Privacy-Preserving Federated Learning in Oil Palm Industry for Fresh Fruit Bunch Ripeness Grading

Patchanee Laddawong [1], Yutthapong Pianroj [2,3], Piyanart Chotikawanid [2], Teerasak Punvichai [3,4], Saysunee Jumrat [2,3], and Jirapond Muangprathub [2,3,*]

[1] College of Digital Science, Prince of Songkla University, Hat Yai Campus, Songkhla, Thailand
[2] Faculty of Science and Industrial Technology, Prince of Songkla University, Surat Thani Campus, Surat Thani, Thailand
[3] Integrated High-Value Oleochemical Research Center, Prince of Songkla University, Surat Thani Campus, Surat Thani, Thailand
[4] Faculty of Innovative Agriculture, Fisheries and Food, Prince of Songkla University, Surat Thani Campus, Surat Thani, Thailand
Email: patchanee.l@psu.ac.th (P.L.); yutthapong.p@psu.ac.th (Y.P.); piyanart.ko@psu.ac.th (P.C.); teerasak.p@psu.ac.th (T.P.); saysunee.j@psu.ac.th (S.J.); jirapond.m@psu.ac.th (J.M.)
*Corresponding author

***Abstract*—This study proposes a novel privacy-preserving federated learning method, Privacy-Preserving Federated Learning with Federated Averaging (PP-FedAvg), tailored for the oil palm industry to enhance data security while predicting ripeness levels and oil yield of fresh fruit bunches without sharing sensitive data. The primary contribution of this work lies in developing the PP-FedAvg model, which combines encryption techniques with federated learning to ensure secure and accurate classification, achieving 92.5% accuracy across 5209 images of four ripeness levels (raw, semi-raw, semi-ripe, ripe). The prediction results are integrated into a web application, providing practical tools to improve decision-making and operational efficiency in palm oil mills.**

***Keywords*—federated learning, privacy-preserving federated learning, oil palm, ripeness, grading, oil palm fruit**

I. Introduction

Thailand is the third-largest producer of palm oil in the world, accounting for 4% of global production. However, the Thai palm oil industry faces significant issues due to the overcapacity of crude palm oil mills, which currently operate at only 43% [1] of their potential capacity. Over the past decade, the number of crude palm oil mills has more than doubled, intensifying competition and reducing profitability. This situation has led to the closure of some mills due to unprofitability.

Errors in ripeness grading exacerbate this problem by directly impacting oil yield and quality. Studies estimate that misgraded fruit can reduce oil extraction efficiency by 10–20%, potentially costing an average mill several tens of thousands of dollars annually. With over 150 palm oil mills in Thailand, the cumulative economic loss due to inaccurate ripeness grading could reach millions of dollars each year. To regulate fairness, the Thai government mandates that palm oil mills purchase Fresh Fruit Bunches (FFB) based on the oil percentage at a fair and reasonable price [2]. The oil percentage is closely tied to the ripeness level of the fruits in the bunch, determined by fruit color and detachment rates [3].

In practice, mill staff visually assess the ripeness of FFB, but this process is prone to inaccuracies caused by limited personnel, time constraints, and subjective judgment. These errors can lead to mills purchasing FFB at prices higher than the actual oil yield, resulting in financial losses and reduced profitability. Furthermore, such inefficiencies strain operational capacities and create competitive disadvantages. Ensuring accurate ripeness grading is therefore critical to the sustainability and profitability of the Thai palm oil industry.

To address this problem, various methods have been considered for assessing the ripeness of palm fruits, including Convolutional Neural Network (CNN) and Federated Averaging (FedAVG). CNN can process and classify images with high accuracy without sending data to external devices, ensuring high data security. However, it requires significant processing resources and slow model updates with new data. On the other hand, FedAVG offers fast model updates and distributed processing across multiple devices, but it has lower data security since updated models are sent to a central server, posing a risk of data leakage. Additionally, combining data will benefit the creation of predictive models, enabling machine learning to perform better than processing data from each node individually. Thus, this work developed a new method that combines the advantages of both approaches, named

Manuscript received November 1, 2024; revised November 22, 2024; accepted January 7, 2025; published March 20, 2025.

Privacy-Preserving Federated Learning Average (PP-FedAvg). This method integrates Federated Learning (FL) with privacy preservation, avoiding the need to send raw data to a central server by updating the model and sending only partial representative data [4, 5], ensuring high data security and fast model updates.

FL based on FedAVG has gained significant attention in various industries due to its ability to enable collaborative model training while safeguarding sensitive data. In healthcare, FL has been employed to train models on patient data across multiple hospitals without sharing raw medical records, ensuring data privacy while improving diagnostic accuracy. For instance, FL has been used for predicting diseases such as diabetes and cancer, where preserving patient confidentiality is critical. Similarly, in the finance sector, FL is applied to detect fraudulent transactions across multiple banks, combining distributed data insights to enhance model performance without exposing sensitive customer information. In the energy industry, FL has been utilized for smart grid optimization, where data from distributed energy resources is aggregated securely to improve energy distribution efficiency. These applications demonstrate FL's adaptability in scenarios that require high levels of data security and collaboration. By drawing on the success of FL in these industries, this study extends its application to the oil palm industry, addressing the unique challenges of privacy and predictive accuracy in ripeness grading.

FL offers a promising solution by enabling collaborative model training without direct data exchange. Despite its advantages, conventional FL approaches such as FedAvg are vulnerable to privacy breaches during model updates. This study addresses this gap by introducing PP-FedAvg, a privacy-preserving federated learning model that combines encryption techniques with FL to enhance data security while maintaining high predictive accuracy in ripeness grading and oil yield prediction.

This study developed a centralized machine learning web application capable of operating across various communication devices. The proposed PP-FedAvg method was built to assess the ripeness and rawness of oil palm fruit images. This method utilizes federated learning with a privacy-preserving technique, ensuring data security and supporting the oil palm industry by developing predictive models without direct data exchange. The system accurately assesses palm bunch ripeness and displays results on a web application dashboard. This enhances ease of use, reduces errors in purchasing palm bunches, and ultimately increases the profitability of palm oil factories.

II. Literature Review

CNNs have been employed in various research contexts. For example, Chen *et al*. [6] studied crack detection on underwater metal surfaces in nuclear inspection videos, using CNNs trained on a dataset of patches containing both cracks and non-cracks, calculating scores for each patch independently. Performance was further enhanced by implementing a data fusion model that aggregates data from multiple video frames using spatial registration and Naive Bayes decision-making. Choi *et al*. [7] utilized CNNs to detect crack damage in structural images, classifying images into broken and unbroken categories. The results demonstrated the feasibility of replacing traditional IPA with a CNN model. Shi *et al*. [8] proposed a system for Autonomous Underwater Vehicles (AUVs) to detect pipe damage in image streams, achieving good results even with a limited number of damaged samples, by training discriminators using a CNN on random noise. Yang *et al*. [9] presented an Online Sequential Classifier and Transfer Learning (OSC-TL) method for online training and classification of Mura defects in flat panel displays. The combination of CNN feature extraction as a feature extractor and Online Sequential Extreme Learning Machine (OS-ELM) classification enables high classification accuracy and fast training and inference for Mura fault classification. Tang *et al*. [10] proposed a bilinear CNN model, a new spatial attention mechanism that combines the spatial attention mechanism and two-line gathering to enhance the performance power of CNN in detecting defects based on X-ray images. The pro- posed model demonstrates improved performance in X-ray testing through multiple experiments and detailed analysis of the experimental results. Ren *et al*. [11] present a general DL-based ASI method that includes feature transfer from a pretrained DL network and the conjugation of a patch-based classifier over the input image using a CNN or Decaf for analysis involving CNN images. Dong *et al*. [12] use a Fully Convolutional Network (FCN) called U-Net, which is a CNN architecture, as the main model for identifying pixels corresponding to abnormalities in the imaged region. It uses a set of sample images with common defects and the corresponding binary label images. Balzategui *et al*. [13] used a CNN to process cell images. The approaches require multiple runs to process a single image and obtain a segmentation map which be obtained in one step. This eliminates the need to perform multiple network operations. Rahman *et al*. [14] modified the U-net architecture, which is a CNN, to perform defect detection in photovoltaic Electroluminescence (EL) images. The modified U-net makes use of multiple attention networks, which consist of channel attention and spatial attention, to extract important features and suppress background noise. Han *et al*. [15] used CNN for defect segmentation in polycrystal silicon wafers, including feature extraction, event analysis, video understanding, and image segmentation. Fine-tuned specifically for the job. The contents have been summarized as shown in Table I.

