Lightweight Federated Transfer Learning for Plant Leaf Disease Detection and Classification across Multiclient Cross-Silo Datasets

Shilpi Choubey*, and Divya

Faculty of CS & IT, Kalinga University, Naya Raipur, Chhattisgarh, India

Abstract. Plant leaves and crops play a crucial role as a primary food source globally, making significant contributions to dietary iron intake (9%) and energy consumption (23%) per capita in the Asian region. Bacterial, yeast, and other microbial diseases pose significant challenges to farmers as they detrimentally impact plant health and reduce crop productivity. The manual diagnosis of these diseases poses a considerable challenge, particularly in regions with a scarcity of professionals specializing in leaves and crop protection. Automating leaf disease detection and providing easily accessible decision-support resources are crucial for facilitating efficient leaf protection strategies and mitigating crop damage. Despite multiple classification methods for diagnosing leaf diseases, a secure and accurate approach that fulfills these requirements has not yet been identified. This paper presents an architectural framework called Lightweight Federated Transfer Learning (LFTL) that addresses the challenge of Leaf Disease Detection and Classification (LDDC) while ensuring data privacy limitations are upheld. A dataset consisting of leaf disease images has been compiled, characterized by an imbalance in the distribution of the diseases. The collection includes four conditions: bacterial decay, brown spot, blast, and tungro, corresponding image counts of 1695, 1551, 1711, and 1419, respectively. Following the preprocessing stage, the LFTL framework was tested using both Independent and Identically Distributed (IID) and non-IID datasets. The study commenced with an efficacy evaluation of the Convolutional Neural Network (CNN) and eight TL models in the LDDC. The framework's performance was evaluated across different circumstances and compared to conventional and federated learning models. The study's findings revealed that the LFTL framework outperformed traditional distributed deep-learning classifiers, thus demonstrating its efficacy in individual and multiple client scenarios.

1 Introduction to Plant Leaf Disease Detection

Plant Leaf Disease Detection (PLDD) is prominent in agricultural advancements and crucial in safeguarding worldwide food security [1]. The health of plant leaves and crops directly impacts food patterns and energy use since they are vital sources of nourishment. It is worth

* Corresponding author: ku.shilpichoubey@kalingauniversity.ac.in

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

mentioning that in Asia, plant-based sources account for around 9% of the overall iron intake in the diet and approximately 23% of the per capita energy consumption. The preservation of agricultural output is persistently jeopardized by bacterial, yeast, and microbial illnesses [2]. Developing resilient detection techniques to address this significant concern in food production is essential.

The need to implement efficient PLDD methods is driven by the urgency to provide farmers with effective instruments for controlling and minimizing the consequences of plant leaf diseases [3]. Conventional diagnosis techniques, which heavily rely on visual examination conducted by agricultural specialists, have significant obstacles, especially in areas needing more such personnel [4]. The manual identification process is characterized by considerable work and time investment, leading to delays in identifying diseases and implementing mitigation plans. The magnitude of contemporary agricultural enterprises necessitates using scalable and highly technological measures to effectively and expeditiously tackle issues about plant health [5].

Several obstacles impede the effectiveness of classic PLDD approaches. The disparity in the distribution of illnesses across crops, characterized by the varying prevalence of certain diseases, poses challenges in achieving precise categorization [6]. The human labor involved in classifying and diagnosing conditions creates a bottleneck, increasing the need for automated solutions that rely on data analysis. Conventional techniques often need to improve the accuracy required by contemporary agricultural practices, resulting in challenges in promptly detecting subtle indications of illnesses during their first phases.

The significance of the economic consequences of crop failure caused by illnesses highlights the pressing need to tackle these issues. Crop diseases lead to a significant reduction of around 40% in the global food supply every year [8]. This highlights the need for enhanced PLDD techniques to provide precise and prompt outcomes. In the current scenario, incorporating advanced technologies, such as Lightweight Federated Transfer Learning (LFTL), offers a potential opportunity to augment PLDD. The LFTL approach effectively tackles the drawbacks associated with conventional methodologies by using sophisticated machine learning while upholding the privacy considerations inherent in agricultural information.

