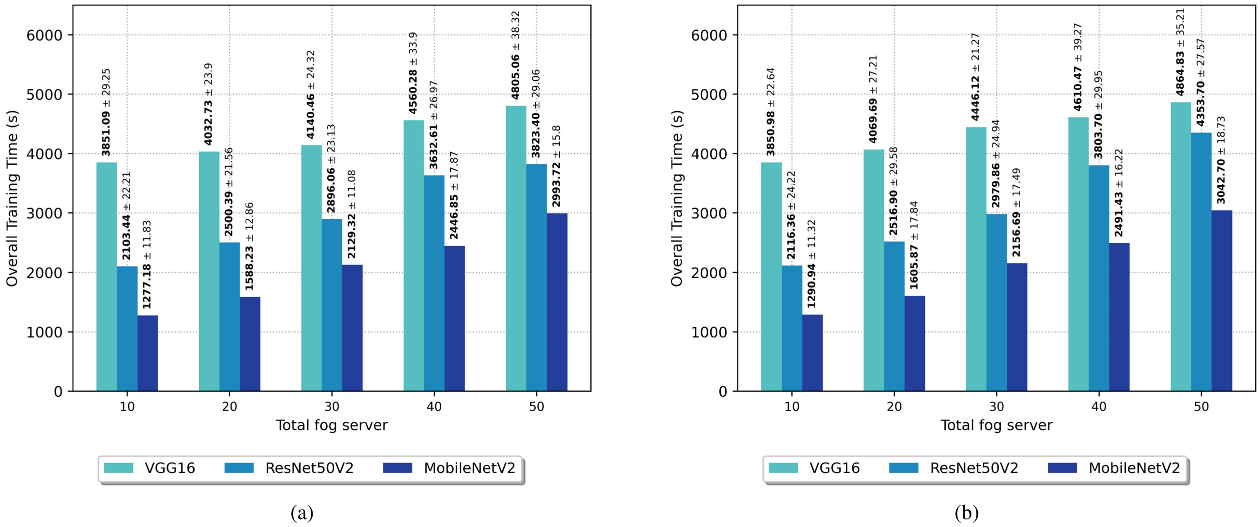
(b)

FIGURE 8. Accuracy and F1-score results among different transfer learning models versus the number of fog servers in HFTL architecture using: (a) IID Data; and (b) non-IID data.

significant difference is due to the basic model's inability to learn from the 3D printer image datasets. Lastly, based on non-IID data, the performance gap between pre-trained models becomes clearer, as shown in Fig. 7 (b). The MobileNetV2 yielded the highest performance, followed by ResNet50V2, VGG16 and the CNN-based model without knowledge transfer. Referring to the comparative results of the CNN basic model with pre-trained models, the federated transfer learning process successfully improves the overall model performance. In contrast, the basic CNN-based model is insufficient to extract 3D printer image data. Therefore, we focused on the pre-trained model investigation for the following performance evaluation.

E. SCALABILITY ANALYSIS

To better understand the federated transfer learning model performance, the total number of fog servers joining the HFTL architecture was investigated by considering a large number of additive manufacturing companies worldwide. Besides, scalability is an important aspect of the FL process. In this work, we evaluate the total number of fog servers ranging from 10 to 50 for IID and non-IID data. The first evaluation was conducted using IID data, which is displayed in Fig. 8 (a). Generally, as the total number of fog servers expanded, the cloud server model performance, such as accuracy and F1-score, decreased. For example, in the first 10 fog servers, the performance of MobileNetV2 in terms of accuracy reaches 99.13% and 99.07% for the F1-score. However, as the total fog server increased to 50, the performance of MobileNetV2 narrowly degraded to 98.63% accuracy and 984% for the F1-score. Similar behavior was achieved by ResNet50V2, with a slight performance reduction. However, the VGG16 network cannot comply with the fog server addition, resulting large performance gap. An average 13.66% performance drop is acquired by increasing the fog server from 10 to 50. The performance degradation occurs because the pre-trained VGG16 model cannot learn from a smaller portion of data, as the edge device's total image reduces while the fog server is added.



(a)

(b)

FIGURE 9. Overall training time required by HFTL architecture with fog server variations to perform hierarchical distributed learning for three pre-trained models using: (a) IID data; and (b) non-IID data.