From the previous research, CNNs have been widely used for image classification tasks, including agricultural applications such as fruit ripeness detection. While CNNs have demonstrated high accuracy in various studies, their application in the oil palm industry presents unique challenges that require further exploration. For example, oil palm fruits exhibit significant variability in size, shape, and color due to genetic differences, environmental factors, and ripeness levels. This variability makes it difficult for CNNs to generalize effectively across datasets. Images captured in natural settings are affected

by lighting variations, shadows, and background noise, which can degrade the performance of CNN models. Moreover, the manual collection and labeling of large datasets for training CNN models are time-consuming and prone to errors. Mislabeling or insufficient representation of ripeness levels can lead to model biases and reduced accuracy. CNNs also require substantial computational resources for training and deployment. Theses limitations are particularly relevant for oil palm mills operating in rural areas with constrained access to high-performance computing infrastructure. While CNNs process images locally, the centralized collection of data for model training raises privacy concerns, particularly for industrial applications where competitive data must remain confidential. To address these limitations, this study leverages the strengths of CNNs while incorporating Federated Learning (FL) to mitigate issues such as data centralization and privacy risks. The proposed PP-FedAvg model combines CNN capabilities with FL's privacy-preserving techniques, ensuring robust and secure predictions for the oil palm industry.

TABLE I. GAP ANALYSIS OF CNN-BASED CLASSIFIERS AND SEGMENTATION NETWORKS IN DEFECT DETECTION

Title	Technique	Strength	Gap analysis
Deep learning implemented structural detection on digital images [7]	-Vision-based methods using image processing algorithms (IPAs) for damage identification - Integration of deep learning (DL) models for image classification and segmentation - Utilization of convolutional neural networks (CNNs) for crack damage detection	- Utilizes deep learning models for structural defect detection in digital images, offering a more advanced approach than traditional methods. -Introduces a deep learning model for image classification and segmentation, showcasing faster, more robust, flexible, and intuitive results compared to existing methods. - Addresses the limitations of current vision-based methods by proposing advanced computer vision techniques using deep learning.	- A new method that is data privacy preserving combines a federated learning averaging and a convolutional neural network. (PP-FedAvg) - Data privacy with no raw data exchange and distributed operations across devices.
NB-CNN: Deep Learning-Based Crack Detection Using Convolutional Neural Network and Na¨ıve Bayes Data Fusion [6]	- Convolutional neural network (CNN) and a Na¨ıve Bayes data fusion scheme (NB-CNN) - Used Support Vector Machine (SVM) and Random Forest for classification tasks. - Recursive Feature Elimination (RFE) are utilized to enhance model performance - Cross-validation is applied to evaluate the models and ensure their generalizability	- It achieves higher hit rate than other approaches with fast operation speed. - Can even detect tiny cracks with low contrast and variant brightness that are hardly visible.	- A new method that is data privacy preserving combines a federated learning averaging and a convolutional neural network. (PP-FedAvg) - Data privacy with no raw data exchange and distributed operations across devices.
Automated underwater pipeline damage detection using neural nets [8]	- Transfer learning from pretrained convolutional neural networks (CNN) - Image pre-processing techniques from MobileNet	- This approach outperforms using VGG16. - The classifier that used MobileNet learned more relevant image features than the one one that used VGG16.	- A new method that is data privacy preserving combines a federated learning averaging and a convolutional neural network. (PP-FedAvg) - Data privacy with no raw data exchange and distributed operations across devices.
Transfer-learning-based online Mura defect classification [9]	- OSC-TL is a new method that combines a deep convolutional feature extractor and a sequential extreme learning machine classifier.	- The computational resources and time consumed by OSC-TL are well below those of other common methods - The proposed OSC-TL can achieve high classification accuracy for six kinds of Mura defects and high speed during training and inferencing. - The approach is suitable for real-time applications.	- A new method that is data privacy preserving combines a federated learning averaging and a convolutional neural network. (PP-FedAvg) - Data privacy with no raw data exchange and distributed operations across devices.
Nondestructive defect detection in castings by using spatial attention bilinear convolutional neural network [10]	- The spatial attention model and bilinear pooling were integrated into a CNN. (SA-BCNN)	- Could be trained in an end-to-end manner. - Extract subtle visual difference - Better detection performance compared to the original	- A new method that is data privacy preserving combines a federated learning averaging and a convolutional neural network. (PP-FedAvg) - Data privacy with no raw data exchange and distributed operations across devices.
Outlier detection Segmentation heterogeneity (Semantic Networks) [11–17]	- Fully Convolutional Networks, (FCNs); U-Net	- Quick training and assessment; detect and isolate the area with defects.	- Need for detailed human annotation at the pixel level; involves tedious and labor-intensive tasks.

To showcase prior studies concerning data security in PP-FedAvg, this work outlines the literature review in Table I. Lyu *et al*. [18] conducted a review of privacy issuesassociated with FL but did not provide or discuss privacy-preserving mechanisms that are specifically tailored for ensuring privacy protection within the context of FL. Li *et al*. [19] presented a brief summary of methods for maintaining privacy in FL, with a focus on strategies involving homomorphic encryption, secure multiparty computing, and differential privacy. Kaissis *et al*. [20] conducted an in-depthinvestigation into the utilization of FL alongside multiple privacy-preserving techniques within the domain of medicalimaging. Their research delved into the intersection of FL methodologies and privacy-preserving strategies, particularlyin the context of analyzing and safeguarding medical image data. Lim *et al*. [21] conducted an extensive survey that delved into the intricacies of FL as applied in mobile edge networks. Their analysis covered a range of critical factors including communication costs, resource allocation dynamics, as well as privacy and security considerations within thiscontext. Similarly, Li *et al*. [22] conducted a detailed survey focusing on FL's challenges and opportunities across different domains. They explored topics such as the high costs associated with communication in FL setups, the complexities arising from system and statistical heterogeneity, and the paramount importance of addressing privacy concerns in FL frameworks. Kairouz *et al*. [23] conducted a survey on FL, focusing on aspects such as efficiency, effectiveness, privacy, robustness, and fairness considerations. However, theiranalysis of privacy concerns primarily centered on external malicious actors and adversarial servers. Overall, the existingsurveys lacked a comprehensive assessment of the privacy preserving elements within FL frameworks. Yang *et al*. [24] presented an overview of FL, discussing distributed machine learning, privacy considerations, and categorization of FL architecture, FL design, and applications of FL. Their focus was particularly on categorizing FL systems and conductingsecurity analyses within this framework.

Based on the information provided earlier, as summarized in Table II, one can note the rapid evolution of FL alongside the limitations identified in current related surveys. This necessitates a thorough investigation. The objective of this survey is to confirm the latest findings, focus on existing gaps, and propose potential future research directions. In this work, the encryption techniques, anonymization methods, training steps, weights, and gradient descent will be explored, culminating in the development of the final model.

FL has emerged as a promising approach to collaborative model training without the need to share raw data, making it particularly suitable for privacy-sensitive applications. However, existing FL models such as FedAvg exhibit several shortcomings in addressing privacy concerns, particularly in industrial contexts. In traditional FL models, updated parameters from individual devices are sent to a central server. Although raw data is not shared, these updates can still be exploited through inference attacks to reconstruct sensitive information from individual datasets. The most FL implementations lack robust encryption or obfuscation mechanisms during data aggregation. This exposes the aggregated data to potential breaches if the central server is compromised.

Centralized aggregation creates a single point of failure, making it an attractive target for attackers. Compromising the central server could lead to the exposure of all model updates, undermining the privacy of all participating nodes. The existing FL models do not account for variations in the quality and quantity of data contributed by different nodes. This imbalance can make certain nodes more susceptible to inference attacks, especially if they provide unique or identifiable patterns. FL models have predominantly been tested in domains like healthcare and finance, where privacy concerns are critical, but their implementation in industrial applications, such as agriculture, has been limited. These applications often involve competitive data that must remain confidential, which is inadequately addressed by existing FL models.