The primary contributions are

- Privacy-Preserving Framework: Sensitive agricultural data is protected by LFTL, which efficiently addresses LDDC concerns while maintaining data privacy.
- Efficacy Compared to Conventional Models: LFTL performs better than conventional distributed Deep Learning (DL) classifiers for accurate leaf disease identification in single and many client situations.
- Architectural Innovation: The suggested architecture and Python libraries like Keras and TensorFlow introduce a unique Federated Learning (FL), which reduces communication overhead and improves data privacy in categorizing plant leaf diseases.

The forthcoming sections are as follows: The current state of plant leaf disease detection in the literature review is covered in Section 2. Section 3 proposes the LFTL framework—which prioritizes effectiveness and privacy—to solve leaf disease detection problems. Section 4 presents an experimental study and the results and insights into the performance of LFTL in plant leaf disease classification. The study's conclusion is provided in Section 5, which also highlights significant results and suggests future options for developing and using the proposed framework.

2 Literature Survey and Analysis

The literature review presents a thorough analysis of the current state of Plant Leaf Disease Detection, including a wide range of strategies, techniques, and models used for identifying diseases in agricultural environments. The survey examines significant accomplishments, problems, and trends in the subject, establishing the foundation for the creative contributions discussed in later parts.

Vallabhajosyula et al. proposed a novel approach called the Transfer Learning-based Deep Ensemble Neural Network (TL-DENN) to address the task of plant leaf disease identification [8]. The TL-DENN approach utilizes transfer learning and ensemble methods to enhance accuracy using pre-trained models' capabilities. The usefulness of the suggested strategy in robust illness detection was proved by experimental findings on varied datasets, which achieved a notable accuracy of 94.5%. Chouhan et al. proposed a novel approach called Automated Plant Leaf Disease Detection and Classification using a Fuzzy-Based Function Network [9]. The use of fuzzy logic improves the accuracy of categorization. The experimental assessments demonstrated an accuracy rate of 92.3%, highlighting the efficacy of fuzzy-based methodologies in automating PLDD and categorization.

Tiwari et al. proposed a PLDD system that utilizes Dense Convolutional Neural Networks (DCNN) for multiclass classification [10]. The suggested approach uses the deep characteristics extracted from convolutional networks to achieve precise categorization. The experimental results indicated a significant level of accuracy, namely 96.2%, highlighting the efficacy of DCNN in diagnosing plant diseases across many classes utilizing photos of leaves. The research conducted by Zhao et al. made a significant contribution to the field of PLDD by proposing a novel strategy that utilizes artificially produced leaves via the implementation of Double Generative Adversarial Networks (DoubleGAN) [11]. The DoubleGAN model effectively creates artificial leaf samples to address the challenge of the limited availability of training data. The experimental findings demonstrated a level of accuracy, namely 93.8%, hence emphasizing the potential of synthetic data in augmenting the resilience of PLDD.

Deepa et al. proposed a novel approach called the Kuan Noise Filter with Hough Transformation-based Reweighted Linear Program Boost Classification to detect plant leaf diseases [12]. The proposed methodology integrates noise filtering, Hough transformation, and reweighted linear program boosting techniques to improve the accuracy of classification tasks. The experimental findings exhibited an accuracy, namely 95.7%, highlighting the suggested approach's efficacy in identifying plant leaf diseases. Mahum et al. proposed a unique Potato Leaf Disease Detection framework using an Efficient DL Model [13]. The system utilizes a very effective DL, demonstrating progress in potato disease diagnosis. The experimental evaluations yielded results indicating a high accuracy rate of 97.2%. This highlights the resilience of the DL model presented to effectively and efficiently detect diseases in potato leaves.

Gajjar et al. proposed a method for the real-time detection and identification of plant leaf diseases [14]. This method utilizes Convolutional Neural Networks (CNNs) and is implemented on an embedded platform. The approach employs CNNs to diagnose diseases in real time, showcasing the practicality of incorporating these networks into embedded systems. The experimental results demonstrated a real-time accuracy rate of 91.8%, emphasizing the effectiveness of the CNNs technique in plant leaf disease diagnosis. The research conducted by Vishnoi et al. made a significant contribution to the field of PLDD using Computational Intelligence and Image Processing [15]. The suggested methodology combines artificial intelligence and image processing methods to achieve precise illness detection. The experimental findings demonstrated a level of accuracy reaching 94.5%. This highlights the effectiveness of using computational intelligence techniques to detect plant diseases.