In addition, the accuracy and F1-score of all pre-trained models used in federated transfer learning for this study were tested using non-IID data. Fig. 8 (b) depicts the overall performance of HFTL architecture using three different models versus the number of fog servers. Although the performance degradation in IID data is acceptable, the results for non-IID data are completely different. The significant accuracy and F1-score reduction are obtained from all transfer learning models. For instance, VGG achieves 60.9% accuracy with a 52.09% F1-score by performing 10 fog servers for the FL process. Surprisingly, after the fog server increased to 50, the average performance degradation was 684%, with accuracy and F1-score of 26.14% and 10.06%, respectively. Not only VGG16, but the degradation also occurs for ResNet50V2 with an average of 68.12% and MobileNetV2 with an average of 65.64%. Based on Fig. 8(a), we discovered that client increment reduces the model performance with IID data. Scaling the HFTL architecture by inserting a new fog server with non-IID data worsens the FL's performance. However, based on the discussed outputs, MobileNetV2 is more suitable for HFTL in an additive manufacturing environment than other pre-trained models. The MobileNetV2 structure that uses residual connection from ResNet and depthwise convolutional layer can extract 3D printer image features better than other models.

F. EFFICIENCY ANALYSIS

Herein, the efficiency of the proposed HFTL by implementing transfer learning is analyzed in terms of overall federated training time that comprise communication between cloud, fog, and edge server. As shown in Fig. 9, the findings indicate that the total federated training time linearly increased as the client extended. In the experiments, we assume that the communication link is in ideal circumstances; hence the communication latency between servers is disregarded. The experiment was conducted five times, and the results were then averaged. Fig. 9 (a) with IID data shows that each pre-trained model dramatically affects the overall federated training using a different number of fog servers (i.e., $Fs = 10; 20; 30; 40; 50$). For instance, MobileNetV2 requires 1227.18 ± 11.83 s to perform 50 rounds of FL, whereas ResNet50V2 and VGG16 require 2103.44 ± 22.21 s and 3851.09 ± 29.25 s, respectively. A significant increase of 213.81% between MobileNetV2 and VGG16 is acquired with 10 fog servers, then 60.50% for 50 fog servers. Therefore, if the number of additive manufacturing companies joining the HFTL is minimal, the MobileNet becomes the best option with high efficiency and accuracy trade-off.

Furthermore, the impact of non-IID data in federated training was also investigated. As depicted in Fig. 9 (b), the fog server increment leads to longer federated training time, similar to IID data outcomes in Fig. 9 (a). This enlargement occurs because the number of selected fog servers for each training iteration is increased following the participant size. The client fraction C was configured to 0.9 to facilitate the model generalization with a larger sample size; hence, 90% of clients are included in the training process.

Lastly, referring to the pre-trained model architecture, MobileNetV2 has a 3.5M parameter, followed by VGG16 with 138.4M and ResNet50V2 with 25.6M. It indicates that the model complexity affects the federated training process. However, the performance gap becomes smaller as the client increases. Finally, the performance trade-off must be carefully considered in the proposed HFTL. ResNet50V2 is the best option for IID and non-IID data for large-scale networks with acceptable performance trade-offs, whereas MobileNetV2 is suitable for low-scale HFTL networks.

G. APPLICABILITY TO OTHER PRINTING TECHNIQUES

Based on the findings presented in this paper, significant improvements have been achieved by employing a hierarchical federated approach in terms of classification accuracy and efficiency for the FDM printing technique. As highlighted in Section III-D, another challenge lies in implementing the proposed HFTL approach to different printing techniques in additive manufacturing, such as LPBF and DED, which utilize metal powders as printing materials. The fundamental concept of our approach remains similar across LPBF and DED, as it considers a layer-wise technique.

We believe the proposed HFTL approach can be successfully applied to LPBF and DED by providing high-quality datasets for each layer alongside relevant sensory data, such as printing surface temperature. Furthermore, the proposed HFTL framework enables multiple companies to collaborate to generate improved models without sharing sensitive information and compromising data privacy.

V. CONCLUSION AND FUTURE DIRECTION

This article introduces hierarchical federated learning by adopting transfer learning from a well-known DL model (i.e., HFTL) for the additive manufacturing industry to perform secure distributed learning. The proposed HFTL is built from edge, fog, and cloud concepts, where the fog server performs the data collection and learning process instead of the edge server. It also presents the effective communication process between each tier server.