The PP-FedAvg model overcomes these challenges by integrating privacy-preserving techniques with federated learning. PP-FedAvg employs encryption mechanisms to ensure that model updates remain secure during aggregation, minimizing the risk of inference attacks. The model avoids relying on a single centralized server, reducing the risk of a single point of failure. Instead of sharing full parameter updates, PP-FedAvg transmits only partial, representative data, further enhancing privacy. PP-FedAvg is specifically designed to address the unique requirements of the oil palm industry, balancing privacy concerns with high predictive accuracy for ripeness grading. By addressing these shortcomings, PP-FedAvg represents a significant advancement in privacy-preserving FL for industrial applications.

TABLE II. OVERVIEW OF RELEVANT FL THAT PROTECTS PRIVACY

Techniques to Safeguard Privacy Security Measures			Instances of Privacy Breach				References
Cryptographic Techniques	Perturbative Techniques	Anonymization Techniques	Internal/External Attackers	Training/ Inference Phase	Weight/Gradient/ The Final Model	Inference Attacks	
√	√		√	√		√	Lyu *et al*. [18]
√	√	√				√	Li *et al*. [19]
√	√					√	Kaissis *et al*. [20]
√	√						Lim *et al*. [21]
√	√				√		Li *et al*. [22]
√	√		√				Kairouz *et al*. [23]
√	√					√	Yang *et al*. [24]
√		√		√	√		Proposed method

Despite advancements in machine learning and FL, several critical gaps remain in their application to privacy-preserving agricultural systems, particularly for ripeness grading in the oil palm industry. Existing FL models like FedAvg are vulnerable to inference attacks and centralized aggregation risks due to insufficient encryption mechanisms, compromising data privacy. Scalability issues also arise, as traditional FL models struggle with heterogeneous datasets, limiting their effectiveness in diverse, real-world environments. Furthermore, FL applications in agriculture, especially for ripeness grading, are underexplored, often failing to address specific industry needs such as data confidentiality and accurate predictions under varying environmental conditions. Additionally, CNN-based solutions, while accurate, demand high computational resources, making them impractical for rural industries with limited infrastructure. Many prior methods focus either on improving accuracy or enhancing privacy, but rarely achieve a balance between the two—an essential requirement for industrial applications. The proposed PP-FedAvg model bridges these gaps by integrating encryption techniques with FL to enhance privacy, ensuring scalability through support for heterogeneous datasets, and tailoring its design for agricultural applications. It balances privacy and accuracy while maintaining resource efficiency, providing a comprehensive solution to the challenges faced by the oil palm industry.

III. Materials and Methods

A. The Proposed Framework

In this research, we propose a system that utilizes FL principles to classify the ripeness levels of oil palm fruit bunches. Information is distributed across various factories through a system known as FL-OPI, which serves as a communication framework for transferring models between nodes within thesystem, as illustrated in Fig. 1.

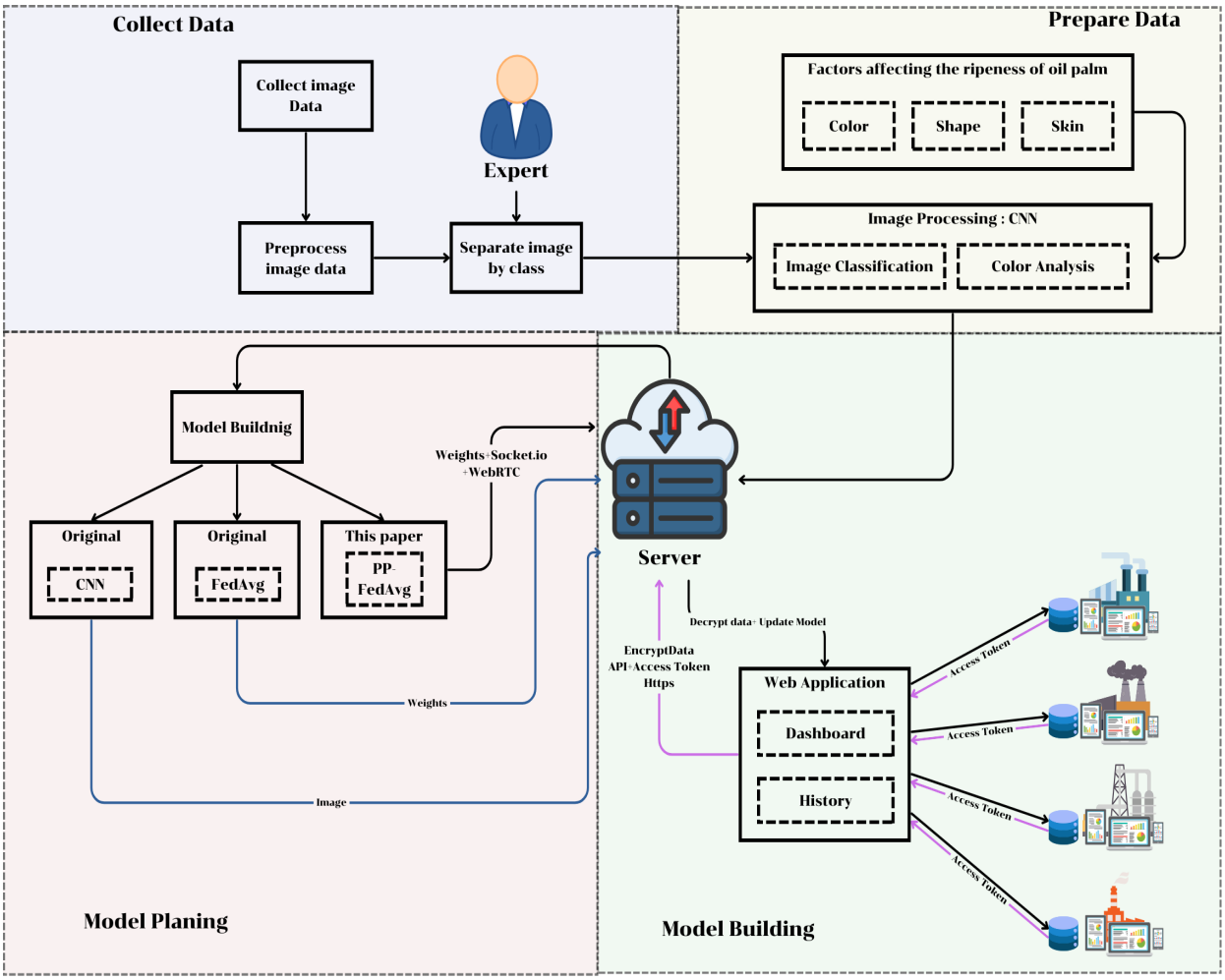

Fig. 1. The proposed framework for federated transfer learning.

Fig. 1 provides a comprehensive overview of the system, beginning with the data collection of palm oil fruit images from mill yards. The image data are preprocessed by resizingthe images to 224×224 pixels and classifying them based onthe color levels of the palm oil fruit bunches, as determined by experts. The classified images are then processed using a CNN to segment them according to ripeness levels, with a focus on the color of the palm fruit as a key factor indicatingthe ripeness of the bunches in data preparation step. The segmented data are uploaded to a

server for model development with model planning and model building in the next step.

During the model planning phase, the characteristics of models created using CNN, FedAvg, and the proposed model will be compared. In centralized training with CNN, image data are sent to the server for model training. In contrast, FedAvg, a technique in FL, does not send image data back to the server but only sends model weights (parameters) from the user's device. The strengths of both methods are combined to develop a new model called PP-FedAvg. In this work, the model training uses the FedAvg technique, which sends model weights and data through the Socket.io and WebRTC protocols for real-time communication to the server to update the central model. In this process, the model weights obtained from FedAvg, which preserve client data privacy, are selected and sent to the server. The server aggregates the model weights from multiple clients and averages them to create a new, more efficient central model. The updated model is then decrypted and deployed in the web application. Users can access the application via authentication using an access token and a secure HTTPS connection. The model data sent from each factory's device are transmitted back to the server for further training and updates. The connection through API, including access tokens and HTTPS, is encrypted to ensure the security and confidentiality of users' personal information. After several rounds of training and comparison of all models, the model with the best performance will be used in the web application.