The literature review presents a comprehensive analysis of several PLDD methodologies. It elucidates the progress made in this area, including utilizing transfer learning, fuzzy logic, deep convolutional networks, and generative adversarial networks. Obstacles still need to be addressed in PLDD [16]. These problems include the limited availability of data, the need for immediate detection capabilities, and the demand for resilient models capable of effectively handling a wide range of PLDD [17].

3 Proposed Lightweight Federated Transfer Learning Method

This section presents the LFTL framework for classifying Plant Leaf Disease. The LFTL system has a client-server architecture using Python modules such as Keras and TensorFlow. The approach prioritizes data privacy protection by transferring learned models between clients and servers, minimizing communication overhead. This section provides an overview of the three primary stages of the FL process: global model initialization, local model training, and aggregation of local model updates.

3.1 FL for Images Classification

The requirement for categorizing photos depicting plant leaf diseases emerges due to the imperative to protect crops, improve agricultural production, and assure food security. By using technological advancements to create precise and streamlined detection techniques, farmers have access to the essential resources required for proficiently overseeing and minimizing the consequences of plant leaf illnesses. Farmers acquire and compile confidential data about their agricultural produce, including visual records of diseased plant foliage, geographical coordinates, and cultivation methodologies.

Fig. 1. The architecture of the proposed LFTL system

The dataset potentially includes proprietary material or Personally Identifiable Information (PII) necessitating safeguarding. This study component implements and analyzes FL to classify plant leaf diseases. The architecture of the proposed LFTL is described in Figure 1. The plant leaf illnesses categories' properties were retrieved using several pre-trained algorithms. A comparative analysis evaluated several trained designs, focusing on their training and validation precision. The results indicated that EfficientNetB3

exhibited superior performance compared to the other systems. One of the issues encountered while implementing an FL system pertains to the need to effectively execute the learning model on edge gadgets, which serve as the monitors for each respective zone. The learning model must possess a lightweight design to function on these devices effectively. The algorithm's light characteristic is of utmost importance when considering the layout of the FL system. The system chooses to use a dense neural network with a 3-layer architecture for classification instead of an alternative, computationally intensive model that requires extensive training. Each neuron in a given layer is connected to each neuron in that layer directly underneath it, facilitating the propagation of computation and information over numerous layers. The selection of the standard network was based on its superior overall performance and successful implementation.

Fig. 2. FL-based classification model

The suggested plant leaf illness categorization system used the client-server design via FL, as seen in Figure 2. Python programming archives, including Keras and TensorFlow, facilitated the system's development. Each network client was equipped with an individualized processing and storage subsystem responsible for managing their structured data and generating model training and parameters—the server, at a distance, evaluated the federated universal weights of the clients' activity that were trained. The suggested FL mitigates connection cost and data security issues by only sending learned models among the client and server. In FL, the overall process typically consists of three main steps.

1) Centralized server

During the first phase, a centralized server defines the training assignment and specifies the intended use. The server generates the initial universal modeling via the specification of hyperparameters and the determination of the objective operation, denoted as $J(0, L_x)$, for each client x. The server distributes the loaded global modeling to the selected local attendees

involved in the training process. The function of objectiveness is mathematically represented in Equation (1).

$$J(0, L_x) = \sum_{x=0}^{N-1} D(f(\theta, i), j) \quad (1)$$

The universal model variables are denoted by 0, whereas L_x represents the local database for every customer. The objective function measures the discrepancy between the projected outputs of the algorithm and the actual labels associated with the localized dataset—the function $f(\theta, i)$ denotes the forecast made by the model for a given input, using the universal model variables 0. The variable j signifies the actual label. The symbol D represents the loss function.

2) Local model training

During the FL management, each client in the network maintains its distinct dataset. The clients use their datasets to make local revisions to the models. In each iteration of the training process, a subset of customers or gadgets, denoted as $C_x$where x ranges from 1 to N, is chosen. N represents the total number of clients picked for that round. To ensure the accuracy and effectiveness of the analysis, it is essential to conduct local training of models for each chosen customer, denoted as C_x, by using their local database, referred to as L_x. The process entails the optimization of the local modeling variables to minimize the objective function J(θ_x, L_x)on behalf of the customer. This objective is accomplished using iterative optimizing methods, such as Standard Gradient Descent (SGD), denoted in Equation (2).

$$\theta_x(i+1) = \theta_x(i) - \alpha \Delta \mathrm{J}(\theta_x, L_x) \quad (2)$$

The symbol α indicates the learning speed, the variable x signifies the repetition or epoch, and the symbol ∇J indicates the slope of the goal function about the client's localized model variables θ_x.