The results were obtained from thorough experiments to validate the effectiveness of the proposed HFTL architecture compared to traditional FL networks. Based on the findings, the proposed HFTL can significantly improve FL performance in terms of accuracy, F1-Score, and training time for both IID and non-IID data. For instance, the performance improvement using non-IID data obtained using the proposed HFTL is 45% in terms of accuracy, 59% in terms of F1-score, and 24% faster training time. Moreover, based on three different pre-trained models, it is concluded that MobileNetV2 is the best transfer learning model for small-scale distributed learning, and ResNet50V2 is suitable for large-scale HFTL networks based on the performance trade-off.

Despite the outstanding findings of the proposed HFTL presented in this work, a few topics can still be addressed in future research, such as:

- The security element can be enhanced by utilizing blockchain between fog and cloud servers; hence, the model parameter is completely secured from malicious attacks.
- Integrating the blockchain network in FL [35] with an incentive mechanism with a blockchain-based reward (i.e., cryptocurrency).
- Implementing a client selection mechanism [36] to enhance the HFTL performance further.
- Real-world implementation and performance evaluation concerning multiple additive manufacturing companies is a promising research direction.
- Finally, implementing the technology on different printing techniques, such as LPBF, should be considered due to the distinct characteristics of the printing processes of FDM and LPBF.

REFERENCES

[1] H. Tran-Dang, N. Krommenacker, P. Charpentier, and D.-S. Kim, "The Internet of Things for logistics: Perspectives, application review, and challenges," *IETE Tech. Rev.*, vol. 39, no. 1, pp. 1–29, 2022.

[2] S. El Hamdi, A. Abouabdellah, and M. Oudani, "Industry 4.0: Fundamentals and main challenges," in *Proc. Int. Colloq. Logistics Supply Chain Manage. (LOGISTIQUA)*, Jun. 2019, pp. 1–5.

[3] M. Kalender, S. E. Kiliç, S. Ersoy, Y. Bozkurt, and S. Salman, "Additive manufacturing and 3D printer technology in aerospace industry," in *Proc. 9th Int. Conf. Recent Adv. Space Technol. (RAST)*, Jun. 2019, pp. 689–694.

[4] A. Coro, L. M. Macareno, J. Aguirrebeitia, and L. N. L. De Lacalle, "A methodology to evaluate the reliability impact of the replacement of welded components by additive manufacturing spare parts," *Metals*, vol. 9, no. 9, p. 932, Aug. 2019. [Online]. Available: https://www.mdpi.com/2075-4701/9/9/932

[5] F. Calignano, D. Manfredi, E. P. Ambrosio, S. Biamino, M. Lombardi, E. Atzeni, A. Salmi, P. Minetola, L. Iuliano, and P. Fino, "Overview on additive manufacturing technologies," *Proc. IEEE*, vol. 105, no. 4, pp. 593–612, Apr. 2017.

[6] T. Mitroudas, K. A. Tsintotas, N. Santavas, A. Psomoulis, and A. Gasteratos, "Towards 3D printed modular unmanned aerial vehicle development: The landing safety paradigm," in *Proc. IEEE Int. Conf. Imag. Syst. Techn. (IST)*, Jun. 2022, pp. 1–6.

[7] Y. H. Ying, "New generation 3D printed on-chip energy storage devices," in *Proc. IEEE Int. Conf. Electron Devices Solid-State Circuits (EDSSC)*, Aug. 2016, pp. 472–475.

[8] F. Baumann and D. Roller, "Vision based error detection for 3D printing processes," in *Proc. MATEC Web Conf.*, vol. 59, 2016, pp. 1–7, doi: 10.1051/matecconf/20165906003.

[9] R. A. Lyngby, J. Wilm, E. R. Eiríksson, J. B. Nielsen, J. N. Jensen, H. Aanæs, and D. B. Pedersen, "In-line 3D print failure detection using computer vision," in *Proc. Euspen*, 2017, pp. 1–4. [Online]. Available: https://www.euspen.eu/knowledge-base/AM17133.pdf

[10] U. Delli and S. Chang, "Automated process monitoring in 3D printing using supervised machine learning," *Proc. Manuf.*, vol. 26, pp. 865–870, Jan. 2018.