B. Data Description

In this study, a dataset of 5,209 images of oil palm bunches from a factory in Surat Thani province, Thailand, was carefully compiled. This dataset is utilized for both model training and testing, with all 5,209 images designated for training and 753 images for testing. The dataset comprises 2,571 images of single bunches and 2,638 images of multiple bunches, categorized into four ripeness levels: raw, semi-raw, semi-ripe, and ripe. As detailed in Table III, the dataset includes four oil palm species: Searad 'Millennium' (Deli X Lame), Spring Black (Deli Nigerian Black), Surat Thani 2 (Deli X Lame), and Surat Thani 7 (Deli X Tanzania).

The dataset used in this study was divided into a training set (85%) and a test set (15%) to ensure a robust evaluation of the model's performance. This split ratio used in this study is a common practice in machine learning to ensure that the training set is sufficiently large for model training while still maintaining a robust test set for evaluation. This split ratio has been shown to offer a balanced approach for training deep learning models with adequate generalization performance. To achieve representative performance across all ripeness levels and varying environmental conditions, a stratified sampling approach was employed. This method preserved the proportion of each ripeness category (raw, semi-raw, semi-ripe, and ripe) within both the training and test sets. By maintaining this balance, the model was trained and tested on a dataset that accurately reflected the real-world distribution of ripeness levels.

Additionally, the dataset contained images captured under diverse environmental conditions, including varying lighting (natural sunlight, shaded areas, artificial lighting) and camera settings (angles, resolutions). These variations were equally distributed across the training and test sets to ensure that the model's performance was evaluated under conditions representative of actual deployment scenarios. This stratified and balanced split allowed the model to generalize effectively across all ripeness categories and environmental settings, demonstrating its robustness in practical applications.

TABLE III. DATASET OF OIL PALM FRUIT

Feature of fruit color	Level of ripness	Amount
black	raw	1943
light yellow-orange	semi-raw	1107
orange-red	semi-ripe	1062
dark brown	ripe	1097

To ensure consistency and improve model performance, several preprocessing steps were applied to standardize the dataset. Images were resized to a uniform resolution of 224×224 pixels to meet the input requirements of the CNN. Contrast enhancement techniques were employed to mitigate the effects of poor lighting, while image normalization was performed to scale pixel values between 0 and 1, reducing the impact of varying intensities. Augmentation techniques, such as rotation, flipping, and zooming, were also used to increase dataset diversity and improve the model's robustness against overfitting.

These preprocessing steps ensured that the dataset was standardized while preserving the inherent variability required for a realistic and robust training process. This approach enabled the proposed PP-FedAvg model to effectively learn features across a wide range of ripeness levels and environmental conditions.

C. The Implementation

This section aims to achieve application model in oil palm industry for fresh fruit bunch ripeness grading. It is divided into 2 parts, i.e., To find the best model of classification based on FL and to deploy the model. The first part is designed to compare the traditional models including CNN that CNN uses centralized training and has the ability to capture features from image data well. The model trained with CNN has high accuracy in image classification tasks but the resulting image data are sent to the server. This problem is generally solved by use of FedAvg method that provides data privacy without sending the raw data back to central server. It only sends the model weights (or model parameters) obtained from the user's device to the server. However, the gap is that this method still lacks data and model security during transmission. Thus, this work designs and implements enhanced data preservation and privacy.

To find the best model suitable for classifying the fresh fruit bunch ripeness in oil palm industry, the three model types CNN, FedAvg, and our work, will be compared. In the comparison, we will consider several factors: data privacy in data transmission, speed of data processing,

model quality, and the accuracy in classifying the ripeness levels of the purchased palm fruit bunches.

The accuracy will be assessed by examining the Loss value, which represents the error between predicted and truevalues, the ROC (Receiver Operating Characteristic) curve, which compares true and misclassified ratios, and AUROC (Area Under the ROC Curve), which summarizes the ROC curve into a single numerical value.

This work has combined the strengths of two methods, namely the advantages of CNN in image classification accuracy with the benefits of FedAvg in privacy preservation, resulting in a model that is not only highly accurate but also secure and effectively maintains data privacy. This is new model is called PP-FedAvg, and it selects FedAvg weights or model parameters that preserve privacy and sends them to the server. The PP-FedAvg model enhances the standard Federated Learning (FedAvg) algorithm by integrating privacy-preserving mechanisms to secure data during model updates. The process begins with the creation of an initial model, where parameters are trained and initialized for distribution to participating devices (clients). Each client then updates the model locally using its own data, ensuring that raw data never leaves the device. To safeguard the data, the updated model parameters are encrypted before being transmitted to the central server, preventing unauthorized access during communication. At the central server, the encrypted updates are collected, decrypted, and aggregated to produce an improved global model by averaging the contributions from all clients. This updated global model is then redistributed to the clients for further local training, and the cycle repeats until the desired model performance is achieved. By preventing raw data sharing and maintaining high model accuracy, PP-FedAvg offers a robust and privacy-preserving solution suitable for sensitive applications like ripeness grading in the oil palm industry. These FL processes are detailed in FedAVG, as shown in Algorithm 1.

Algorithm 1: FederatedAveraging (FedAvg)

1) Initialize w_0
2) **for** each round $t = 0, 1, \ldots$ do
3) $m \leftarrow \max(\lfloor C \cdot K \rfloor, 1)$
4) S_t = random set of m clients
5) **for** each client $k \in S_t$ in parallel do
6) $\lfloor w_{t+1}^k = ClientUpdate(k, w_t)$
7) $w_{t+1} = \sum_{k \epsilon s_t} \frac{n_k}{n_\sigma} w_{t+1}^k, where\ n_\sigma = \sum_{k \in s_t} n_k$

The operation of Algorithm 1 can be explained as follows:

(1) The collected dataset of oil palm fruit images is prepared for training the model, and the initial model parameters, denoted as w_0, are set for preliminary training.

(2) The process begins with the first round of training, indexed as $t = 0$, and continues through successive rounds until the training is complete.

(3) The model is trained using a neural network structure and the oil palm fruit image dataset to calculate the number of clients (i.e., web applications connected to each factory) that will participate in the training. The parameter m is determined based on the calculation as in Eq. (1).

$$m \leftarrow \max(\lfloor C \cdot K \rfloor, 1) \quad (1)$$

where C represents the fraction of clients expected to participate, and K is the total number of clients. If m is calculated to be less than 1, it is set to 1.

(4) A set of m clients (web applications) is randomly selected from the total pool, and this set is stored in thevariable S_t for the current training round.

(5) For each client k in the set S_t, perform local updates in parallel.

(6) Each client k performs a local update of the model, resulting in an updated model w_{t+1}^k. This update is performed using a function called ClientUpdate, which depends on the client's local data and the current global model w_t.

(7) For each round t, the current global model w_t is distributed to each client $k \epsilon s_t$. Each client then performs a local update, resulting in updated models w_{t+1}^k, using a function called ClientUpdate that takes into account theclient's local data and the current global model. These locallyupdated models are aggregated to compute the new global model w_{t+1} using a weighted average based on the amount ofdata n_k contributed by each client k. The new global modelis calculated as in Eq. (2).

$$w_{t+1} = \frac{1}{n_\sigma} \sum_{k \epsilon s_t} n_k \, w_{t+1}^k \quad (2)$$

where $n_\sigma = \sum_{k \in s_t} n_k$ represents the total weight across all clients in S_t. The updated global model w_{t+1} is then uploaded to the central system for use in the next training round. This process is repeated for subsequent rounds, with each iteration updating and aggregating the model parameters to progressively improve the global model's accuracy while ensuring that client data privacy is preserved.

In this paper, we introduce a privacy-preserving extensionto the original FedAvg algorithm. As detailed in Algorithm 2, this enhanced version incorporates privacy-preserving measures into the model updating process. The original FedAvg utilizes the *ClientUpdate*(), where each client locally updates its model using its own data. In contrast, the enhanced approach integrates these updates within the new *PrivacyPreservingClientUpdate*(). This function introduces additional privacy-preserving steps prior to the model update, thus ensuring that client data remains confidential throughout the training process as follows:

(1) For each client k in the set S_t, the loop executes in parallel, as indicated in line 9. Here, S_t represents the subset of clients selected at round t. The model update for each client is performed using the *PrivacyPreservingClientUpdate*(), which includes privacy-preserving steps in addition to updating the

model w_t locally. This process occurs within the loopdescribed in line 10.