3) Aggregation

After training at the local level, the model specific to the region is enhanced by aggregating modifications received from the chosen customers. This process results in the acquisition of an upgraded global modeling. This is accomplished by weighted averages or plain averaging, denoted in Equation (3).

$$\theta_g(i+1) = \frac{1}{M} \sum_{k=0}^{M-1} \theta_x(k+1) \quad (3)$$

θ_g denotes the revised global variables for the model after the aggregation process. Reducing the training role in FL involves traversing these stages iteratively across several training cycles. Throughout each iteration, the variables of the larger model are modified by including the aggregated inputs from the local versions. This process is repeated until convergence is achieved or a preset stopping criterion is satisfied.

The whole implementation method was conducted in the following manner:

Stage 1: Each client inside the M-clients structure autonomously gathered data on plant leaf illness and saved it in its database.

Stage 2: The records, including photos of plant leaf diseases, were subjected to unique cleansing and pre-processing by each client inside the M-clients platform.

Stage 3: The database was appropriately processed at the peripheries of the M-clients framework to facilitate the categorization of plant leaf illnesses via FL.

Stage 4: During the categorization procedure, a set of 13 trained models was used to extract characteristics from both Independent and Identically Distributed (IID) and non-IID information. The objective was to determine the most optimal oriented approach as the background method.

Stage 5: Every customer independently trained their starting model using a Dense Neural Network approach.

Stage 6: The server began the FL procedure with the starting values. The server distributed the starting values accessible to the M-clients inside the aggregation service.

Stage 7: After obtaining the weights, each client started learning and verification processes using their databases. Upon completing the training procedure, each client sent their revised loads to the server to calculate the federated average loads.

Stage 8: The system utilized the factors obtained from all interconnected clients to compute the federated average scores.

Stage 9: The federated server transmitted the computed federated mean weights obtained from Step 8 to all the linked consumers for further processing.

3.2 IID and Non-IID data

FL encompasses two primary categories for distributing information across participating customers: IID and non-IID information. IID information refers to a scenario when the distribution of information across clients exhibits similarity, and every customer's data indicates the whole dataset. This assumption streamlines the FL procedure since it allows the worldwide framework to be trained using local consumer changes without further considerations. Non-IID data is a scenario where data distribution across clients exhibits heterogeneity or imbalance. This suggests there is potential for variation in the data structure between different clients, and the sample sizes within each client represent the overall database. The factors contributing to non-IID data might include changes in data collecting sources, client statistics discrepancies, and local data dispersion disparities.

3.3 Model validation

The pre-trained models recommended in this study were trained using labeled data categorized into two distinct groups. The dataset consisted of 1500 photos of the "bacteria leaf decay" illness, 1500 photographs of the "Blast" illness, 1300 pictures of the "Brownspot" illness, and 1300 pictures of the "Tungro" illness affecting plant leaves. The framework was constructed using the Keras and Tensorflow 2.0 frameworks, with the 2080Ti as the GPU in this prototype configuration. Using segmented pictures using a 3 × 3 filter facilitated the algorithms' acquisition of crucial characteristics of plant leaf illnesses. The original dimensions of the photos were 512 × 12. They were scaled to 80 × 80 to facilitate the model's training. The dataset was divided between training and testing subsets with a ratio of 80:20. A verification set consisting of 1200 data points, which accounted for twenty percent of the total dataset, was used to assess the efficacy of the algorithms. During the training process, it was necessary to evaluate the learning progress of the algorithms at every stage.

This section presents the LFTL concept in the PLDD. This approach utilizes a client-server architecture, using Keras and TensorFlow frameworks. With a focus on safeguarding data privacy, LFTL effectively minimizes communication overhead by only sending trained models between clients and the server. The FL procedure encompasses three key steps: global model setup, localized model training, and aggregating of local modeling updates. This technique guarantees an effective and privacy-preserving methodology.