[11] I. A. Okaro, S. Jayasinghe, C. Sutcliffe, K. Black, P. Paoletti, and P. L. Green, "Automatic fault detection for laser powder-bed fusion using semi-supervised machine learning," *Additive Manuf.*, vol. 27, pp. 42–53, May 2019.

[12] V. Kadam, S. Kumar, A. Bongale, S. Wazarkar, P. Kamat, and S. Patil, "Enhancing surface fault detection using machine learning for 3D printed products," *Appl. Syst. Innov.*, vol. 4, no. 2, p. 34, May 2021.

[13] M. Verana, C. I. Nwakanma, J. M. Lee, and D. S. Kim, "Deep learning-based 3D printer fault detection," in *Proc. 12th Int. Conf. Ubiquitous Future Netw. (ICUFN)*, Aug. 2021, pp. 99–102.

[14] G. A. R. Sampedro, D. J. S. Agron, G. C. Amaizu, D.-S. Kim, and J.-M. Lee, "Design of an in-process quality monitoring strategy for FDM-type 3D printer using deep learning," *Appl. Sci.*, vol. 12, no. 17, p. 8753, Aug. 2022.

[15] K. Paraskevoudis, P. Karayannis, and E. P. Koumoulos, "Real-time 3D printing remote defect detection (Stringing) with computer vision and artificial intelligence," *Processes*, vol. 8, no. 11, p. 1464, Nov. 2020.

[16] L. Scime, D. Siddel, S. Baird, and V. Paquit, "Layer-wise anomaly detection and classification for powder bed additive manufacturing processes: A machine-agnostic algorithm for real-time pixel-wise semantic segmentation," *Additive Manuf.*, vol. 36, Dec. 2020, Art. no. 101453.

[17] G. Drainakis, K. V. Katsaros, P. Pantazopoulos, V. Sourlas, and A. Amditis, "Federated vs. Centralized machine learning under privacy-elastic users: A comparative analysis," in *Proc. IEEE 19th Int. Symp. Netw. Comput. Appl. (NCA)*, Nov. 2020, pp. 1–8.

[18] B. Li, S. Ma, R. Deng, K. R. Choo, and J. Yang, "Federated anomaly detection on system logs for the Internet of Things: A customizable and communication-efficient approach," *IEEE Trans. Netw. Service Manage.*, vol. 19, no. 2, pp. 1705–1716, Jun. 2022.

[19] S. A. Langeland, ''Automatic error detection in 3D pritning using computer vision,'' M.S. thesis, Dept. Inform., Univ. Bergen, Bergen, Norway, 2020.

[20] A. L. Petsiuk and J. M. Pearce, ''Open source computer vision-based layer-wise 3D printing analysis,'' *Additive Manuf.*, vol. 36, Dec. 2020, Art. no. 101473.

[21] G. A. R. Sampedro, S. M. Rachmawati, D.-S. Kim, and J.-M. Lee, ''Exploring machine learning-based fault monitoring for polymer-based additive manufacturing: Challenges and opportunities,'' *Sensors*, vol. 22, no. 23, p. 9446, Dec. 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/23/9446

[22] M. Rezapour Sarabi, M. M. Alseed, A. A. Karagoz, and S. Tasoglu, ''Machine learning-enabled prediction of 3D-printed microneedle features,'' *Biosensors*, vol. 12, no. 7, p. 491, Jul. 2022.

[23] A. Bustillo, L. N. López De Lacalle, A. Fernández-Valdivielso, and P. Santos, ''Data-mining modeling for the prediction of wear on forming-taps in the threading of steel components,'' *J. Comput. Design Eng.*, vol. 3, no. 4, pp. 337–348, Oct. 2016, doi: 10.1016/j.jcde.2016.06.002.

[24] M. Mehta and C. Shao, ''Federated learning-based semantic segmentation for pixel-wise defect detection in additive manufacturing,'' *J. Manuf. Syst.*, vol. 64, pp. 197–210, Jul. 2022.

[25] K. Simonyan and A. Zisserman, ''Very deep convolutional networks for large-scale image recognition,'' 2014, *arXiv:1409.1556*.

[26] K. He, X. Zhang, S. Ren, and J. Sun, ''Deep residual learning for image recognition,'' 2015, *arXiv:1512.03385*.