(2) Before any local data *local_data*[*k*] for client *k* are sent to the server, they are first encrypted using the *EncryptData*(), as described in line 11. This ensures that client data *local_data*[*k*] remain confidential during trans- mission to the server. This step enhances the security and privacy of the model training process.

(3) The encrypted data from client *k* are transmitted to the central server using the *SendDataToServer*(), as detailed in line 12. The communication is secured using HTTPS (Hypertext Transfer Protocol Secure), and the API access is authenticated using an access token. This ensures that only authorized clients can transmit data. This step improves client-server communication security, which was not present in the original FedAvg.

(4) Upon receiving the encrypted data from client *k*, the server decrypts the data using the *DecryptData*(), as specified in line 13. The decryption key is generated during the encryption step, ensuring that only authorized clients can access the data. This step allows the client *k* to securely retrieve the processed data.

(5) After decrypting the data, the model for client *k*, $w_{k,t+1}$,is updated using the *UpdateModel*() with the decrypted data, as described in line 14. The updated model $w_{k,t+1}$, shown in line 15, is then returned to the server as the new local model for client *k*, as indicated in line 16. This step mirrors the model update in the original FedAvg but includesthe additional privacy-preserving steps.

Algorithm 2: Privacy-Preserving Federated Averaging (PP-FedAvg)

1) Initialize w_0
2) Initialize *local_data* for each client *k*
3) **for** each round $t = 0, 1, \ldots$ do
4) $m \leftarrow \max(\lfloor C \cdot K \rfloor, 1)$
5) S_t = random set of *m* clients
6) **for** each client $k \in S_t$ in parallel do
7) $\lfloor w_{t+1}^k = ClientUpdate(k, w_t)$
8) **Endfor**
9) **for** each client $k \in S_t$ in parallel do
10) function PrivacyPreservingClientUpdate(*k*, w_t):
11) *encrypted_data* = *EncryptData*(*local_data*[*k*])
12) *processed_data* = *SendDataToServer*(*encrypted_data*)
13) *decrypted_data* = *DecryptData*(*processed_data*)
14) *updated_model* = *UpdateModel* (w_t, *decrypted_data*)
15) $\lfloor w_{t+1}^k = UpdateModel(w_t$, *decrypted_data*)
16) **return** *updated_model*
17) **Endfunction**
18) **Endfor**
19) $w_{t+1} = \sum_{k \in S_t} \frac{n_k}{n_\sigma} w_{t+1}^k, where\ n_\sigma = \sum_{k \in S_t} n_k$
20) **Endfor**

To address the need for a pseudocode snippet, the encryption and decryption process in PP-FedAvg has been explicitly detailed to complement Algorithm 3. During the federated learning workflow, model updates are encrypted at the client side using a predefined encryption key before being transmitted to the central server. The server collects these encrypted updates, decrypts them using the same key, and aggregates the decrypted updates to form the updated global model. This updated model is then redistributed to the clients for further training. By incorporating this encryption and decryption mechanism, PP-FedAvg ensures the secure transmission of model parameters while maintaining the privacy of sensitive data throughout the federated learning process. The pseudocode simplifies the understanding of this critical step and aligns with the overall workflow of the proposed algorithm.

Algorithm 3: Encryption and Decryption in PP-FedAvg

Input: Model parameters (*W*), encryption key (*K*)
Output: Encrypted model parameters (*W_encrypted*), aggregated model (*W_global*)
Client Side:
1. Encrypt model updates:
W_encrypted = Encrypt(W, K)
2. Send *W_encrypted* to the server.
Server Side:
3. Collect encrypted updates from all clients:
W_encrypted_list = [W1_encrypted, W2_encrypted, ..., Wn_encrypted]
4. Decrypt updates:
W_decrypted_list = [Decrypt(W1_encrypted, K), Decrypt(W2_encrypted, K), ..., Decrypt(Wn_encrypted, K)]
5. Aggregate decrypted updates:
W_global = Aggregate(W_decrypted_list)
6. Send updated global model (*W_global*) to all clients.
Repeat the process until model convergence.

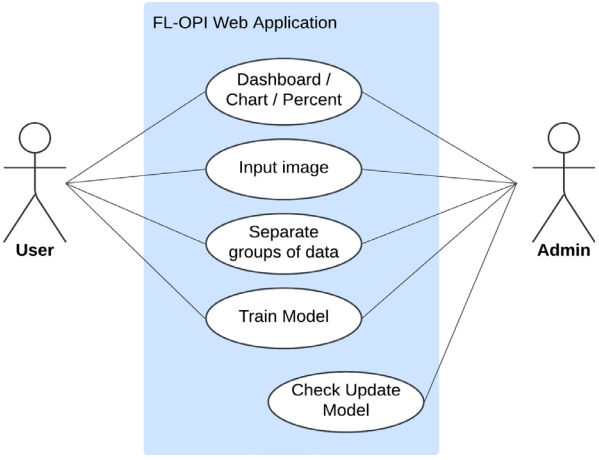

Fig. 2. Use case diagram for the web application part.

To deploy the model described above, the real-world application of the model through a web-based application will be designed and implemented. This implementation is designed for two main user groups: administrators and general users, as depicted in the use case diagram in Fig. 2. General users can select images of oil palm fruit bunches to assess their ripeness via the dashboard, as

shown in Fig. 3. The dashboard displays the percentage of ripeness at different levels, the time taken for the assessment, as shown in Fig. 4, and a feature for calculating the palm oil price based on the market's reference price. This calculation considers the actual weight of palm oil sold and the ripeness level from the assessment. Additionally, users can view their usage history.

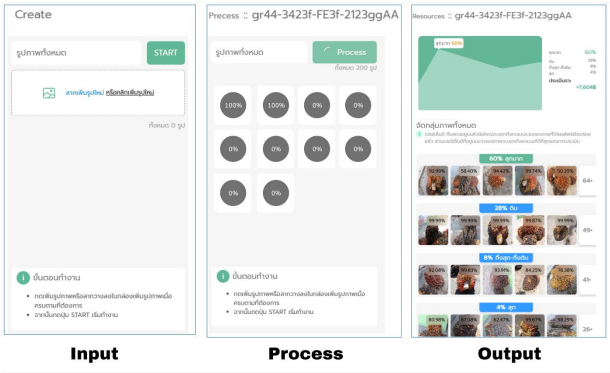

Fig. 3. Workflow on the application web page.

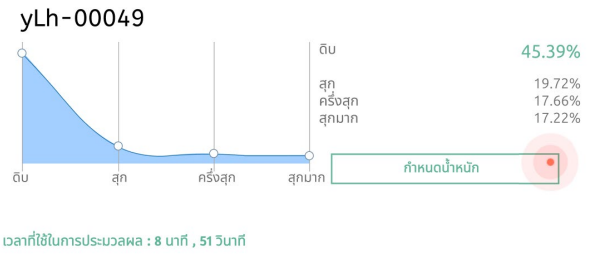

Fig. 4. Image processing time.

IV. Result and Discussion

To achieve the best model, the results of the comparison among CNN, FedAvg, and our approach are summarized in Table IV. Next, the best model was selected to be implemented and deployed in a web application.

TABLE IV. Comparison of CNN, FedAvg, and FedAvg + Privacy-Preserving

Characteristics	CNN	FedAvg	PP-FedAvg
Security (Data Privacy)	Low	Medium	High
Speed (mins)	5.07	8.00	8.51
Model Quality	Good	Good	Good
-Accuracy	0.8658	0.9870	0.9790
-Loss	0.0625	0.0526	0.0589
-ROC	1.0	0.8	0.9
-AUROC	1.0	0.9	0.9

Table IV visually represents the performance metrics, including Accuracy, Loss, ROC, and AUROC, for different methods—CNN, FedAvg, and PP-FedAvg, as illustrated in Fig. 5.