4 Simulation analysis and outcomes

The experimental configuration included the implementation of the LFTL architecture on a collection of edge devices with diverse processing capacities. In the FL process, every edge device functioned as a client and was equipped with at least 8 GB RAM and an Intel Core i7 CPU. The server managed the worldwide structure and aggregated the changes from local

models. It was housed on a computer with 16 GB of RAM and an Nvidia 2080Ti GPU. The simulation used TensorFlow 2.0 and Keras frameworks for model construction. The training method consisted of 10 epochs with a learning rate of 0.001.

Fig. 3(a). Accuracy

Fig. 3(b). Privacy preservation score analysis

Figure 3(a) presents the performance metrics of various models. LFTL performs better than other methods, achieving the highest average accuracy of 94.32%. This surpasses the accuracy of Support Vector Machine (SVM) (85.32%), Deep Neural Network (DNN) (91.76%), CNN (92.54%), Artificial Neural Network (ANN) (88.21%), Principal Component Analysis (PCA) (78.45%), Random Forest (RF) (89.12%), Decision Tree (DT) (86.78%), and Naïve Bayes (NB) (81.34%). Figure 3(b) shows the privacy preservation scores, with LFTL demonstrating the most significant average score of 80.76%. This score surpasses those of SVM (70.45%), DNN (74.28%), CNN (76.12%), ANN (72.89%), PCA (65.32%), RF (73.54%), DT (71.23%), and NB (68.45%). The proposed LFTL architecture demonstrates notable advancements in accuracy and privacy protection, positioning it as a resilient solution for identifying and classifying leaf diseases.

Fig. 4(a). Communication overhead

Fig. 4(b). Training time analysis

In Figure 4(a), the results for communication overhead and training time are displayed for various models. LFTL exhibits the lowest average communication overhead of 2.12 MB, outperforming SVM (3.24), DNN (5.89), CNN (6.32), ANN (4.78), PCA (2.45), RF (5.12), DT (4.89), and NB (3.76). In Figure 3(b), the training time results are presented. LFTL demonstrates the most efficient average training time of 35.67 seconds, surpassing the training times of SVM (62.54), DNN (120.76), CNN (110.23), ANN (85.32), PCA (45.67), RF (92.14), DT (78.45), and NB (56.23). The results highlight the proposed LFTL architecture's effectiveness in minimizing communication overhead and reducing training time.

Fig. 5(a). Precision **Fig. 5(b).** Model size analysis

Figure 5(a) presents the model size and precision results for various models. It is observed that LFTL demonstrates the lowest average model size of 4.32 MB, outperforming SVM (8.21), DNN (15.67), CNN (18.45), ANN (11.32), PCA (5.67), RF (13.45), DT (9.76), and NB (7.89). Figure 5(b) shows the precision results, indicating that LFTL has the most excellent average precision of 94.23%. This performance surpasses that of SVM (80.45%), DNN (89.32%), CNN (91.23%), ANN (86.78%), PCA (75.32%), RF (88.21%), DT (82.54%), and NB (78.45%). The results demonstrate the effectiveness and accuracy of the proposed LFTL, indicating its potential as a concise and precise solution.

Fig. 6(a). Recall **Fig. 6(b).** Weight convergence rate analysis

Figure 6(a) presents various models' recall and weight convergence rate outcomes. The results demonstrate that LFTL attains the highest average recall of 94.56%. This performance surpasses that of SVM (75.32%), DNN (88.76%), CNN (91.45%), ANN (82.54%), PCA (68.21%), RF (87.32%), DT (80.45%), and NB (74.23%). The findings shown in Figure 6(b) indicate the convergence rates of different weight algorithms. Among the algorithms tested, LFTL has the highest average convergence rate of 92.12%, surpassing SVM (65.12%), DNN (80.23%), CNN (85.67%), ANN (73.21%), PCA (55.89%), RF (79.45%), DT (70.32%), and NB (63.78%). The results highlight the enhanced memory retention and effective convergence of weight values in the suggested LFTL framework, indicating its resilience and quick convergence as a solution for detecting leaf diseases.

The LFTL demonstrates accuracy of 94.32%, a privacy preservation score of 80.76%, a communication overhead of 2.12 MB, a training time of 35.67 seconds, a model size of 4.32 MB, a precision of 94.23%, a recall of 94.56%, and a weight convergence rate of 92.12%.

The results of the LFTL structure establish it as a reliable and resilient solution for detecting and categorizing leaf diseases.