[27] W. Hao, M. Han, H. Yang, F. Hao, and F. Li, ''A novel Chinese herbal medicine classification approach based on EfficientNet,'' *Syst. Sci. Control Eng.*, vol. 9, no. 1, pp. 304–313, Jan. 2021.

[28] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, ''MobileNets: Efficient convolutional neural networks for mobile vision applications,'' 2017, *arXiv:1704.04861*.

[29] J. D. Pérez-Ruiz, L. N. L. de Lacalle, G. Urbikain, O. Pereira, S. Martínez, and J. Bris, ''On the relationship between cutting forces and anisotropy features in the milling of LPBF inconel 718 for near net shape parts,'' *Int. J. Mach. Tools Manuf.*, vol. 170, Nov. 2021, Art. no. 103801. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0890695521001103

[30] J. D. Pérez-Ruiz, F. Marin, S. Martínez, A. Lamikiz, G. Urbikain, and L. N. L. De Lacalle, ''Stiffening near-net-shape functional parts of inconel 718 LPBF considering material anisotropy and subsequent machining issues,'' *Mech. Syst. Signal Process.*, vol. 168, Apr. 2022, Art. no. 108675. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0888327021009985

[31] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, J. Fernandez-Marques, Y. Gao, L. Sani, K. Hei Li, T. Parcollet, P. P. B. de Gusmão, and N. D. Lane, ''Flower: A friendly federated learning research framework,'' 2020, *arXiv:2007.14390*.

[32] M. Abadi et al., ''TensorFlow: A system for large-scale machine learning,'' 2016, *arXiv:1605.08695*.

[33] A. Paszke et al., ''PyTorch: An imperative style, high-performance deep learning library,'' 2019, *arXiv:1912.01703*.

[34] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, A. Müller, J. Nothman, G. Louppe, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, ''Scikit-learn: Machine learning in Python,'' 2012, *arXiv:1201.0490*.

[35] S. Bai, G. Yang, G. Liu, H. Dai, and C. Rong, ''NttpFL: Privacy-preserving oriented no trusted third party federated learning system based on blockchain,'' *IEEE Trans. Netw. Service Manage.*, vol. 19, no. 4, pp. 3750–3763, Dec. 2022.

[36] 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 Things*, vol. 21, Apr. 2023, Art. no. 100657. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S254266052200138X

MADE ADI PARAMARTHA PUTRA received the M.S. degree in electrical and computer engineering from the Bandung Institute of Technology, West Java, Indonesia, in 2019. He is currently a full-time Lecturer of informatics engineering with STMIK Primakara, Bali, Indonesia. His main research interests include named data networks (NDN), the real-time Internet of Things, federated learning optimization, and energy efficient architecture.

SYIFA MALIAH RACHMAWATI received the B.S. degree from Telkom University, Indonesia, in 2018. From 2019 to 2021, she was a Software Developer in Indonesia. She is currently with the Research and Development Center, Philippine Coding Camp, Manila, Philippines. Her research interest includes deep learning techniques and their application in the industrial Internet of Things.

MIDETH ABISADO (Member, IEEE) is currently the Director of the CCIT Graduate Programs Department, National University, Manila. She heads research on Harnessing Natural Language Processing for Community Participation. Social science, machine learning, and natural language processing principles and techniques are used in the study. It is well anticipated that thematic based on dashboard analytics will be used for policy recommendations for the Government. She has 23 years of experience in education and research. Her passion is to encourage women to choose careers in computing and prepare and mold the next generation of Filipino IT professionals and leaders in the country. Her research interests include emphatic computing, social computing, human–computer interaction, and human language technology.. She is an Associate Member of the National Research Council of the Philippines and a Board Member of the Computing Society of the Philippines Special Interest Group for Women in Computing.

GABRIEL AVELINO SAMPEDRO (Member, IEEE) received the B.S. and M.S. degrees in computer engineering from Mapúa University, Manila, Philippines, in 2018, and the Ph.D. degree in IT convergence engineering from the Kumoh National Institute of Technology, in 2023. She is currently an Assistant Professor with the Faculty of Information and Communication Studies, University of the Philippines Open University. Her research interests include real-time systems, embedded systems, robotics, and biomedical engineering.