From Fig. 5, the performance comparison across methods—CNN, FedAvg, and PP-FedAvg—highlights the strengths and trade-offs of each approach in terms of accuracy, loss, ROC, and AUROC. FedAvg achieves the highest accuracy (98.7%) and lowest loss (0.0526), demonstrating strong predictive reliability and overall efficiency in model training. PP-FedAvg closely follows with an accuracy of 97.9% and a loss of 0.0589, showcasing its capability to maintain high performance while integrating privacy-preserving features. In contrast, CNN achieves a lower accuracy of 86.6% and a higher loss of 0.0625, suggesting that it may struggle more in reducing prediction errors compared to the federated approaches.

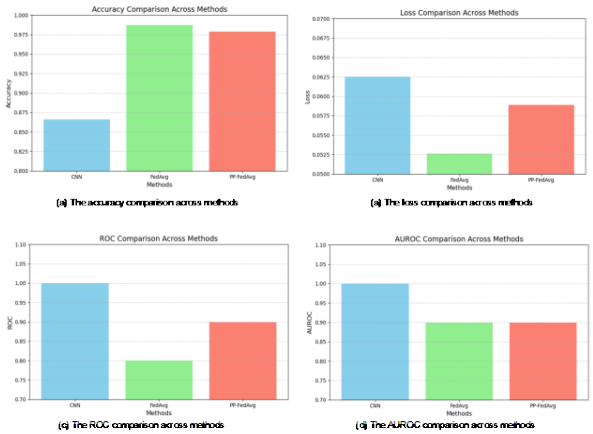

Fig. 5. The performance comparison for each method.

When considering ROC and AUROC, CNN achieves a perfect ROC score of 1.0 and an AUROC of 1.0, reflecting its excellent ability to distinguish between classes. PP-FedAvg maintains strong classification performance with an ROC of 0.9 and AUROC of 0.9, highlighting its effectiveness in balancing privacy and predictive accuracy. Meanwhile, FedAvg shows slightly reduced performance in class distinction with an ROC of 0.8, though it still maintains a competitive AUROC of 0.9.

To consider speed of model training, the slightly longer training time for PP-FedAvg compared to FedAvg is primarily attributed to the additional computational overhead introduced by its privacy-preserving measures. These measures, such as encrypting model updates and performing secure aggregation, require extra processing at both the client and server levels. While FedAvg transmits model updates without encryption, making it faster, this approach compromises data security. In contrast, PP-FedAvg ensures robust privacy by encrypting updates, which adds marginal computational time. This trade-off, however, is not significant when considering the overall benefits for industry adoption. The added training time is negligible compared to the critical need for protecting sensitive data in privacy-sensitive applications such as ripeness grading in the oil palm industry. Industries prioritize data confidentiality and compliance with privacy regulations, which makes PP-FedAvg's slightly longer training time an acceptable compromise. Furthermore, the scalability of PP-FedAvg ensures that the additional overhead remains manageable even as the number of clients increases, further supporting its feasibility for practical deployment in industrial settings.

In summary, while FedAvg demonstrates the best overall accuracy and loss metrics, PP-FedAvg provides a significant advantage by preserving privacy without

substantial sacrifices in classification performance. CNN, although achieving strong ROC and AUROC results, lags behind in accuracy and loss, indicating potential limitations in this dataset. Ultimately, PP-FedAvg emerges as a robust and balanced choice, offering high performance with enhanced privacy for applications like ripeness grading in the oil palm industry.

PP-FedAvg offers several advantages, making it a robust and practical solution for privacy-sensitive applications. By integrating encryption techniques into the federated learning framework, it ensures that raw data remains securely stored on local devices, significantly reducing the risk of data breaches. Despite its privacy-preserving focus, PP-FedAvg maintains high predictive accuracy, achieving a competitive accuracy of 97.9%, and strikes a balance between privacy and performance. It is highly scalable, enabling distributed model training across multiple devices or nodes without requiring centralized data aggregation. This scalability makes it particularly effective in handling heterogeneous datasets, which are common in real-world scenarios. Additionally, PP-FedAvg reduces the risk of single-point failure by minimizing reliance on a central server, enhancing system resilience. Its ability to adapt to diverse data distributions and process data locally makes it well-suited for industrial applications, such as ripeness grading in the oil palm industry, where data privacy and accurate predictions are critical. Furthermore, by distributing computational tasks across devices, PP-FedAvg reduces the need for high-performance centralized infrastructure, making it more feasible for deployment in resource-constrained environments. Overall, PP-FedAvg combines privacy, accuracy, scalability, and efficiency, making it an ideal choice for applications requiring secure and reliable predictions.

This is due to its use of encryption techniques and procedures to safeguard data privacy while updating the model centrally, as well as the absence ofphysical data exchange. This approach will prevent data theft during transit. In contrast, FedAvg offers moderate security because it lacks encryption when transmitting models to a central location. CNN, on the other hand, has low securityas it often relies on data at a single, potentially vulnerable source, which can expose personal data without adequate data privacy protection. In terms of speed, CNN is the fastestbecause it trains the model using data at a single source. In contrast, FedAvg and PP-FedAvg involve centralizing and processing data from multiple devices, which may slow down the process. Additionally, there could be minor differences in how PP-FedAvg utilizes the encoded model. Nonetheless, CNN, FedAvg, and PP-FedAvg all deliver high model quality, so the choice should be based on the specific intended use.

This study also evaluates the model's performance by examining accuracy and loss metrics to ensure the reliabilityof predictions. From Table IV, the PP-FedAvg is selected to deploy with the highest performance of all. The results showthat the developed model can predict the test dataset with a high degree of accuracy. However, when we train the modelwith a training dataset to convergence, we get the loss in the convergence dataset to 0.0002% per epoch, which is a relatively low prediction error for the model. As a result, themodel can effectively modify the parameters to fit with the training set data. This model is also selected as the initial global model.

Next, the global model is used by the client web application, which will automatically update with new information.As shown in Table IV, PP-FedAvg achieves high accuracyat 97.9%, close to FedAvg's 98.7%, while significantly enhancing data privacy (rated 'High' vs. FedAvg's 'Medium'). While it takes slightly longer to train, PP-FedAvg is the selected model for its combination of high accuracy and superior data protection, making it ideal for web applicationsthat prioritize both performance and privacy.

The performance of each model through the web application platform was compared in terms of accuracy, RMSE,MAE, and time, with actual usage data shown in Table V.

TABLE V. COMPARING THE PERFORMANCES OF EACH METHOD

Location	Model V.	No. Input	CNN				FedAvg				PP-FedAvg			
			Accuracy (%)	RMSE	MAE	Time	Accuracy (%)	RMSE	MAE	Time	Accuracy (%)	RMSE	MAE	Time
General	Default	5005	87	0.01	0.10	5:07	90	0.02	0.10	7:58	92.5	0.03	0.09	8:51
Industries 1	0004	102	50	0.03	0.12	0:10	93.5	0.03	0.10	0:07	93.5	0.04	0.12	0:09
Industries 2	0005	100	59	0.05	0.10	0:09	92.5	0.05	0.12	0:08	94.0	0.03	0.10	0:09
Industries 3	0006	99	75	0.04	0.09	0:08	95	0.01	0.09	0:07	96.5	0	0.10	0:10
Industries 4	0007	100	80	0.03	0.10	0:07	96	0.02	0.12	0:07	96.7	0.01	0.12	0:10

In the comparative analysis shown in Fig. 6, an accuracy higher than 90% is considered very good. PP-FedAvg achieved the highest accuracy at 96.7%, followed by FedAvg at 96%, and CNN at 80%. The Root Mean Square Error (RMSE), which represents the difference between predicted and actual values, did not show significant variation across models, as indicated in Fig. 7. PP-FedAvg had RMSE values ranging from 0.01 to 0.04, while CNN and FedAvg had values ranging from 0.01 to 0.05. These values are considered satisfactory, as a typical RMSE should be less than 0.1. The Mean Absolute Error (MAE), which representsthe average absolute difference between predicted and actual values, ranged from 0.09 to 0.12 across all models. This is considered a favorable result, as it falls within the acceptablerange of 0.1 to 0.2 for MAE, as shown in Fig. 8. In terms of computation time, CNN completed the task in the shortesttime among all models. FedAvg took slightly less time than PP-FedAvg, as demonstrated in Fig. 9.