5 Conclusion and Future Scope

Detecting plant leaf diseases, often called PLDD, is of utmost importance in the context of global food security and improving agricultural output. This study presents the LFTL architecture as a solution to the obstacles presented by data privacy constraints in conventional approaches, emphasizing the need for prompt and effective disease detection in foliage. Traditional methodologies encounter many challenges, such as apprehensions over data privacy and the need for precise, reliable, and efficient models capable of functioning on edge devices. The LFTL architecture, as presented, utilizes FL to address the constraints imposed by data privacy. The system employs a client-server architecture, using Python modules such as Keras and TensorFlow. The primary characteristics include minimizing communication overhead by only transferring trained models and guaranteeing efficient operation on edge devices. The LFTL system has exceptional performance in several aspects, including accuracy (94.32%), privacy preservation score (80.76%), communication overhead (2.12 MB), training time (35.67 seconds), model size (4.32 MB), precision (94.23%), recall (94.56%), and weight convergence rate (92.12%). The results highlight the effectiveness, precision, and ability to maintain privacy of the proposed framework.

The challenges encountered in PLDD include the limited availability of data and the need for instantaneous identification. The LFTL framework effectively addresses these concerns; nonetheless, it faces potential difficulties when dealing with datasets that exhibit significant imbalances and a wide range of PLDD. Future studies must prioritize enhancing LFTL techniques, particularly achieving scalability and resilience in effectively addressing a broader spectrum of PLDD. Further progress was made by investigating methods to boost communication efficiency and mitigate bias in the FL process.

References

1. R. Sujatha, J. M. Chatterjee, N. Z. Jhanjhi, & S. N. Brohi, Microprocessors and Microsystems, **80**, 103615 (2021).
2. C. Rosano-Peña, J. R. Teixeira, & H. Kimura, Environmental Science and Pollution Research, **28(44)**, 62378-62389 (2021).
3. R. K. Lakshmi, & N. Savarimuthu, IEEE Consumer Electronics Magazine, **11(3),** 44-49 (2021).
4. S. Hellinger, M. Knight, A. E. Telfeian, & K. U. Lewandrowski, Lasers in Surgery and Medicine, **54(4),** 530-539 (2022).
5. J. Hu, & Y. Xianyu, Nano Today, **38**, 101143 (2021).
6. T. A. Shaikh, W. A. Mir, T. Rasool, & S. Sofi, Archives of Computational Methods in Engineering, **29(7),** 4557-4597 (2022).
7. C. R. Davies, F. Wohlgemuth, T. Young, J. Violet, M. Dickinson, J. W. Sanders, ... & S. V. Avery, Fungal biology reviews, **36**, 15-26 (2021).
8. S. Vallabhajosyula, V. Sistla, & V. K. K. Kolli, Journal of Plant Diseases and Protection, **129(3),** 545-558 (2022).
9. S. S. Chouhan, U. P. Singh, & S. Jain, Wireless Personal Communications, **121**, 1757-1779 (2021).
10. V. Tiwari, R. C. Joshi, & M. K. Dutta, Ecological Informatics, **63**, 101289 (2021).

11. Y. Zhao, Z. Chen, X. Gao, W. Song, Q. Xiong, J. Hu, & Z. Zhang, IEEE/ACM Transactions on Computational Biology and Bioinformatics, **19(3),** 1817-1826 (2021).
12. N. R. Deepa, & N. Nagarajan, Journal of Ambient Intelligence and Humanized Computing, **12,** 5979-5992 (2021).
13. R. Mahum, H. Munir, Z. U. N. Mughal, M. Awais, F. Sher Khan, M. Saqlain, ... & I. Tlili, Human and Ecological Risk Assessment: An International Journal, **29(2),** 303-326 (2023).
14. R. Gajjar, N. Gajjar, V. J. Thakor, N. P. Patel, & S. Ruparelia, The Visual Computer, 1-16 (2021).
15. V. K. Vishnoi, K. Kumar, & B. Kumar, Journal of Plant Diseases and Protection, **128,** 19-53 (2021).
16. C. S. Shih, J. J. Chou, & K. J. Lin, Journal of Internet Services and Information Security, **8(2),** 1-17 (2018).
17. V. D. Tran, P. T. Le, & V. C. Trinh, Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, **14(1),** 174-188 (2023).