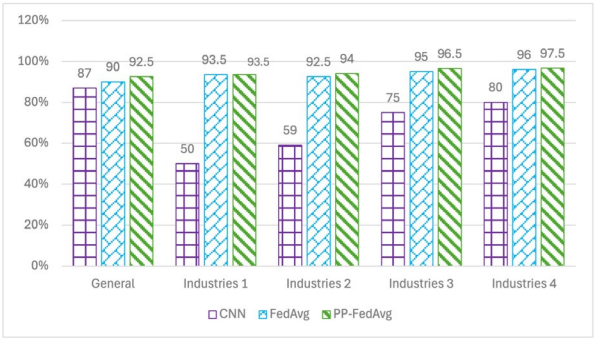

Fig. 6. Accuracy comparison of the methods.

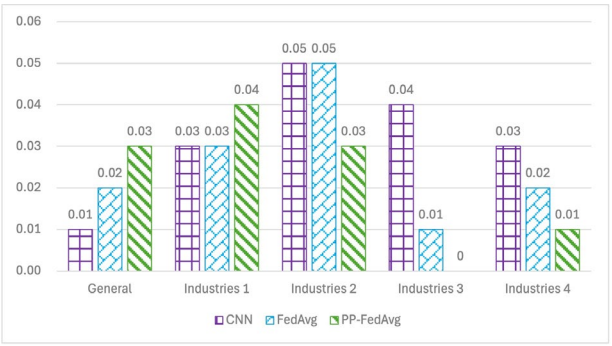

Fig. 7. RMSE of each method.

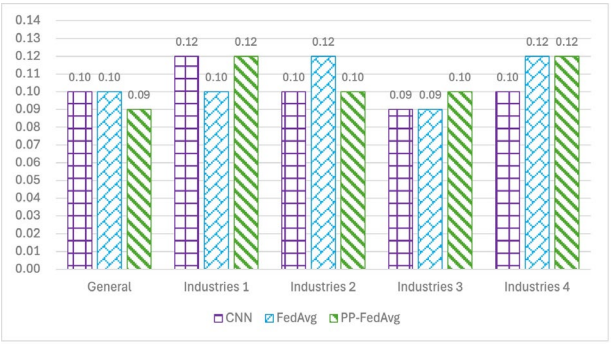

Fig. 8. MAE of each method.

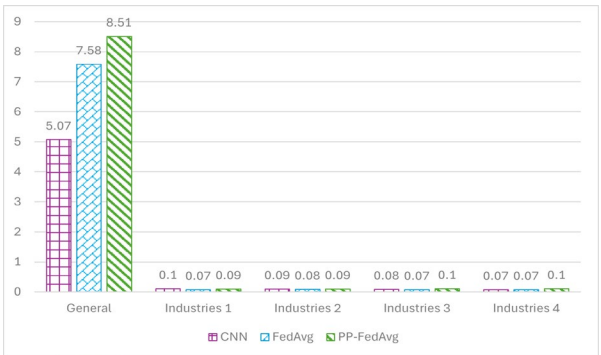

Fig. 9. Time of each method.

Therefore, from the above comparative analysis, PP-FedAvg achieved the best results in classification accuracy, measuring the prediction error using the RMSE, MAE, and prediction time of the system. This study had different work processes compared to the prior work as summarized in Table VI.

TABLE VI. A COMPARISON OF THE ORIGINAL SYSTEM AND THE PROPOSED SYSTEM WITH FL-OPI

System/ Method	Input	Process	Output
Original	Single file/time	- Countdown timer for processing	- Show percentages separated by class
FL-OPI	Multiple files/time	- Notify the queue order in processing. - Shows total processing time.	- Graph showing overall percentage. - Group imported data by class. - Shows the total percentage of the class, and percentage of each file within the class. - Summary of results, arranged in order of the percentage of data that has a lot to the least. - Show input leader information.

In this study, an FL framework was utilized to update a global machine learning model without using raw data, making it available to all clients for predictive categorization. Clients could use the global model to update their local models by retraining with private data. Data are encrypted and sent back to the server securely, where the are storedin encrypted files. The system ensures safety through encrypted passwords and authentication token login sessions. The global model is then updated to enhance its functionality before being redistributed to clients. This approach not onlyensures data privacy but also improves the accuracy of predictive models. Moreover, the method supports the creation of a cross-platform web application that users can access from any device or location.

In terms of accuracy evaluation, conventional models are developed using data that have been gathered and directly assessed by specialists. Training and test accuracy values areused to assess the proposed system's correctness. The modelsare fine-tuned by adjusting hyperparameters such as learningrate, batch size, epoch count, and model structure. This iterative enhancement, based on accuracy indicators, improves prediction efficiency. The system effectively determines theripeness stages of oil palm fruit bunches, displaying processing times and percentages for each ripeness stage.

The newly developed model was integrated into a web application, providing users with an intuitive interface for seamless interaction. The system predicts the ripeness level of oil palm fruit bunches and presents results as percentages for each ripeness stage, accompanied by graphical visualizations on the web dashboard. Users can also access historical interactions and usage records, enhancing the overall user experience.

PP-FedAvg delivers significant business benefits due to its high accuracy and robust privacy measures. Its enhanced accuracy in ripeness grading minimizes classification errors, ensuring that palm oil mills pay fair prices based on the actual oil content of Fresh Fruit Bunches (FFB). This leads to greater efficiency in oil extraction, reduces wastage, and ultimately boosts profitability. Additionally, the privacy-preserving features of PP-FedAvg safeguard sensitive operational data, mitigating the risk of data breaches and ensuring

compliance with privacy regulations. This fosters trust among stakeholders, including farmers, mill operators, and regulatory bodies, strengthening business relationships. Furthermore, the secure management of data enhances a company's reputation, positioning it as a leader in adopting innovative and ethical technologies. By striking a balance between performance and privacy, PP-FedAvg not only optimizes operational efficiency but also supports long-term sustainability and competitiveness in the palm oil industry.

Following the evaluation of the PP-FedAvg model, a web application was developed featuring a user-friendly dashboard. This dashboard allows users to interact with the model for assessing the ripeness of oil palm fruit. It provides real-time predictions of ripeness levels, presented both as percentages for each ripeness stage and as visual graphs for easier interpretation. The dashboard also includes features to review historical data, allowing users to track previous assessments and trends over time. This feature supports better decision-making and operational planning for mills. Furthermore, the application is designed with user-friendliness in mind, offering clear navigation and responsive performance across various devices.

Preliminary user feedback highlighted the ease of use and practicality of the dashboard. Users appreciated the graphical representation of results, which simplified complex data into actionable insights. However, some users suggested enhancements such as more customizable settings for specific use cases, which have been noted for future development. Users also recommend that it be applied and developed in conjunction with the automation for embedded systems. Overall, the web application dashboard not only enhances accessibility and usability but also reinforces the practicality of the PP-FedAvg model in real-world industrial applications.

The deployment of the PP-FedAvg model into the web application presented some challenges, particularly related to latency and server compatibility. During the initial deployment, latency was observed when handling real-time data processing, especially as the number of connected client devices increased. This issue was mitigated by optimizing the server's workload distribution and implementing asynchronous data processing techniques, which ensured smoother real-time interactions. Additionally, ensuring compatibility across different server environments posed challenges due to variations in operating systems, hardware configurations, and network setups. These were addressed by containerizing the application using Docker, which provided a consistent deployment environment and simplified the integration process across diverse server infrastructures. Despite these challenges, the deployment was successful after the optimizations, resulting in a robust and scalable system that supports real-time data processing and privacy-preserving operations. These insights highlight the importance of addressing technical obstacles during deployment to ensure the system's efficiency and reliability in industrial settings.

V. Conclusion

This research compared the modeling techniques CNN, FedAvg, and PP-FedAvg. The researchers selected PP-FedAvg to develop a prediction model with centralized functionality that can be shared without direct data exchange. PP-FedAvg employs Privacy-Preserving techniques to ensure data security against unauthorized events. This study utilizeda total of 5209 images of oil palm bunches, comprising 2571 single bunch images and 2638 multiple bunch images, takenboth indoors and outdoors. All images were categorized intofour maturity levels: raw, semi-raw, semi-ripe, and ripe, withan additional 15% of the images used for testing the local model. After training the model, it was found that the color of oil palm bunches significantly affects the oil yield, with each maturity level providing different quantities of oil. Theprediction results are displayed in a web application dash- board for user convenience. The developed model achieved an accuracy of 92.5%. One key limitation is the scalability of the proposed PP-FedAvg model across different regions. While the model demonstrates robust performance on the dataset collected from a single location, its generalizability to other regions with different environmental conditions, palm fruit species, and agricultural practices requires further investigation. Additionally, the dataset used in this study may contain inherent biases, such as imbalanced representation of ripeness levels and variations in image quality, which could affect the model's predictive accuracy.

Future work will focus on enhancing the PP-FedAvg model by integrating advanced privacy-preserving techniques, such as differential privacy, to ensure individual data anonymity while safeguarding against inference attacks. This enhancement will maintain the model's scalability and efficiency while expanding its applicability to a broader range of agricultural products, including bananas, mangoes, and tomatoes, which face similar challenges related to data privacy and predictive accuracy. Additionally, future iterations of the model could incorporate additional factors into the analysis, such as nutrient levels (e.g., nitrogen, phosphorus, potassium, magnesium, and boron), environmental conditions (e.g., soil type, weather, and soil moisture), and physiological measurements (e.g., bunch weight, frond production, canopy dimensions, and leaf area) to further improve the oil yield prediction model's accuracy and comprehensiveness. To address the reviewer's comment, the exclusion of additional parameters such as nutrient levels (e.g., nitrogen, phosphorus, potassium, magnesium, and boron), environmental factors (e.g., soil type, weather, and soil moisture), and physiological measurements (e.g., bunch weight, frond production, canopy dimensions, and leaf area) in the oil yield prediction model is acknowledged and justified in the manuscript. Efforts will also focus on expanding the dataset to include diverse regions and conditions, utilizing techniques like stratified sampling and advanced augmentation to mitigate potential biases. These initiatives aim to refine the model's robustness, adaptability to various environmental and domain-specific features, and compliance with industry

standards for data privacy and accuracy, ensuring its broader applicability in agricultural and industrial contexts.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

PL performed data analysis and results visualization and wrote the manuscript. SJ, JM, TP and YP collected the data and made contributions to data analysis. JM, PC, and SJ made significant contributions to the conceptualization, methodology, data analysis, supervision, validation, and writing—review and editing of the study. All authors read and approved the final manuscript. Correspondence to Jirapond Muangrathub.

FUNDING

This work was supported by the Digital Science for Economy, Society, Human Resources Innovative Development and Environment project funded by Reinventing Universities & Research Institutes under grant no. 2046735, Ministry of Higher Education, Science, Research and Innovation, Thailand. In addition, this research was supported byNational Science, Research and Innovation Fund (NSRF) and Prince of Songkla University Ref. No. SIT6701364S.

ACKNOWLEDGMENT

The authors are deeply grateful to the Faculty of Science and Industrial Technology, Prince of Songkla University, Surat Thani campus, Thailand.

REFERENCES

[1] D. N. Arulnathan, B. C. W. Koay, W. K. Lai *et al.*, "Background subtraction for accurate palm oil fruitlet ripeness detection," in *Proc. IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*, June 2022, pp. 48–53.

[2] N. Kumari, R. K. Dwivedi, A. K. Bhatt, and R. Belwal, "Automated fruit grading using optimal feature selection and hybrid classification by self-adaptive chicken swarm optimization: Grading of mango," *Neural Computing and Applications*, vol. 34, pp. 1285–1306, January 2022.

[3] B. Xu, X. Cui, W. Ji *et al.*, "Apple grading method design and implementation for automatic grader based on improved YOLOv5," *Agriculture*, vol. 13, no. 1, Jan 2023.

[4] M. S. M. Alfatni, S. Khairunniza-Bejo, M. H. B. Marhaban *et al.*, "Towards a real-time oil palm fruit maturity system using supervised classifiers based on feature analysis," *Agriculture*, vol. 12, no. 9, 2022.

[5] R. Taheri, M. Shojafar, M. Alazab, and R. Tafazolli, "Fed-IIot: A robust federated malware detection architecture in industrial Iot," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 12, pp. 8442–8452, December 2021.

[6] F.-C. Chen and M. R. Jahanshahi "Nb-CNN: Deep learning-based crack detection using convolutional neural network and naïve bayes data fusion,"*IEEE Transactions on Industrial Electronics*, vol. 65, no. 5, pp. 4392–4400, May 2018.

[7] W. Choi, "Deep learning implemented structural defect detection on digital images," Ph.D. dissertation, Dept. Civil. Engineering., University of Manitoba., Winnipeg, Manitoba, Canada, 2020.

[8] J. Shi, W. Yin, Y. Du *et al.*, "Automated underwater pipeline damage detection using neural nets," in *Proc. ICRA 2019, Workshop on Under-water Robotics Perception*, 2019.

[9] H. Yang, S. Mei, K. Song *et al.*, "Transfer-learning-based online mura defect classification," *IEEE Transactions on Semiconductor Manufacturing*, vol. 31, no. 1, pp. 116–123, February 2018.

[10] Z. Tang, E. Tian, Y. Wang *et al.*, "Nondestructive defectdetection in castings by using spatial attention bilinear convolutional neural network," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 1, pp. 82–89, Jan 2021.

[11] R. Ren, T. Hung, and K. C. Tan, "A generic deep-learning-based approachfor automated surface inspection," *IEEE Transactions on Cybernetics*, vol. 48, no. 3, pp. 929–940, March 2018.

[12] X. Dong, C. J. Taylor, and T. F. Cootes, "Small defect detection using convolutional neural network features and random forests," in *Proc. European Conference on Computer Vision (ECCV) Workshops*, 2018, pp. 398–412.

[13] J. Balzategui, L. Eciolaza, and N. Arana-Arexolaleiba, "Defect detection on polycrystalline solar cells using electroluminescence and fully convolutional neural networks," in *Proc. 2020 IEEE/SICE International Symposium on System Integration (SII)*, 2020, pp. 949–953.

[14] M. R. U. Rahman and H. Chen, "Defects inspection in polycrystalline solar cells electroluminescence images using deep learning," *IEEE Access*, vol. 8, pp. 40547–40558, 2020.

[15] H. Han, C. Gao, Y. Zhao *et al.*, "Polycrystalline silicon wafer defect segmentation based on deep convolutional neural networks," *Pattern Recognition Letters*, vol. 130, pp. 234–241, 2020.

[16] Z. He and Q. Liu, "Deep regression neural network for industrial surface defect detection," *IEEE Access*, vol. 8, pp. 35583–35591, 2020.

[17] S. Zambal, C. Heindl, C. Eitzinger, and J. Scharinger, "End-to-end defect detection in automated fiber placement based on artificially generated data." in *Proc. Fourteenth International Conference on Quality Control by Artificial Vision.* 2019, 68.

[18] L. Lyu, H. Yu, and Q. Yang, "Threats to federated learning: A survey," arXiv preprint, arXiv:2003.02133, 2020.

[19] Z. Li, V. Sharma, and S. P. Mohanty, "Preserving data privacy via federated learning: Challenges and solutions," *IEEE Consumer Electronics Magazine,* vol. 9, no. 3, pp. 8–16, May 2020.

[20] G. A. Kaissis, M. R. Makowski, D. Rückert *et al.*, "Secure, privacy-preserving and federated machine learning in medical imaging," *Nature Machine Intelligence*, vol. 2, no. 6, pp. 305–311, 2020.

[21] W. Y. B. Lim, N. C. Luong, D. T. Hoang *et al.*, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.

[22] Q. Li, Z. Wen, Z. Wu *et al.*, "A surveyon federated learning systems: Vision, hype and reality for data privacy and protection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 4, pp. 3347–3366, 2021.

[23] P. Kairouz, S. Oh, and P. Viswanath, "The composition theorem for differential privacy," in *Proc. International Conference on Machine Learning*, 2015, pp. 1376–1385.

[24] Y. Qian, L. Hu, J. Chen *et al.*, "Privacy-aware service placement for mobile edge computing via federated learning," *Information Sciences*, vol. 505, pp. 562–570, 2019.